Simulation of Zeno Hybrid Automata*

Karl Henrik Johansson† John Lygeros† Shankar Sastry† Magnus Egerstedt‡

Abstract

Zeno hybrid automata are hybrid systems that can exhibit infinitely many discrete transitions in a finite time interval. Such automata arise due to modeling simplifications and may deteriorate simulation efficiency and accuracy considerably. Some basic properties of Zeno hybrid automata are explored. Possible ways to extend a simulation beyond the Zeno time are suggested. It is shown that for a particular class of Zeno hybrid automata, Filippov solutions give a natural extension.

1 Introduction

Despite considerable recent progress towards the development of formal controller synthesis and analysis methods for hybrid systems [23, 6, 16], design evaluation through simulation remains an invaluable tool. The simulation of hybrid systems poses a number of theoretical and computational problems, not encountered in conventional continuous systems. These problems include guard crossing, lack of existence of solutions for certain initial conditions, lack of uniqueness of solutions, and a situation where the model of the system being simulated takes an infinite number of discrete transitions in a finite amount of time. The latter is referred to as Zeno. The problems of blocking and non-determinism have been addressed formally for several classes of hybrid systems [25, 9, 15]. However, the Zeno phenomenon may still make hybrid simulation imprecise and time-consuming. Several simulation packages, such as Dymola [8], Omola [19], and SHIFT [7], have been developed for hybrid systems. Some of them use numerical routines for detection of discrete transitions. The mixed continuous-time and discrete-event simulation is efficient and accurate. In some cases, however, the simulation problem is generically difficult and time-consuming due to fast switching between the discrete states. The reason for this is that the state-of-the-art numerical solvers divide the simulation problem into sequences of regular continuous simulations and instances of solving algebraic equations for the transition times [3]. Therefore the simulation get stuck when a large number of transitions appear in a short time interval.

Zeno hybrid automata give infinitely many transitions in finite time. They have only been studied to some extent [1, 10, 4, 2], even if Zenoness is an important property of systems all kind of systems with mixed continuous and discrete dynamics. Physical systems are not Zeno. But due to modeling simplification, models of real systems can be Zeno. In complex hybrid systems, it may be difficult to

*Manuscript submitted to CDC 1999 in Phoenix, AZ

†Department of EECS, UC Berkeley, 333 Cory Hall # 1770, Berkeley, CA 94720-1770, +1 510 642-1906 (voice), +1 510 642-1341 (fax), {johans|lygeros|sastry}@eecs.berkeley.edu

‡Division of Optimization and Systems Theory, Royal Institute of Technology, Stockholm, Sweden, magnuse@math.kth.se

detect Zeno. Therefore, it would be nice if simulation softwares could detect Zeno and either resolve the Zenoness automatically or with support from the user.

The main contribution of this paper is to show some basic properties of Zeno hybrid automata together with possible methods to resolve Zenoness. A number of Zeno systems is presented. They are simple in their setup, but still illustrate several interesting phenomena. Motivation for the work is to support the development of general-purpose simulation tools, such as Dymola, Omola, and SHIFT. These packages are not restricted to certain classes of system, such as mechanical or chemical systems, which makes the problem challenging. Zeno systems or systems close to Zeno have been studied in the literature in particular cases. Averaging techniques have been fruitful when analyzing mechanical impact oscillators [21], DC/DC buck converters [5], and relay feedback systems [11]. In this paper we briefly outline averaging for general hybrid systems. Regularization is another possible way to analyze Zeno hybrid automata that we described. It was shown in [12] that the relaxed solution may be depending on the way the regularization is performed. A third approach to analyze Zeno systems is to restrict the attention to a class of hybrid automata that we will denote Filippov automata. These are described by a number of closed subsets of $\mathbb{R}^n$ with disjoint interiors and with a smooth vector field in each set. Zeno may in these systems arise if the boundary of two sets intersect and the vector field in each set is pointing out of the set. Such Zenoness can be resolved by introducing a sliding mode solution as the continuation beyond the Zeno time. This was suggested in [18].

The outline of the paper is as follows. In Section 2 the formal notation of hybrid automata is presented. Zeno hybrid automata is introduced in Section 3, by illustrating their variations by examples. Section 4 presents some convergence results for Zeno executions. We propose methods in Section 5 for extending Zeno executions to infinity. Some conclusions are given in Section 6.

2 Hybrid Automata

In this section we present some background material, such as definitions of hybrid automaton and executions. The definitions are based on [16, 12]. We start by recalling some notation.

For a finite collection V of variables, let $\mathbf{V}$ denote the set of valuations of these variables. We use lower case letters to denote both a variable and its valuation. We refer to variables whose set of valuations is finite as *discrete* and to variables whose set of valuations is a subset of a Euclidean space as *continuous*. The Euclidean norm is denoted $\|\cdot\|$ and the convex hull $\text{co}\{\cdot\}$. For a subset U of a topological space we use $\overline{U}$ to denote its closure, U^o its interior, ∂U its boundary, U^c its complement, $|U|$ its cardinality, and 2^U the set of all subsets of U. We use $\wedge$ to denote the logical "and", and $\vee$ to denote the logical "or".

Definition 1 (Hybrid Automaton)
A hybrid automaton H is a collection $H = (Q, X, \text{Init}, f, I, E, G, R)$, where

- *Q is a finite collection of discrete variables;*
- *X is a finite collection of continuous variables with $\mathbf{X} = \mathbb{R}^n$;*
- $\text{Init} \subseteq \mathbf{Q} \times \mathbf{X}$ *is a set of initial states;*
- *$f : \mathbf{Q} \times \mathbf{X} \to T\mathbf{X}$ is a vector field, Lipschitz continuous in its second argument;*
- *$I : \mathbf{Q} \to 2^{\mathbf{X}}$ assigns to each $q \in \mathbf{Q}$ an invariant set;*

- $E \subset \mathbf{Q} \times \mathbf{Q}$ *is a collection of edges;*
- $G : E \to 2^{\mathbf{X}}$ *assigns to each edge* $e = (q, q') \in E$ *a guard; and*
- $R : E \times \mathbf{X} \to 2^{\mathbf{X}}$ *assigns to each edge* $e = (q, q') \in E$ *and* $x \in \mathbf{X}$ *a reset relation.*

We refer to $(q, x) \in \mathbf{Q} \times \mathbf{X}$ as the state of H. Pictorially, a hybrid automaton can be represented by a directed graph, $(\mathbf{Q}, E)$, with vertices $\mathbf{Q}$ and edges E. With each vertex, $q \in \mathbf{Q}$, we associate a set of continuous initial states $\mathrm{Init}_q = \{x \in \mathbf{X} \ : \ (q, x) \in \mathrm{Init}\}$, a vector field, $f_q(x) = f(q, x)$, and an invariant, $I(q)$. With each edge, $e \in E$, we associate a guard, $G(e)$, and a reset relation $R(e, x)$.

