

# Is there Life after Zeno?

## Taking Executions Past the Breaking (Zeno) Point

Aaron D. Ames, Haiyang Zheng, Robert D. Gregg and Shankar Sastry  
 Department of Electrical Engineering and Computer Sciences  
 University of California at Berkeley  
 Berkeley, CA 94720  
 {adames,hyzheng,rdgregg,sastry}@eecs.berkeley.edu

**Abstract**—Understanding Zeno phenomena plays an important role in understanding hybrid systems. A natural—and intriguing—question to ask is: what happens after a Zeno point? Inspired by the construction of [9], we propose a method for extending Zeno executions past a Zeno point for a class of hybrid systems: Lagrangian hybrid systems. We argue that after the Zeno point is reached, the hybrid system should switch to a holonomically constrained dynamical system, where the holonomic constraints are based on the unilateral constraints on the configuration space that originally defined the hybrid system. These principles are substantiated with a series of examples.

### I. INTRODUCTION

Zeno behavior occurs in hybrid systems when there are an infinite number of discrete transitions in a finite period of time. Since each discrete transition takes a finite (and nonzero) amount of computation time, Zeno behavior prevents a simulation from progressing past a certain time, i.e., the simulator “breaks.” Because the physical system that the hybrid system is modeling can exist past the Zeno point, the simulator should be able to correctly predict this “life after Zeno.” This motivates the need to carry a Zeno execution (or trajectory) past the point at which Zeno occurs: the Zeno point.

In this paper, we propose a method for carrying executions past the Zeno point. Although we do not *prove* that this is the correct way to carry executions beyond Zeno points, we argue that our method correctly represents the physical, post-Zeno, behavior of the system being modeled. In order to support this argument, we consider a class of hybrid systems in which we can make inferences about the behavior of the systems after the Zeno point based on their mathematical structure—hybrid systems obtained from (simple) *hybrid Lagrangians: Lagrangian hybrid systems*.

A hybrid Lagrangian consists of a configuration space, a Lagrangian, and a unilateral constraint defining the set of admissible configurations of the system (and usually dictated by physical constraints on the system). From a hybrid Lagrangian, we are able to explicitly construct a Lagrangian hybrid system. Hybrid systems of this form are general enough to model a wide range of physical systems (cf. [7] and the more than 1000 references therein), while

being concrete enough to allow for analysis, e.g., [4] and [5] study how to reduce these systems.

Using the special structure of Lagrangian hybrid systems obtained from hybrid Lagrangians, we are able to demonstrate that the Zeno point must satisfy constraints imposed by the unilateral constraint function. These constraints are *holonomic* in nature, and this implies that *after* the Zeno point the hybrid system should switch to a holonomically constrained dynamical system. The resulting system obtained by “composing” the hybrid system with this dynamical system defines a *completed hybrid system*, which inherently allows an execution to continue past the Zeno point. Since the Zeno point will never be reached in a simulation environment, we discuss how to practically implement a completed hybrid system, and we illustrate these concepts with a series of examples.

### II. HYBRID LAGRANGIANS

In this section, we introduce the notion of a hybrid Lagrangian. Many different forms of “hybrid Lagrangians” have appeared in the literature, although not under this specific name. The goal of the definition introduced here is to be general enough to include these definitions, while being specific enough to allow for explicit constructions.

Let  $Q$  be the *configuration space* for a mechanical system (assumed to be a smooth manifold) and  $TQ$  the tangent bundle of  $Q$ . In this paper, we will consider Lagrangians,  $L : TQ \rightarrow \mathbb{R}$ , describing mechanical, or robotic, systems, which are Lagrangians of the form

$$L(q, \dot{q}) = \frac{1}{2} \dot{q}^T M(q) \dot{q} - V(q), \quad (1)$$

where  $M(q)$  is the inertial matrix,  $\frac{1}{2} \dot{q}^T M(q) \dot{q}$  is the kinetic energy and  $V(q)$  is the potential energy. In this case, the Euler-Lagrange equations yield the equations of motion for the system:

$$M(q) \ddot{q} + C(q, \dot{q}) \dot{q} + N(q) = 0, \quad (2)$$

where  $C(q, \dot{q})$  is the *Coriolis matrix* (cf. [13]) and  $N(q) = \frac{\partial V}{\partial q}(q)$ . Setting  $x = (q, \dot{q})$ , the Lagrangian vector field,  $f_L$ , associated to  $L$  takes the familiar form:

$$\dot{x} = f_L(x) = \begin{pmatrix} \dot{q} \\ M(q)^{-1} (-C(q, \dot{q}) \dot{q} - N(q)) \end{pmatrix}. \quad (3)$$

\*This research is supported by the National Science Foundation (NSF award number CCR-0225610)

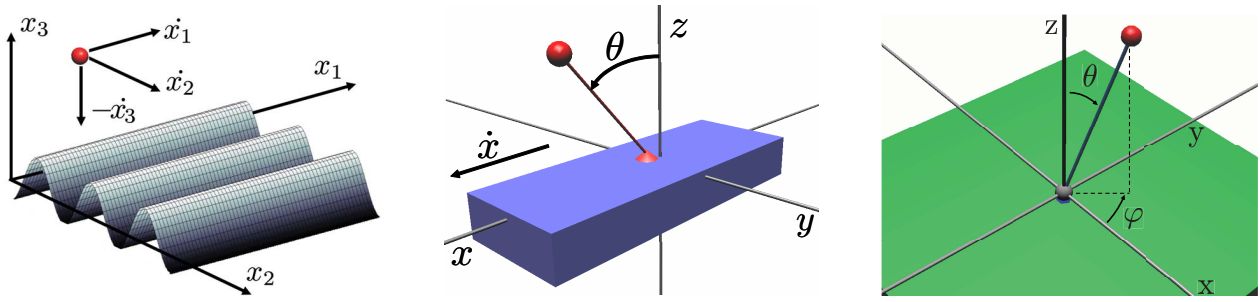


Fig. 1. Ball bouncing on a sinusoidal surface (left). Pendulum on a cart (middle). Spherical pendulum mounted on the ground (right).

This process of associating a dynamical system to a Lagrangian will be mirrored in the setting of hybrid systems. First, we introduce the notion of a hybrid Lagrangian.

**Definition 1:** A simple hybrid Lagrangian is defined to be a tuple

$$\mathbf{L} = (Q, L, h),$$

where

- $Q$  is the configuration space,
- $L : TQ \rightarrow \mathbb{R}$  is a hyperregular Lagrangian,
- $h : Q \rightarrow \mathbb{R}$  provides unilateral constraints on the configuration space; we assume that  $h^{-1}(0)$  is a smooth manifold.

**Example 1 (Ball):** The first running example of this paper is a ball bouncing on a sinusoidal surface (cf. Fig. 1). In this case

$$\mathbf{B} = (Q_{\mathbf{B}}, L_{\mathbf{B}}, h_{\mathbf{B}}),$$

where  $Q_{\mathbf{B}} = \mathbb{R}^3$ , and for  $x = (x_1, x_2, x_3)$ ,

$$L_{\mathbf{B}}(x, \dot{x}) = \frac{1}{2}m\|\dot{x}\|^2 - mgx_3.$$

Finally, we make the problem interesting by considering the sinusoidal constraint function

$$h_{\mathbf{B}}(x_1, x_2, x_3) = x_3 - \sin(x_2).$$

So, for this example, there are trivial dynamics and a nontrivial constraint function.

**Example 2 (Cart):** Our second running example is a constrained pendulum on a cart (cf. Fig. 1). This is a variation on the classical pendulum on a cart, where the pendulum is not allowed to “pass through” the cart, i.e., the cart gives physical constraints on the configuration space. In this case

$$\mathbf{C} = (Q_{\mathbf{C}}, L_{\mathbf{C}}, h_{\mathbf{C}}),$$

where  $Q_{\mathbf{C}} = \mathbb{S}^1 \times \mathbb{R}$ ,  $q = (\theta, x)$ , and  $L_{\mathbf{C}}$  is the standard Lagrangian associated to this system. Finally, the constraint that the pendulum is not allowed to pass through the cart is manifested in the constraint function  $h_{\mathbf{C}}(\theta, x) = \cos(\theta)$ .

**Example 3 (Pendulum):** Our final running example is a spherical pendulum mounted on the ground (cf. Fig. 1). In

this case, the presence of the ground gives the physical constraints on the configuration space. That is, we define

$$\mathbf{P} = (Q_{\mathbf{P}}, L_{\mathbf{P}}, h_{\mathbf{P}}),$$

where  $Q_{\mathbf{P}} = \mathbb{S}^2$ ,  $q = (\theta, \varphi)$ , and  $L_{\mathbf{P}}$  is the standard Lagrangian for this system. Finally, the constraint that the pendulum is not allowed to pass through the ground is manifested in the constraint function  $h_{\mathbf{P}}(\theta, \varphi) = \cos(\theta)$ .

**Definition 2:** A simple hybrid system is a tuple:

$$\mathcal{H} = (D, f, G, R),$$

where

- $D$  is a smooth manifold,
- $f$  is a vector field on that manifold,
- $G$  is an embedded submanifold of  $G$  called the *guard*,
- $R$  is a smooth embedding  $R : G \rightarrow D$  called the *reset map*.