Definition 2 (Hybrid Time Trajectory)
A hybrid time trajectory $\tau = \{I_i\}_{i=0}^N$ *is a finite or infinite sequence of intervals of the real line, such that*

- $\tau_i \leq \tau_i'$ *for* $i \geq 0$ *and* $\tau_i = \tau_{i-1}'$ *for* $i > 0$;
- $I_i = [\tau_i, \tau_i']$ *for* $i < N$; *and*
- $I_N = [\tau_N, \tau_N']$ *or* $I_N = [\tau_N, \tau_N')$ *if* $N < \infty$.

Note that hybrid time trajectories can extend to infinity if τ is an infinite sequence or if it is a finite sequence ending with an interval of the form $[\tau_N, \infty)$. We denote by $\mathcal{T}$ the set of all hybrid time trajectories.

Definition 3 (Execution)
An execution χ *of a hybrid automaton* H *is a collection* $\chi = (\tau, q, x)$ *with* $\tau \in \mathcal{T}$, $q : \tau \to \mathbf{Q}$, *and* $x : \tau \to \mathbf{X}$, *satisfying*

- $\big(q(\tau_0), x(\tau_0)\big) \in \mathrm{Init}$ *(initial condition);*
- *for all* i *with* $\tau_i < \tau_i'$, $x(t)$ *is continuously differentiabla and* $q(t)$ *is constant for* $t \in [\tau_i, \tau_i']$, *and* $x(t) \in I\big(q(t)\big)$ *and* $dx(t)/dt = f\big(q(t), x(t)\big)$ *for all* $t \in [\tau_i, \tau_i')$ *(continuous evolution); and*
- *for all* i, $e = \big(q(\tau_i'), q(\tau_{i+1})\big) \in E$, $x(\tau_i') \in G(e)$, *and* $x(\tau_{i+1}) \in R\big(e, x(\tau_i')\big)$ *(discrete evolution).*

We say a hybrid automaton *accepts* an execution χ. For an execution $\chi = (\tau, q, x)$ we use $(q_0, x_0) = \big(q(\tau_0), x(\tau_0)\big)$ to denote the initial state of χ. An execution is *finite* if τ is finite sequence ending with a closed interval; it is called *infinite* if it is either an infinite sequence or if $\sum_i(\tau_i' - \tau_i) = \infty$.

Definition 4 (Non-Blocking and Deterministic Automaton)
A hybrid automaton H *is called non-blocking if it accepts at least one infinite executions for all* $(q_0, x_0) \in \mathrm{Init}$. *It is called deterministic if it accepts at most one infinite execution for all* $(q_0, x_0) \in \mathrm{Init}$.

See [15] for some results on existence and uniqueness of executions.

Definition 5 (Reachable State)
A state $(\widehat{q}, \widehat{x}) \in \mathbf{Q} \times \mathbf{X}$ *is called reachable by* H, *if there exists a finite execution* $\chi = (\tau, q, x)$ *with* $\tau = \{[\tau_i, \tau_i']\}_{i=0}^N$ *and* $(q(\tau_N'), x(\tau_N')) = (\widehat{q}, \widehat{x})$.

The set of states reachable by H is denoted $\text{Reach}(H) \subseteq \mathbf{Q} \times \mathbf{X}$.

Definition 6 (Analytic Automaton)
A hybrid automaton is called analytic if for all $q \in \mathbf{Q}$, $f(q,x)$ is analytic as a function of x and $I(q) = \{x \in \mathbf{X} : \sigma(q,x) \geq 0\}$ for some $\sigma(q,x)$ also analytic in x.

A hybrid automaton is said to *preserve its invariants* if $x(t) \in \overline{I(q(t))}$ for all executions $\chi = (\tau, q, x)$ and for all $t \in \tau$.

Proposition 1
A hybrid automaton preserves its invariants if

- $x_0 \in \overline{I(q_0)}$ *for all* $(q_0, x_0) \in \text{Init}$*; and*
- $R(q, q', x) \subseteq \overline{I(q')}$ *for all* $(q, q') \in E$ *and* $x \in G(q, q') \cap \overline{I(q)}$.

Proof: By the first condition, the continuous state starts in the closure of the corresponding invariant. By the definition of an execution, the continuous state cannot leave the closure of the invariant along continuous evolution. By the second condition, the continuous state can not leave the closure of the invariant as a result of a discrete transition. The claim follows by induction along the length of the executions. ■

3 Zeno

Three examples are introduced in this section to illustrate various aspects of Zeno hybrid automata. Some properties of Zeno are studied in next section.

Definition 7 (Zeno Hybrid Automaton)
An infinite execution is called Zeno if $\sum_{i=0}^{\infty}(\tau_i' - \tau_i)$ is bounded. The time $\tau_\infty = \sum_{i=0}^{\infty}(\tau_i' - \tau_i)$ is the Zeno time. If all executions are infinite and Zeno for some initial state, then the automaton is called a Zeno hybrid automaton.

Example 1 (Chattering System)
Consider the hybrid automaton, H, defined by

- $\mathbf{Q} = \{q_1, q_2\}$ and $\mathbf{X} = \mathbb{R}$;
- $\text{Init} = \mathbf{Q} \times \mathbf{X}$;
- $f(q_1, x) = -1$ and $f(q_2, x) = 1$;
- $I(q_1) = \{x \in \mathbf{X} : x \geq 0\}$ and $I(q_2) = \{x \in \mathbf{X} : x \leq 0\}$;
- $E = \{(q_1, q_2), (q_2, q_1)\}$;
- $G(q_1, q_2) = \{x \in \mathbf{X} : x \leq 0\}$ and $G(q_2, q_1) = \{x \in \mathbf{X} : x \geq 0\}$; and
- $R(q_1, q_2, x) = R(q_2, q_1, x) = \{x\}$.