For a Lagrangian (1), there is an associated dynamical system (3). Similarly, for a hybrid Lagrangian,  $\mathbf{L} = (Q, L, h)$ , there is an associated *Lagrangian hybrid system*:

$$\mathcal{H}_{\mathbf{L}} = (D_{\mathbf{L}}, f_{\mathbf{L}}, G_{\mathbf{L}}, R_{\mathbf{L}}).$$

First, we define

$$\begin{aligned} D_{\mathbf{L}} &= \{(q, \dot{q}) \in TQ : h(q) \geq 0\}, \\ G_{\mathbf{L}} &= \{(q, \dot{q}) \in TQ : h(q) = 0 \text{ and } A(q)\dot{q} \leq 0\}, \end{aligned}$$

where

$$A(q) = \frac{\partial h}{\partial q}(q) = \left( \frac{\partial h}{\partial q_1}(q) \quad \cdots \quad \frac{\partial h}{\partial q_n}(q) \right).$$

Since this is the setup for systems with unilateral constraints (cf. [7]), we define the reset map

$$R_{\mathbf{L}}(q, \dot{q}) = (q, P(q, \dot{q})),$$

based on the impact equation

$$P(q, \dot{q}) = \dot{q} - (1 + e) \frac{A(q)\dot{q}}{A(q)M(q)^{-1}A(q)^T} M(q)^{-1} A(q)^T,$$

where  $0 \leq e \leq 1$  the coefficient of restitution which is a measure of the energy dissipated through impact.

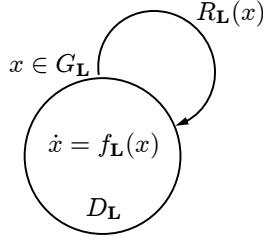


Fig. 2. Lagrangian hybrid system obtained from a hybrid Lagrangian.

Finally,  $f_L = f_L$  is the Lagrangian vector field associated to  $L$ . A graphical representation of the hybrid system is given in Fig. 2.

**Example 4:** Utilizing the above construction, we can associate to each of our running examples—the bouncing ball, the pendulum on a cart and the spherical pendulum—hybrid systems  $\mathcal{H}_B$ ,  $\mathcal{H}_C$  and  $\mathcal{H}_P$ , respectively.

**Definition 3:** An execution of  $\mathcal{H}$  is a tuple

$$\chi^{\mathcal{H}} = (\Lambda, \mathcal{J}, \mathcal{C}),$$

where

- $\Lambda = \{0, 1, 2, \dots\} \subseteq \mathbb{N}$  is a finite or infinite indexing set,
- $\mathcal{J} = \{I_i\}_{i \in \Lambda}$  is a *hybrid interval* where  $I_i = [\tau_i, \tau_{i+1}]$  if  $i, i+1 \in \Lambda$  and  $I_{N-1} = [\tau_{N-1}, \tau_N]$  or  $[\tau_{N-1}, \tau_N]$  or  $[\tau_{N-1}, \infty)$  if  $|\Lambda| = N$ ,  $N$  finite, with  $\tau_i, \tau_{i+1}, \tau_N \in \mathbb{R}$  and  $\tau_i \leq \tau_{i+1}$ ,
- $\mathcal{C} = \{c_i\}_{i \in \Lambda}$  is a collection of solutions of  $f$ , i.e.,  $\dot{c}_i(t) = f(c_i(t))$  for all  $i \in \Lambda$ ,

such that the following conditions hold for every  $i, i+1 \in \Lambda$ ,

- $c_i(\tau_{i+1}) \in G$ ,
- $R(c_i(\tau_{i+1})) = c_{i+1}(\tau_{i+1})$ .

### III. UNDERSTANDING ZENO BEHAVIOR

We begin this section by defining Zeno executions (cf. [1]-[6], [10], [11] and [15] for more on Zeno behavior). We then discuss two important classes of Zeno executions: *chattering Zeno executions* and *genuinely Zeno executions*. In the context of Lagrangian hybrid systems obtained from hybrid Lagrangians, we relate the coefficient of restitution with these two types of executions, and we give conditions that the Zeno point of a Zeno execution must satisfy in either case.

**Definition 4:** An execution  $\chi^{\mathcal{H}_L}$  is *Zeno* if  $\Lambda = \mathbb{N}$  and

$$\lim_{i \rightarrow \infty} \tau_i = \sum_{i=0}^{\infty} (\tau_{i+1} - \tau_i) = \tau_{\infty}$$

for some finite  $\tau_{\infty}$ , termed the *Zeno time*. If  $\chi^{\mathcal{H}_L}$  is Zeno, then its *Zeno point* is defined to be

$$x_{\infty} = (q_{\infty}, \dot{q}_{\infty}) = \lim_{i \rightarrow \infty} c_i(\tau_i) = \lim_{i \rightarrow \infty} (q_i(\tau_i), \dot{q}_i(\tau_i)).$$

Here  $c_i = (q_i, \dot{q}_i) \in \mathcal{C}$ , and the Zeno point is necessarily a single point because of the specific problem formulation considered in this paper.

A hybrid system is Zeno if it admits a Zeno execution, i.e., if there exists an execution  $\chi^{\mathcal{H}_L}$  that is Zeno.