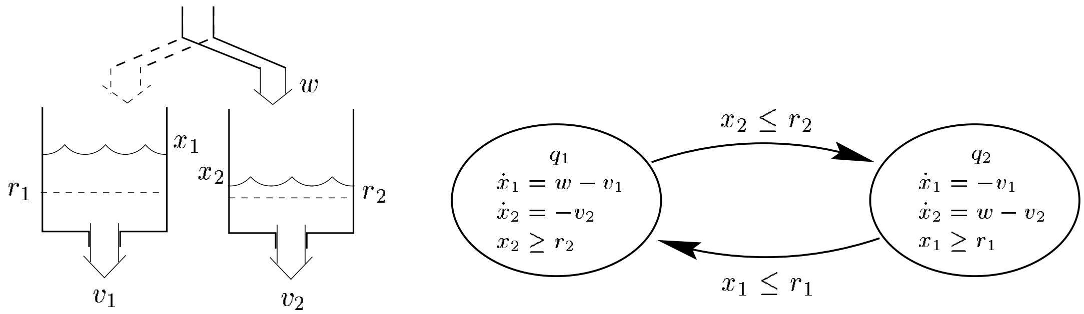

Figure 1: Water tank system and the corresponding hybrid automaton.

One can show that this automaton is non-blocking and deterministic, and therefore accepts a unique infinite execution for all initial states. It is also easy to show that the executions are Zeno. In particular, an execution starting in x_0 at τ_0 reaches $x = 0$ in finite time $\tau_0' = \tau_0 + |x_0|$ and takes an infinite number of transitions from then on, without any time progress. The Zeno time is thus equal to $\tau_\infty = \tau_0'$.

Example 2 (Water Tank System [2])
Consider the water tank system in Figure 1. For $i = 1, 2$, let x_i denote the volume of water in Tank i, and $v_i > 0$ denote the constant flow of water out of Tank i. Let w denote the constant flow of water into the system, dedicated exclusively to either Tank 1 or Tank 2 at each point in time. The control task is to keep the water volumes above r_1 and r_2, respectively, assuming that $x_1(0) > r_1$ and $x_2(0) > r_2$. This is to be achieved by a switched control strategy that switches the inflow to Tank 1 whenever $x_1 \leq r_1$ and to Tank 2 whenever $x_2 \leq r_2$. More formally, the water tank automaton is a hybrid automaton with

- $\mathbf{Q} = \{q_1, q_2\}$ and $\mathbf{X} = \mathbb{R}^2$;
- $\text{Init} = \mathbf{Q} \times \{x \in \mathbf{X} : (x_1 > r_1) \wedge (x_2 > r_2)\}$, $r_1, r_2 > 0$;
- $f(q_1, x) = (w - v_1, -v_2)^T$ and $f(q_2, x) = (-v_1, w - v_2)^T$, $v_1, v_2, w > 0$;
- $I(q_1) = \{x \in \mathbf{X} : x_2 \geq r_2\}$ and $I(q_2) = \{x \in \mathbf{X} : x_1 \geq r_1\}$;
- $E = \{(q_1, q_2), (q_2, q_1)\}$;
- $G(q_1, q_2) = \{x \in \mathbf{X} : x_2 \leq r_2\}$ and $G(q_2, q_1) = \{x \in \mathbf{X} : x_1 \leq r_1\}$; and
- $R(q_1, q_2, x) = R(q_2, q_1, x) = \{x\}$.

One can show that the water tank automaton is non-blocking and deterministic, and therefore accepts a unique infinite execution for each initial state. This execution is Zeno if $w < v_1 + v_2$. Figure 2 shows a simulation of the system with $r_1 = r_2 = 1$, $v_1 = 2$, $v_2 = 3$, and $w = 4$. The Zeno time is equal to $\tau_\infty = 2$.

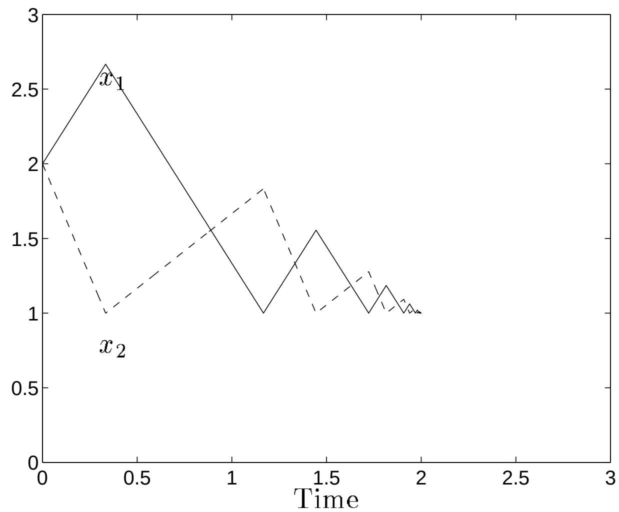

Figure 2: Simulation of Zeno water tank system. The tank levels given by the continuous states x_1 (solid) and x_2 (dashed) are shown. The parameters for the system is $r_1 = r_2 = 1$, $v_1 = 2$, $v_2 = 3$, and $w = 4$. The simulation get stuck at the Zeno time $\tau_\infty = 2$.

Example 3 (Ball Bouncing on a Staircase)
Consider a ball bouncing down on a N step staircase. Assume step $k = 1, \ldots, N$ has width $w_k > 0$ and height $h_k > 0$, and define $\widehat{w}_m = \sum_{k=1}^{m} w_k$ and $\widehat{h}_m = \sum_{k=m}^{N} h_k$. Furthermore, assume that the ball loses a proportional amount of its vertical velocity each bounce and that the ball has constant horizontal speed. A hybrid automaton model for this system is

- $\mathbf{Q} = \{q_1, \ldots, q_{N+1}\}$ and $\mathbf{X} = \mathbb{R}^3$;
- $\mathrm{Init} = \{q \in \mathbf{Q} : q = q_\ell\} \times \{x \in \mathbf{X} : x_1 > \widehat{h}_{N-\ell+1} \ \wedge \ x_3 \in (\widehat{w}_{\ell-1}, \widehat{w}_\ell)\}$ for some $\ell = 1 \ldots, N+1$;
- $f(q_k, x) = (x_2, -g, v)^T$ for $k = 1, \ldots, N+1$, $g, v > 0$;
- $I(q_k) = \{x \in \mathbf{X} : x_1 \geq \widehat{h}_k \ \wedge \ x_3 \leq \widehat{w}_k\}$;
- $E = \{(q_k, q_k)\}_{k=1}^{N+1} \cup \{(q_k, q_{k+1})\}_{k=1}^{N}$;
- $G(q_k, q_k) = \{x \in \mathbf{X} : (x_1 < \widehat{h}_k) \ \vee \ (x_1 = \widehat{h}_k \ \wedge \ x_2 \leq 0 \ \wedge \ x_3 < \widehat{w}_k)\}$ for $k = 1, \ldots, N+1$ and $G(q_k, q_{k+1}) = \{x \in \mathbf{X} : x_3 \geq \widehat{w}_k\}$ for $k = 1, \ldots, N-1$; and
- $R(q_k, q_k, x) = \{(x_1, -cx_2, x_3)^T\}$ for $k = 1, \ldots, N+1$ and $R(q_k, q_{k+1}, x) = \{x\}$ for $k = 1, \ldots, N-1$.

This automaton is non-blocking and deterministic. It can also be shown to be Zeno, if $c \in [0, 1)$. The Zeno time depends on the dimensions of the steps. Figure 3 shows an example with $N = 4$ steps. The dimensions of the steps are given by $w_k = 1$ and $h_k = k$ for $k = 1, \ldots, 4$. The horizontal velocity is $v = 1$. The steps are sufficiently narrow to let the automaton accept Zeno executions with infinitely many discrete transitions only in the last state.

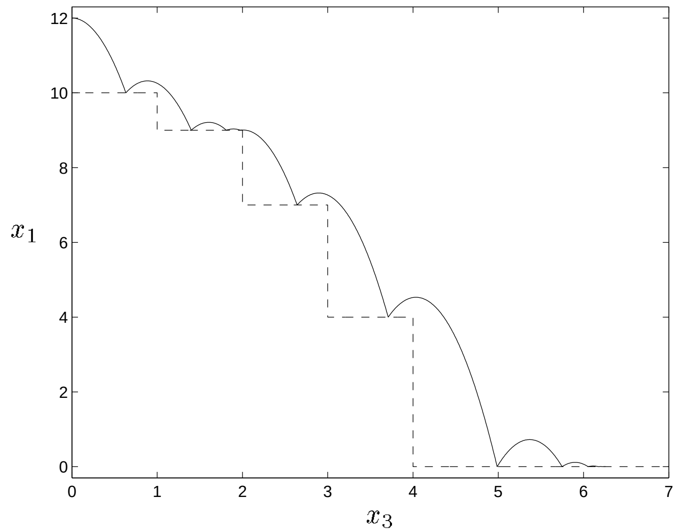

Figure 3: Simulation of hybrid automaton describing a simplistic model for a ball bouncing down a staircase. The position of the ball is given by the solid line (x_1 versus x_3). For the particular choice of parameters and initial state, the execution is Zeno with Zeno time $\tau_\infty \approx 6.2$. The Zeno property of the system will for other dimensions of the steps lead to that the simulation stops at a step.

Examples 1–3 illustrate different aspects of Zeno executions. The first example is an instance of what will later be called Filippov systems. For this class of systems an infinite number of transitions takes place at the same point in time, and, in some cases, sliding mode methods can be used to extend the execution of the system beyond the Zeno time. The second example illustrates the analysis problems that can arise due to Zeno executions. One can easily show that along all executions of the automaton the water level in the two tanks will never go bellow r_1 and r_2, respectively, even if $w < v_1 + v_2$. This is clearly not the case for the real system. Finally, the bouncing ball example illustrates the fact that if a Zeno system is simulated straightforwardly by integrating the vector field in each state and accepting each discrete transition, then the simulation will not pass the Zeno time. Such a simulation is in many cases not acceptable, because it does not give any idea about the dynamical behavior of the real system beyond the Zeno time. For the bouncing ball example, if the first step of the staircase is wide enough for the zeno time to be reached without falling off the edge, the simulation may never reveal the fact that the ball will eventually fall to the next step and will start bouncing again.

4 Properties of Zeno Executions

In this section we present some particular features of Zeno executions. First we introduce the notion of Zeno state.

Definition 8 (Zeno State)
The Zeno state $Z_\infty \subset \mathbf{Q} \times \mathbf{X}$ of a Zeno execution $\chi = (\tau, q, x)$ consists of all states $(\widehat{q}, \widehat{x}) \in \mathbf{Q} \times \mathbf{X}$ such that for all $N > 0$ and all $\epsilon > 0$, there exist $i > N$ with $q(\tau_i) = \widehat{q}$ and $\|x(\tau_i') - \widehat{x}\| < \epsilon$.

The discrete part of the Zeno state consists of the states that are visited infinitely often. For the chattering system, the Zeno state is $Z_\infty = \{q_1, q_2\} \times \{0\}$. For the water tank automaton, $Z_\infty = \{q_1, q_2\} \times \{(r_1, r_2)\}$. The Zeno state of the bouncing ball system is equal to $Z_\infty = \{(q_k, \widehat{h}_k)\}$, where $k \in \{1, \ldots, N+1\}$ depends on the dimensions of the steps. For the particular case shown in Figure 3, we have $k = N+1 = 5$ and $Z_\infty = \{(q_5, 0)\}$.

Clearly, the graph representing a Zeno hybrid automaton must have a closed path.

Proposition 2
A hybrid automaton H is Zeno only if (Q, E) is a cyclic graph.

The following result is also straightforward to prove.

Proposition 3
If there exists a finite collection of states $\{(q_i, x_i)\}_{i=1}^N$ such that

- $(q_1, x_1) = (q_N, x_N)$;
- $(q_i, x_i) \in \mathrm{Reach}(H)$ *for some* $i = 1, \ldots, N$; *and*
- $(q_i, q_{i+1}) \in E$, $x_i \in G(q_i, q_{i+1})$, *and* $x_{i+1} \in R(q_i, q_{i+1}, x_i)$ *for all* $i = 1, \ldots, N$;

then there exists a Zeno execution with $Z_\infty = \{(q_i, x_i)\}_{i=1}^N$.

The following two propositions describes the structure of the Zeno state for two classes of Zeno hybrid automata.