The definition of a Zeno execution results in two qualitatively different types of Zeno behavior (cf. [3]); they are defined as follows: a Zeno execution  $\chi^{\mathcal{H}_L}$  is

**Chattering Zeno:** If there exists a finite  $C$  such that  $\tau_{i+1} - \tau_i = 0$  for all  $i \geq C$ .

**Genuinely Zeno:** If  $\tau_{i+1} - \tau_i > 0$  for all  $i \in \mathbb{N}$ .

The difference between these is especially prevalent in their detection and elimination. Chattering Zeno executions result from the existence of a switching surface on which the vector fields “oppose” each other; for this reason they are easy to detect. Filippov solutions can be defined on these surfaces in order to force the flow to “slide” along the switching surface (cf. [9]). Later in this paper we will generalize this technique to extend genuinely Zeno executions past the Zeno point.

Genuinely Zeno executions are much more complicated in their behavior. There are very few methods currently available to detect the existence of genuinely Zeno executions, namely [1] and [6]. Little has been done in the area of eliminating these executions, although there have been some results ([2] and [11]) for special classes of hybrid systems. This is the case because genuinely Zeno executions are fundamentally global in nature, which prevents the use of local techniques in their analysis.

The hybrid systems considered in this paper display both chattering and genuinely Zeno behavior; roughly speaking, the coefficient of restitution can be used to differentiate between these systems. Moreover, Zeno points must satisfy certain constraints based on the unilateral constraint function. That is, we make the following observations:

**CZ:** If  $\mathcal{H}_L$  has a chattering Zeno execution,  $\chi^{\mathcal{H}_L}$ , then  $C = 1$ , i.e.,  $\tau_{\infty} = \tau_1 - \tau_0$  and  $x_{\infty} = (q_1(\tau_1), \dot{q}_1(\tau_1))$  with  $h(q_1(\tau_1)) = 0$  and  $A(q_1(\tau_1))\dot{q}_1(\tau_1) = 0$ .

**GZ:** If  $\mathcal{H}_L$  has a genuinely Zeno execution, then  $0 < e < 1$ . Moreover, if  $\chi^{\mathcal{H}_L}$  is a genuinely Zeno execution, then  $x_{\infty} = (q_{\infty}, \dot{q}_{\infty})$  is a point with  $h(q_{\infty}) = 0$ , and  $A(q_{\infty})\dot{q}_{\infty} = 0$ .

### IV. GOING BEYOND ZENO

In this section, we propose a method for going beyond Zeno points in the class of systems considered in this paper: Lagrangian hybrid systems. This method is supported by the structure of the systems begin considered, although we do not claim to *prove* that this is the right way to carry executions beyond Zeno points, or even that executions should necessarily be carried beyond Zeno points. We do claim that if one wishes to carry an execution beyond a Zeno point, and the system being considered is obtained from a hybrid Lagrangian, then this procedure gives a method by

which to do so. The authors are unaware of any similar results in the literature.

We will assume that the hybrid system being considered here is Zeno, and that  $\chi^{\mathcal{H}_L}$  is a Zeno execution. Of course, for this procedure to be justifiable in a simulation framework, one must verify *a priori* that the system being considered is Zeno. Although we will not explicitly derive conditions on when hybrid systems are Zeno, this is an active area of research for the authors (cf. [1] and [6]). In the next section, we will indicate how to practically implement the results given in this section. We begin by summarizing the observations of the previous section by stating them concisely.

**Main Observation.** *If  $\chi^{\mathcal{H}_L}$  is a Zeno execution of a Lagrangian hybrid system  $\mathcal{H}_L$ , then the Zeno point  $x_\infty = (q_\infty, \dot{q}_\infty)$  is a point satisfying  $h(q_\infty) = 0$  and  $A(q_\infty)\dot{q}_\infty = 0$ .*

This observation indicates how the system should behave *after* the Zeno point, i.e., it should satisfy a holonomic constraint. This holonomic constraint forces the system to slide along surface  $h^{-1}(0) = \{q \in Q : h(q) = 0\}$ . From this, we argue that after the Zeno point, the hybrid system should switch to a holonomically constrained dynamical system.