Proposition 4
Consider a Zeno hybrid automaton with $f(q, x)$ bounded and $R(q, q', x) = \{x\}$ for all $(q, q') \in E$. For every Zeno execution, $\chi = (\tau, q, x)$, it holds that $Z_\infty = Q_\infty \times \{\widehat{x}\}$ for some $Q_\infty \subseteq \mathbf{Q}$ and $\widehat{x} \in \mathbf{X}$. If in addition the automaton is invariant preserving and $G(q, q') \subset \big(I(q)^o\big)^c$ for all $(q, q') \in E$, then $\widehat{x} \in \partial I(q)$ for some $q \in Q_\infty$.

Proof: Consider a Zeno execution (τ, q, x) and assume $\|f(q, x)\| < K$ for all q and x. Because $f(q, \cdot)$ is continuous and bounded and $x(\tau_{i+1}) = R\big(q, q', x(\tau_i')\big) = x(\tau_i')$, we have

$$
\begin{aligned}
x(\tau_{i+1}) &= x(\tau_i') \\
&= x(\tau_i) + \int_{\tau_i}^{\tau_i'} f\big(q(\tau_i), x(t)\big) dt \\
&= x(\tau_i) + (\tau_i' - \tau_i) f\big(q(\tau_i), x(\theta_i)\big),
\end{aligned}
$$

for some $\theta_i \in (\tau_i, \tau_i')$. Hence, for all $k > \ell \geq 0$,

$$
x(\tau_k) = x(\tau_\ell) + \sum_{i=\ell}^{k-1} (\tau_i' - \tau_i) f\big(q(\tau_i), x(\theta_i)\big), \quad \theta_i \in (\tau_i, \tau_i'),
$$

which gives that

$$
\|x(\tau_k) - x(\tau_\ell)\| < K \sum_{i=\ell}^{k-1} (\tau_i' - \tau_i).
$$

Since $\sum_{i=0}^{\infty}(\tau_i' - \tau_i) < \infty$, it follows that for all $\epsilon > 0$ there exists $m > 0$ such that $k, \ell > m$ implies that $\|x(\tau_k) - x(\tau_\ell)\| < \epsilon$. Hence, $\{x(\tau_i)\}_{i=0}^{\infty}$ is a Cauchy sequence. The space $\mathbf{X} = \mathbb{R}^n$ is complete, so the sequence has a limit $\widehat{x} = \lim_{i\to\infty} x(\tau_i)$. This concludes the proof of the first part of the proposition.

For the second part, note that if the automaton is invariant preserving $\widehat{x} \in \bigcup_{q\in\mathbf{Q}} \overline{I(q)}$. Assume that $\widehat{x} \notin \partial I(q)$ for any $q \in Q_\infty$. Then, $\widehat{x} \in \bigcap_{q\in Q_\infty} I(q)^o$. This together with existence of $\lim_{i\to\infty} x(\tau_i)$ gives that for every sufficiently small ball $B_\epsilon(\widehat{x})$, there exists $M > 0$ such that $x(\tau_i) \in B_\epsilon(\widehat{x}) \subset I(\widehat{q})^o$ for all $i > M$ and some $\widehat{q} \in Q_\infty$. However, $x(\tau_i') = x(\tau_{i+1})$ and

$$x(\tau_j') \in G\big(q(\tau_j'), q(\tau_{j+1})\big) \subset \big(I(q(\tau_j'))^o\big)^c.$$

By the definition of Zeno state $q(\tau_j) = \widehat{q}$ for some $j > M$. This leads to a contradiction, because there exists i such that $x(\tau_j') = x(\tau_{j+1}) = x(\tau_i) \in I(\widehat{q})^o$ but also $x(\tau_j') \in \big(I(\widehat{q})^o\big)^c$. Hence, $\widehat{x} \in \partial I(q)$ for some $q \in Q_\infty$. ■

Examples 1–2 satisfy the conditions of Proposition 4. As expected the continuous part of the Zeno state is a singleton in both cases. The proposition does not apply to the bouncing ball system, however. Zeno hybrid automaton with non-trivial reset maps can have more involved Zeno state. As an example, consider a Zeno execution with $Z_\infty = \{(\widehat{q}, \widehat{x})\}$. Modify this automaton by extending the continuous state with an extra state $\widetilde{x}$ with trivial continuous dynamics $d\widetilde{x}/dt = 0$, reset map $\widetilde{x} := 4\widetilde{x}(1-\widetilde{x})$, and initial condition $\widetilde{x}(\tau_0) = 0.9$. The iteration of this logistic map takes any value in $(0,1)$ [22]. The Zeno state of the modified automaton is then $\{\widehat{q}\} \times \{(\widehat{x}, \widetilde{x}) : \widetilde{x} \in [0,1]\}$. If the reset map on the other hand is a contraction, then the continuous part of the Zeno state is a singleton as stated next.

Proposition 5
Consider a Zeno execution with $Z_\infty = \{\widehat{q}\} \times X_\infty$ for some $\widehat{q} \in \mathbf{Q}$, $X_\infty \subseteq \mathbf{X}$. If $R(\widehat{q}, \widehat{q}, x)$ is a function and for some $\delta \in (0,1)$

$$\|R(\widehat{q}, \widehat{q}, x) - R(\widehat{q}, \widehat{q}, y)\| \le \delta \|x - y\|, \quad \text{for all } x, y \in G(\widehat{q}, \widehat{q}),$$

then $Z_\infty = \{\widehat{q}\} \times \{\widehat{x}\}$ for some $\widehat{x} \in \mathbf{X}$.

Proof: It holds that $x(\tau_i) \in G(\widehat{q}, \widehat{q})$ for all sufficiently large $i > 0$, since $\widehat{q}$ is the only discrete state defining Z_∞. Note that

$$x(\tau_i') = x(\tau_i) + \int_{\tau_i}^{\tau_i'} f\big(\widehat{q}, x(t)\big) dt$$

and $x(\tau_{i+1}) = r\big(x(\tau_i')\big)$ for some function r satisfying the contraction assumption. Therefore, $\|x(\tau_i') - x(\tau_i)\| = O(\tau_i' - \tau_i)$ and thus, since r is continuous,

$$x(\tau_{i+1}) = r\big(x(\tau_i)\big) + O(\tau_i' - \tau_i).$$

From the assumption it follows that $\|r(x) - r(y)\| < \delta\|x - y\|$ for all $x, y \in G(\widehat{q}, \widehat{q})$. By choosing i sufficiently large, $x(\tau_i) \mapsto x(\tau_{i+1})$ describes a contraction with δ replaced by $\widehat{\delta} \in (\delta, 1)$. Therefore it follows from the contraction mapping theorem that $x(\tau_i)$ converges to a unique point. ■

Proposition 5 indicates that all executions of the bouncing ball automaton in Example 3 with $c \in [0,1)$ will have a Zeno state consisting of only one element.