Recall that for a holonomically constrained system described by a Lagrangian,  $L$ , of the form given in (1), the equations of motion for the holonomically constrained system are obtained from the equations of motion for the unconstrained system (2); they are given by (cf. [13])

$$M(q)\ddot{q} + C(q, \dot{q})\dot{q} + N(q) + A(q)^T \lambda = 0,$$

where  $\lambda$  is the Lagrange multiplier, which in this case is given by

$$\lambda = (A(q)M(q)^{-1}A(q)^T)^{-1} \left( A(q)\dot{q} - A(q)M(q)^{-1}(C(q, \dot{q})\dot{q} + N(q)) \right).$$

From the constrained equations of motion, for  $x = (q, \dot{q})$ , we get the vector field

$$\begin{aligned} \dot{x} &= f_L^\infty(x) \\ &= \begin{pmatrix} \dot{q} \\ M(q)^{-1}(-C(q, \dot{q})\dot{q} - N(q) - A(q)^T \lambda) \end{pmatrix}. \end{aligned}$$

Note that the  $f_L^\infty$  defines a vector field on the manifold  $TQ|_{h^{-1}(0)}$ , from which we obtain the dynamical system

$$\mathcal{D}_L^\infty = (TQ|_{h^{-1}(0)}, f_L^\infty).$$

This, when coupled with the Main Observation, will be essential to understanding how to carry executions beyond Zeno points.

We begin by considering the case when  $\mathcal{H}_L$  is a chattering Zeno hybrid system; in this case, the idea of carrying executions past the Zeno point has been well-studied. In [9], it is argued that once the solution hits the ‘‘switching surface’’ (or, in our case, the guard), the solution should slide along the switching surface. In terms of Zeno points, this implies that

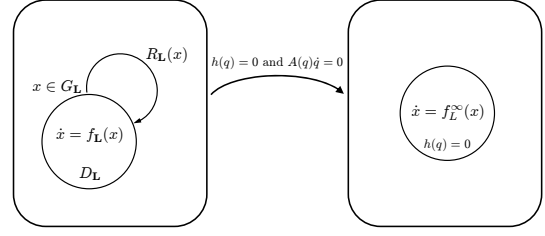


Fig. 3. A completed hybrid system:  $\overline{\mathcal{H}}_L$ .

before the Zeno point the dynamics should be dictated by  $f_L$ , while after the Zeno point the dynamics should be dictated by  $f_L^\infty$ . We can generalize this construction to include genuinely Zeno Lagrangian hybrid systems.

**Definition 5:** If  $\mathcal{H}_L$  is a Lagrangian hybrid system, we define the corresponding *completed hybrid system*<sup>1</sup> (or the *completion* of  $\mathcal{H}_L$ ) as

$$\overline{\mathcal{H}}_L := \begin{cases} \mathcal{D}_L^\infty & \text{if } h(q) = 0 \text{ and } A(q)\dot{q} = 0 \\ \mathcal{H}_L & \text{otherwise.} \end{cases}$$

A completed hybrid system, as obtained from a Lagrangian hybrid system  $\mathcal{H}_L$ , can be seen in Fig. 3. To make the definition of the completed system somewhat more transparent, some comments are in order. The Main Observation indicates that the only way for the transition to be made from the hybrid system  $\mathcal{H}_L$  to the dynamical system  $\mathcal{D}_L^\infty$  is if a specific Zeno execution reaches its Zeno point. Therefore, before the Zeno point, the Zeno system simply will be the hybrid system  $\mathcal{H}_L$ , while after the Zeno point, the completed system will be the dynamical system  $\mathcal{D}_L^\infty$ . Since the dynamical system forces the system dynamics to be constrained to the manifold defined by  $h^{-1}(0)$ , this implies that the completed system will slide along the guard (switching surface) after the Zeno point.

This can be understood further on the level of executions. We can define an execution of the completed hybrid system  $\overline{\mathcal{H}}_L$  by concatenating a Zeno execution of  $\mathcal{H}_L$  with a solution to the dynamical system  $\mathcal{D}_L^\infty$ .

Specifically, let  $\chi^{\mathcal{H}_L} = (\mathbb{N}, \mathcal{J}, \mathcal{C})$  be a Zeno execution of  $\mathcal{H}_L$ . We obtain an execution (or solution) of the completed hybrid system  $\overline{\mathcal{H}}_L$  by defining it to be

$$\chi^{\overline{\mathcal{H}}_L} = (\mathbb{N} \cup \{\infty\}, \overline{\mathcal{J}}, \overline{\mathcal{C}}),$$

where

$$\overline{\mathcal{J}} = \mathcal{J} \cup \{I_\infty = [\tau_\infty, \infty)\}, \quad \overline{\mathcal{C}} = \mathcal{C} \cup \{c_\infty\},$$

with  $c_\infty(t)$  a solution to  $f_L^\infty$  with initial condition the Zeno point:

$$c_\infty(\tau_\infty) = x_\infty = (q_\infty, \dot{q}_\infty).$$

In the next section, we will discuss how to simulate solutions of completed systems.

<sup>1</sup>This terminology (and notation) is borrowed from topology, where a metric space can be completed to ensure that ‘‘limits exist.’’

## V. MODELING AND SIMULATION

We discuss two practical issues when modeling and simulating completed hybrid systems (see Fig. 3). These issues are related to the transition from the left state  $\mathcal{H}_L$  to the right state  $\mathcal{D}_L^\infty$ , and the corresponding initial conditions of  $\mathcal{D}_L^\infty$ . The theoretical framework established in this paper allows us to justifiably surmount the practical problems introduced in simulation.

The first simulation issue is derived from the unavoidable numerical errors caused by finite representations of values in a computer and truncation errors introduced by practical ODE solvers, i.e., a simulator produces an approximate execution  $\widehat{\chi}^{\mathcal{H}_L}$  to the execution  $\chi^{\mathcal{H}_L}$ . Therefore, we can't guarantee or expect a solution to reach the exact Zeno point  $x_\infty = (q_\infty, \dot{q}_\infty)$ . Moreover, in order to reach the Zeno point, an infinite number of computation steps have to be performed (in a finite amount of time). Therefore, instead of resolving a solution that passes through the Zeno point exactly, we will compute an approximation of the Zeno solution; the approximated solution will pass through a neighborhood of the Zeno point, so we must modify the transition to the system  $\mathcal{D}_L^\infty$  accordingly. Before discussing the details of the construction of the approximate solution, we first address the second modeling issue.

The other concern is the reinitialization of the new constrained system  $\mathcal{D}_L^\infty$ . In other words, after the transition to this system, we must give initial conditions for the constrained system. Theoretically, the initial condition is the Zeno point, but because in simulation we do not actually reach the Zeno point, an initial condition must be estimated—one that satisfies the same conditions as a Zeno point:  $h(q_\infty) = 0$  and  $A(q_\infty)\dot{q}_\infty = 0$ .

The approximation to the completed hybrid system  $\overline{\mathcal{H}}_L$ , denoted by  $\overline{\mathcal{H}}_L^\epsilon$ , is given by

$$\overline{\mathcal{H}}_L^\epsilon := \begin{cases} \mathcal{D}_L^\infty & \text{if } \text{abs}(h(q)) \leq \epsilon \text{ and } \text{abs}(A(q)\dot{q}) \leq \epsilon \\ \mathcal{H}_L & \text{otherwise.} \end{cases}$$

When switching from  $\mathcal{H}_L$  to  $\mathcal{D}_L^\infty$  via the approximated guard condition, we use a map which resets the variables so that they satisfy the conditions of a Zeno point:  $h(q_\infty) = 0$  and  $A(q_\infty)\dot{q}_\infty = 0$ . Specifically, for a point  $(q, \dot{q})$  satisfying the approximate guard condition

$$\text{abs}(h(q)) \leq \epsilon \text{ and } \text{abs}(A(q)\dot{q}) \leq \epsilon,$$

we define a reset map  $R^\infty$  which sends  $(q, \dot{q})$  to an approximate Zeno point,  $(\widehat{q}_\infty, \widehat{\dot{q}}_\infty) = R^\infty(q, \dot{q})$ , satisfying

$$h(\widehat{q}_\infty) = 0 \text{ and } A(\widehat{q}_\infty)\widehat{\dot{q}}_\infty = 0.$$

We now briefly discuss how to construct the map  $R^\infty$  for the running examples in this paper. In all of these examples, the vector fields for the constrained dynamical systems are easy to calculate.

**Example 5:** We begin by considering the ball. Note that

$$\begin{aligned} h_{\mathbf{B}}(x) &= 0 & \Rightarrow & x_3 = \sin(x_2), \\ A_{\mathbf{B}}(x)\dot{x} &= 0 & \Rightarrow & \dot{x}_3 - \cos(x_2)\dot{x}_2 = 0. \end{aligned}$$

In the above two equations, there are four variables— $x_2$ ,  $\dot{x}_2$ ,  $x_3$ , and  $\dot{x}_3$ —involved but only two constraints, so we can't resolve all of the variable values, i.e., we must use part of the simulation results to obtain the initial conditions for the constrained systems. Finally, there are no extra constraints for the rest of the variables of the system— $x_1$  and  $\dot{x}_1$ —and their initial conditions are simply the simulation results. Therefore, if we use the simulation results of  $x_2$  and  $\dot{x}_2$  to obtain the initial conditions for the constrained system, the complete reset map is:

$$R_{\mathbf{B}}^\infty(x_1, x_2, x_3, \dot{x}_1, \dot{x}_2, \dot{x}_3) = (x_1, x_2, \sin(x_2), \dot{x}_1, \dot{x}_2, \cos(x_2)\dot{x}_2).$$

For the other running examples, the calculations are the same. The end result is that:

$$\begin{aligned} R_{\mathbf{C}}^\infty(\theta, x, \dot{\theta}, \dot{x}) &= (\text{sign}(\theta)\pi/2, x, 0, \dot{x}), \\ R_{\mathbf{P}}^\infty(\theta, \varphi, \dot{\theta}, \dot{\varphi}) &= (\pi/2, \varphi, 0, \dot{\varphi}). \end{aligned}$$

We use HyVisual (cf. [8]) as the modeling and simulation tool. The semantics of a transition in this tool is that whenever its guard expression becomes true, the transition is taken immediately (cf. [12]). An event detection mechanism is deployed to ensure that a transition is taken close to the time point when its guard expression becomes enabled. These semantics are very important in ensuring that the simulation approximates the exact Zeno solutions.