5 Extension of Zeno Executions

Imagine trying to simulate a hybrid automaton along a Zeno execution. At some point the fast switching is bound to stall the simulation, provided the simulation is accurate enough. One can infer the occurrence of this phenomenon, either off-line by theoretical analysis or on-line by detecting the increasing switching rate, and choose to stop the simulation at some time instant close to the Zeno time τ_∞. The question then becomes, can one continue the simulation beyond τ_∞, in a way that is consistent with the dynamics of the hybrid automaton. Such continuations of a Zeno execution is discussed in this section. Three methods based on regularization, averaging, and Filippov solutions,respectively, are described.

To allow us to capture all the possible continuations of a Zeno execution beyond the Zeno time, we introduce the notion of an *extension*. An extension is a family of hybrid automata, each one of which can be used to simulate a possible continuation for the Zeno execution. For automata with trivial reset relations ($R(q, q', x) = \{x\}$) this notion can be formalized as follows.

Definition 9 (Extension of Zeno execution)
Consider a hybrid automaton H with a Zeno execution $\chi = (\tau, q, x)$ having Zeno state $Z_\infty = Q_\infty \times \{\widehat{x}\}$. An extension *of χ is a collection $H_\chi = (Q_\chi, X_\chi, \mathrm{Init}_\chi, F_\chi, I_\chi, E_\chi, G_\chi, R_\chi)$ with*

- $\mathbf{Q}_\chi = (\mathbf{Q} \setminus Q_\infty) \cup q_\infty$ *and* $X_\chi = X$;
- $\mathrm{Init}_\chi = \{(q_\infty, \widehat{x})\} \subseteq \mathbf{Q}_\chi \times \mathbf{X}_\chi$;
- $F : \mathbf{Q}_\chi \times \mathbf{X}_\chi \to 2^{T\mathbf{X}_\chi}$ *defines a differential inclusion with*

$$F(q, x) = \begin{cases} f(q, x), & \text{if } q \neq q_\infty \\ \mathrm{co}\{\bigcup_{q \in Q_\infty} f(q, x)\}, & \text{if } q = q_\infty; \end{cases}$$

- $I_\chi : \mathbf{Q}_\chi \to 2^{\mathbf{X}_\chi}$ *with* $I_\chi(q) = I(q)$ *if* $q \neq q_\infty$ *and* $I(q_\infty) = \bigcap_{q \in Q_\infty} \overline{I(q)}$;
- $E_\chi \subset \mathbf{Q}_\chi \times \mathbf{Q}_\chi$ *with* $e \in E_\chi$ *if and only if*
  - $e \in E$,
  - $e = (q, q_\infty)$ *with* $q \neq q_\infty$ *and there exists* $q' \in Q_\infty$ *such that* $(q, q') \in E$, *or*
  - $e = (q_\infty, q)$ *with* $q \neq q_\infty$ *and there exists* $q' \in Q_\infty$ *such that* $(q', q) \in E$;
- $G_\chi : E_\chi \to 2^{\mathbf{X}_\chi}$ *with*
  - $G_\chi(q, q') = G(q, q')$ *if* $(q, q') \in E$,
  - $G_\chi(q, q_\infty) = \bigcup_{q' \in Q_\infty} G(q, q')$
  - $G_\chi(q_\infty, q) = \bigcup_{q' \in Q_\infty} G(q', q)$; *and,*
- $R_\chi : E_\chi \times \mathbf{X}_\chi \to 2^{\mathbf{X}_\chi}$ *with* $R_\chi(e, x) = \{x\}$.

An extension of a Zeno is a family of hybrid automata, with the discrete part of the Zeno state collected in one state denoted q_∞, see Figure 4, and continuous dynamics in q_∞ given by a differential inclusion consisting of the convex hull of all vector fields in the Zeno state. A *continuation* of the Zeno

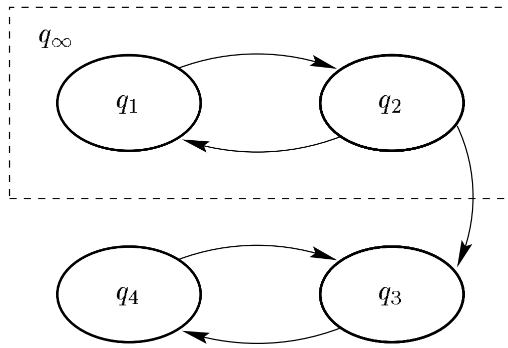

Figure 4: Zeno hybrid automaton with discrete part of the Zeno state Z_∞ equal to $\{q_1, q_2\}$. A Zeno execution can be continued by introducing a discrete state q_∞ with vector field belonging to the convex hull of the vector fields in q_1 and q_2.

execution χ is any execution accepted by a hybrid automaton in the extension H_χ. The intuition is that since the Zeno phenomena are primarily due to modeling simplifications, the extension captures the uncertainty in the model by considering all executions that satisfy a differential inclusion instead of the vector fields of each individual discrete state of the Zeno state for the original Zeno automaton. Note that the reset relations in the discrete Zeno state are neglected in the definition of an extension. It is unclear how to extend this notion to systems with non-trivial reset relations, such as the bouncing ball automaton.

Once the extension has been fixed, the question now becomes how to select an automaton out of the extension in order to simulate beyond the Zeno time. Here we propose three different methods for doing the selection: regularization, averaging, and applying the definition of Filippov solution.

5.1 Continuation by Regularization

Regularization involves modifying the original Zeno automaton by adding regularization parameters that force the executions to be non-Zeno. For example, a temporal regularization that impose a lower bound on the amount of time that elapses between successive discrete transitions gives a non-Zeno automaton. Executions for the regularized automaton are defined beyond the Zeno time for arbitrarily small values of the regularization parameters. As the regularization parameters tend to zero, the regularized automaton should in some sense tend to the original automaton. A continuation of the Zeno execution is then obtained as the limit of the regularized executions as the regularization parameters tend to zero.