Next, we take a close look at the simulation results of the Ball example to illustrate how we approximate the exact Zeno solution.

**Example 6 (Ball):** In this example, we choose the initial conditions as  $x_0 = (q_0, \dot{q}_0) = (0.0, \pi/3, 2.0, 0.5, 0.0, 0.0)$ , and the coefficient of restitution  $e$  is chosen as 0.6.

We begin by simulating the hybrid system  $\mathcal{H}_{\mathbf{B}}$ , the results of which are shown in Fig. 4 (to which the rest of the paragraph refers). The simulation time is set to 6.0, but the simulation gets stuck at around 2.04; the bottom figure indicates that the ball tries, but is unable, to climb upwards along the surface (a sinusoidal waveform). Its velocities decrease during this process due to the energy loss through impact as can be seen in the top figure. As a consequence, more and more collisions are triggered and the time interval between two consecutive collisions keeps shrinking. The dense points near time 2.04 in the middle figure indicate that more and more computation steps are taken, which makes the simulation halt. This behavior is indicative of genuinely Zeno behavior.

Fig. 5 shows a simulation of the completed hybrid system  $\overline{\mathcal{H}}_{\mathbf{B}}^\epsilon$  with the same initial conditions. Note that the simulation closely approaches the Zeno point before the behavior of the ball automatically switches to what  $f_{\mathbf{B}}^\infty$  specifies, i.e., the ball oscillates along the surface (a sinusoidal waveform). Therefore, the simulation does not halt, freely moving beyond the Zeno point in a manner consistent with physical reality.

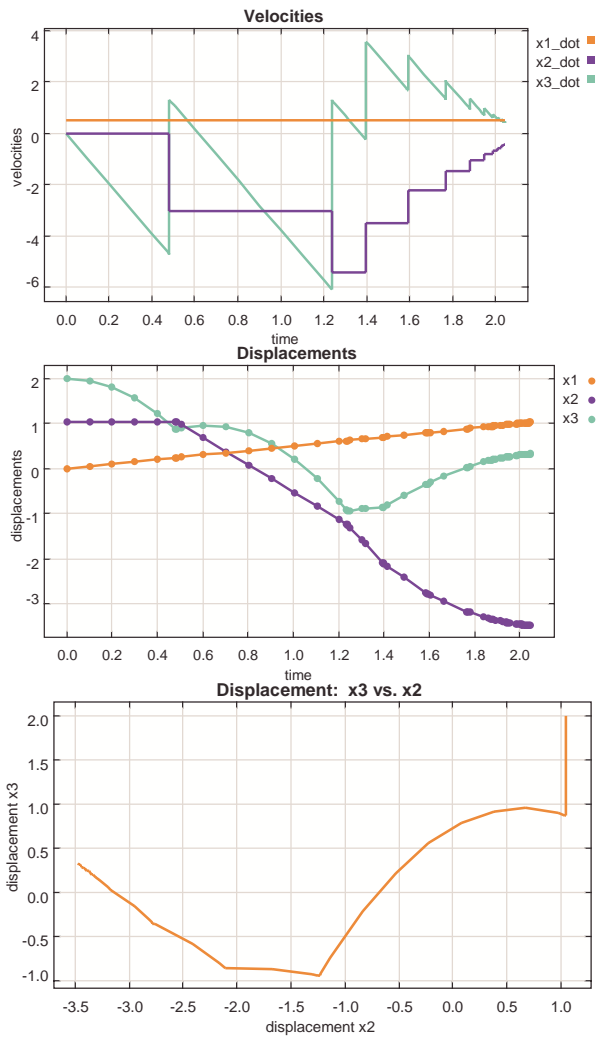


Fig. 4. Simulation gets stuck at the Zeno point. (Top) Velocities over the time. (Middle) Displacements over the time. (Bottom) Displacement on the  $x_3$  direction vs. the displacement on the  $x_2$  direction.