As an example of regularization, consider spatial and temporal regularization of the water tank automaton in Example 2. The spatial regularization is defined by introducing a minimum deviation in the continuous state variables between the discrete transitions. A physical interpretation of the regularization is to assume that the measurement devices for x_1 and x_2 are based on floats, which have to move a certain distance corresponding to the volume ϵ to respond. The temporal regularization of the water tank automaton, on the other hand, corresponds to introducing a lower bound $\epsilon > 0$ on the time it takes to change the inflow from Tank 1 to Tank 2, and vice versa. Simulations of these two regularizations are given in Figures 5 and 6, respectively. Both regularizations define continuations that are admissible by the extensions of the corresponding Zeno executions. Note that

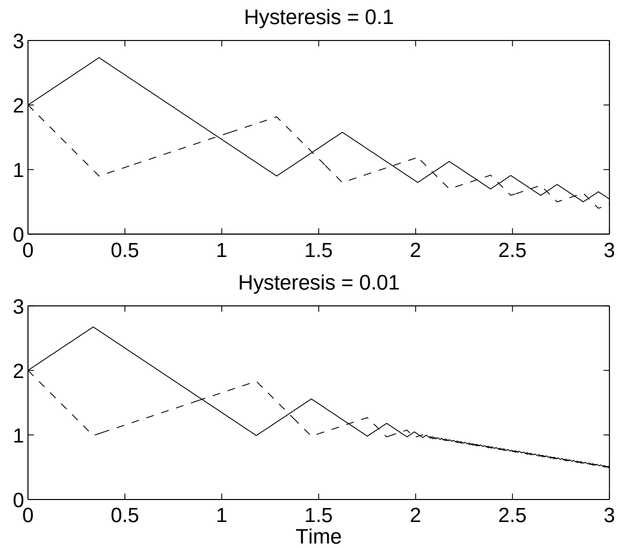

Figure 5: Simulation of spatial regularized water tanks. The upper plot corresponds to hysteresis $\epsilon = 0.1$ and the lower to $\epsilon = 0.01$. The solid line is x_1 and the dashed x_2.

the continuations are different. The spatial regularization suggests $x_1(t) = x_2(t) = 2 - t/2$ for $t > \tau_\infty$, while the temporal regularization suggests $x_1(t) = 1$ and $x_2(t) = 3 - t$ for $t > \tau_\infty$.

Regularization also provides a way of continuing solutions for which an extension (in the sense of the above definition) is not defined. Figures 7 and 8 show simulations for two different regularizations of the bouncing ball automaton: one where each bounce takes a certain amount of time $\epsilon > 0$ and one where an undamped spring with spring constant $1/\epsilon$ is used to model the elasticity of the ball. Note that in this case the two regularizations suggest the same continuation of the execution beyond the Zeno time and that this continuation agrees with the physical intuition.

The lack of uniqueness for suggested continuation based on regularization, as observed for the water tank example, can also be found in systems with Zeno properties as in Example 1. These systems will be denoted Filippov automata and further discussed in the end of this section. Here we only recall the following example for a non-unique regularization limit.

Example 4 (Utkin [24])
Consider a hybrid automaton H given by

- $\mathbf{Q} = \{q_1, q_2, q_3, q_4\}$ and $\mathbf{X} = \mathbb{R}^3$;
- $\text{Init} = \{(q, x) \in \mathbf{Q} \times \mathbf{X} : x \in I(q)\}$;
- $f(q_1, x) = (1, 1, 1)^T$, $f(q_2, x) = (-1, 1, -1)^T$, $f(q_3, x) = (-1, -1, 1)^T$, and $f(q_4, x) = (1, -1, -1)^T$;
- $I(q_1) = \{x \in \mathbf{X} : x_1 \leq 0 \ \wedge \ x_2 \leq 0\}$, $I(q_2) = \{x \in \mathbf{X} : x_1 \geq 0 \ \wedge \ x_2 \leq 0\}$, $I(q_3) = \{x \in \mathbf{X} : x_1 \leq 0 \ \wedge \ x_2 \geq 0\}$, and $I(q_4) = \{x \in \mathbf{X} : x_1 \geq 0 \ \wedge \ x_2 \geq 0\}$;
- $E = \{(q_i, q_i + 1), (q_i + 1, q_i)\}_{i=1}^{3} \cup \{(q_1, q_4), (q_4, q_1)\}$;

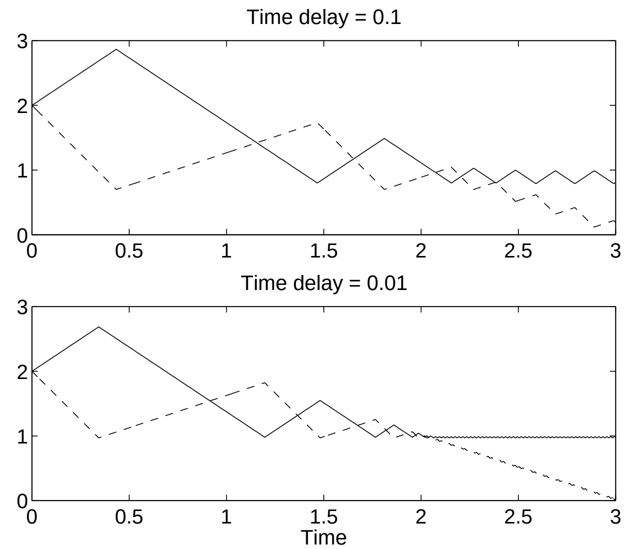

Figure 6: Simulation of temporally regularized water tank automaton. The upper plot corresponds to time delay $\epsilon = 0.1$ and the lower to $\epsilon = 0.01$. The solid line is x_1 and the dashed x_2. Note that this temporal regularization suggests another continuation than the spatial regularization in Figure 5.