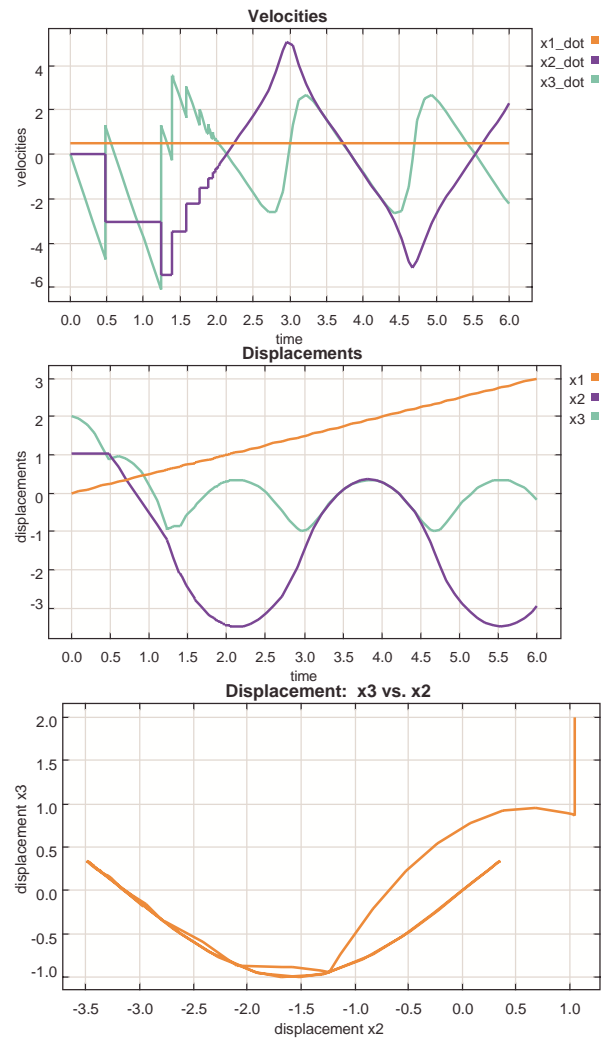


Fig. 5. Simulation goes beyond the Zeno point. (Top) Velocities over the time. (Middle) Displacements over the time. (Bottom) Displacement on the  $x_3$  direction vs. the displacement on the  $x_2$  direction.

For the other running examples of this paper—the cart and the pendulum—the modeling and simulation techniques are basically the same. All of these models are available as demos in release 6.0 of HyVisual and Ptolemy [14].

## REFERENCES

- [1] A. D. Ames, A. Abate, and S. Sastry, "Sufficient conditions for the existence of Zeno behavior," in *Proceedings of the 44th IEEE Conference on Decision and Control and European Control Conference*, Seville, Spain, 2005.
- [2] A. D. Ames and S. Sastry, "Blowing up affine hybrid systems," in *Proceedings of the 43rd Conference on Decision and Control*, Paradise Island, Bahamas, 2004.
- [3] —, "Characterization of Zeno behavior in hybrid systems using homological methods," in *Proceedings of the 24th American Control Conference*, Portland, OR, 2005.
- [4] —, "Hybrid cotangent bundle reduction of simple hybrid mechanical systems with symmetry," in *Proceedings of the 25th American Control Conference*, Minneapolis, MN, 2006.
- [5] —, "Hybrid Routhian reduction of Lagrangian hybrid systems," in *Proceedings of the 25th American Control Conference*, Minneapolis, MN, 2006.
- [6] A. D. Ames, P. Tabuada, and S. Sastry, "On the stability of Zeno equilibria," in *Hybrid Systems: Computation and Control*, J. Hespanha and A. Tiwari, Eds., vol. LNCS 3927. Santa Barbara, CA: Springer-Verlag, 2006, pp. 34–48.
- [7] B. Brogliato, *Nonsmooth Mechanics*. Springer-Verlag, 1999.
- [8] A. Cataldo, C. Hylands, E. A. Lee, J. Liu, X. Liu, S. Neuendorffer, and H. Zheng, "Hyvisual: A hybrid system visual modeler," University of California, Berkeley, Tech. Rep. UCB/ERL M03/30, 2003.
- [9] A. Filippov, *Differential equations with discontinuous right-hand sides*. Kluwer Academic Publishers, 1988.
- [10] M. Heymann, F. Lin, G. Meyer, and S. Resmerita, "Analysis of Zeno behaviors in hybrid systems," in *Proceedings of the 41st IEEE Conference on Decision and Control*, Las Vegas, NV, 2002.
- [11] K. H. Johansson, J. Lygeros, S. Sastry, and M. Egerstedt, "Simulation of Zeno hybrid automata," in *Proceedings of the 38th IEEE Conference on Decision and Control*, Phoenix, AZ, 1999.
- [12] E. A. Lee and H. Zheng, "Operational semantics of hybrid systems," in *Hybrid Systems: Computation and Control*, M. Morari and L. Thiele, Eds., vol. LNCS 3414. Zurich, Switzerland: Springer-Verlag, 2005, pp. 25–53.
- [13] R. M. Murray, Z. Li, and S. Sastry, *A Mathematical Introduction to Robotic Manipulation*. CRC Press, 1993.
- [14] <http://ptolemy.eecs.berkeley.edu>, the Ptolemy Project.
- [15] J. Zhang, K. H. Johansson, J. Lygeros, and S. Sastry, "Zeno hybrid systems," *Int. J. Robust and Nonlinear Control*, vol. 11, no. 2, pp. 435–451, 2001.