- $G(q_1, q_2) = G(q_3, q_4) = \{x \in \mathbf{X} : x_1 \geq 0\}$, $G(q_2, q_1) = G(q_4, q_3) = \{x \in \mathbf{X} : x_1 \leq 0\}$, $G(q_2, q_3) = G(q_1, q_4) = \{x \in \mathbf{X} : x_2 \geq 0\}$, and $G(q_3, q_2) = g(q_4, q_1) = \{x \in \mathbf{X} : x_2 \leq 0\}$;
- $R(q_i, q_j, x) = \{x\}$ for all $i, j = 1, \ldots, 4$ and $i \neq j$.

Let a regularization of H be a hybrid automaton with hysteresis in the discrete transitions. For small values of the hysteresis parameters, the state will end up in the set $\{x = (x_1, x_2, x_3)^T : x_1 \in [-\delta_1, \delta_2] \ \wedge \ x_2 \in [-\delta_3, \delta_4]\}$, where $\delta_i > 0$ are the hysteresis parameters. It follows that $x_1 \to 0$ and $x_2 \to 0$ as $\delta_i \to 0$ for all $i = 1, \ldots, 4$. The limit for x_3 is, however, depending on the way $\delta_1, \ldots, \delta_4$ tend to zero.

5.2 Continuation by Averaging

Extending a Zeno execution $\chi = (\tau, q, x)$ through averaging the vector field close to the Zeno time is an extension of χ in the sense previously defined.

Consider a hybrid automaton with trivial reset maps and bounded vector fields. It follows then by Proposition 4 that the Zeno state is $Z_\infty = Q_\infty \times \{x_\infty\}$, where x_∞ is a singleton. Let $k = |Q_\infty|$ and define

$$\overline{f}(q_\infty, x_\infty) = \lim_{i \to \infty} \frac{1}{\tau_{i+k} - \tau_i} \sum_{\ell=1}^{k} \int_{\tau_{i+\ell}}^{\tau_{i+\ell+1}} f\big(q_\ell, x(t)\big) dt,$$

that is, let the vector field in the state (q_∞, x_∞) be the average of the vector fields as the Zeno time

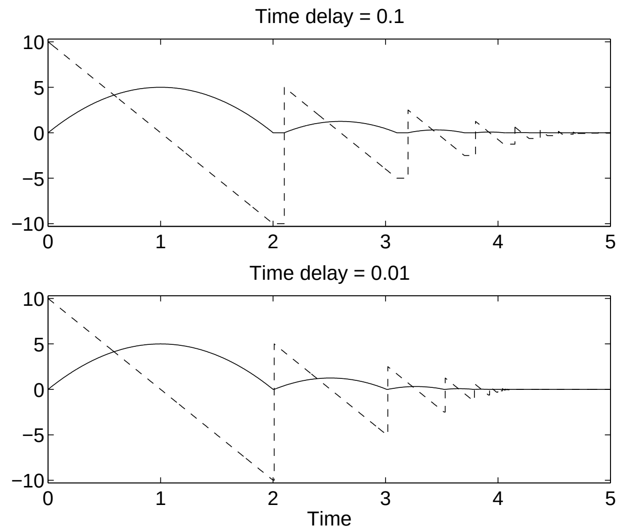

Figure 7: Simulation of bouncing ball with temporal regularization. The upper plot corresponds to time delay $\epsilon = 0.1$ and the lower to $\epsilon = 0.01$. For the non-regularized system, we have the Zeno time at $\tau_\infty = 4$. The executions of the regularized system tends to the execution of the Zeno system as $\epsilon \downarrow 0$ for $t \in [0, \tau_\infty)$. The continuous state tends to zero for t in every closed interval $[\tau_\infty, t_1]$.

is approached. Because f is Lipschitz continuous in x, the expression simplifies to

$$\overline{f}(q_\infty, x_\infty) = \lim_{i\to\infty} \frac{1}{\tau_{i+k} - \tau_i} \sum_{\ell=1}^{k} \left[(\tau_{i+\ell+1} - \tau_{i+\ell}) f\big(q_\ell, x(\tau_{i+\ell})\big) \right].$$

If we let the vector field in q_∞ be equal to $\overline{f}(q_\infty, x_\infty)$ in an open neighborhood of continuous part of the Zeno state, then this form of averaging suggest a continuation of the Zeno execution through integration of the constant vector field $\overline{f}(q_\infty, x_\infty)$. It is easy to check that for the water tank example $\overline{f}(q_\infty, x_\infty) = (1,1)^T(w - v_1 - v_2)/2$. Note that this agrees with the continuation suggested by the spatial regularization illustrated in Figure 5.

The averaging proposed here is not applicable to Example 1 due to that $\tau_{k+1} = \tau_k$ for all $k > 0$. Due to the non-trivial reset maps for the bouncing ball, the averaging is neither applicable to that example.

5.3 Continuation by the Filippov Solution

For special classes of automata, the extension can be restricted in a natural way using the notion of Filippov solution for differential equations with discontinuous right-hand sides [9]. Under certain conditions, this process gives a uniquely defined extension.

Definition 10 (Filippov Automaton)
A Filippov automaton *is an analytic hybrid automaton that satisfies*

- Init $= \{(q, x) \in \mathbf{Q} \times \mathbf{X} : x \in I(q)\}$

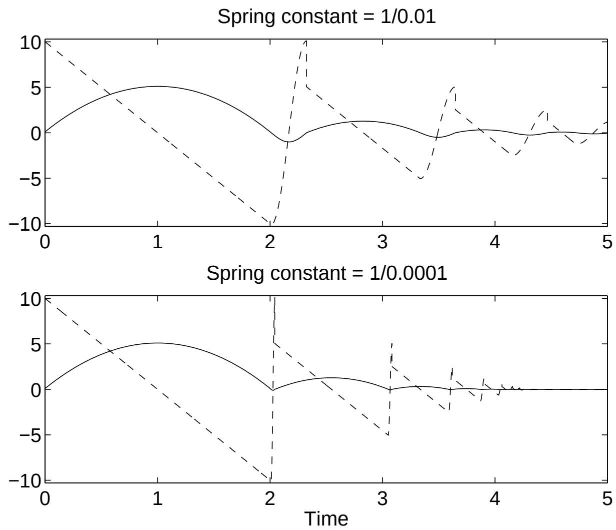

Figure 8: Simulation of bouncing ball with dynamical regularization. The upper plot corresponds to spring constant $1/\epsilon = 1/0.01$ and the lower to $1/\epsilon = 1/0.0001$. As the spring gets stiffer, the execution of the regularized system tends to the execution of the Zeno system for $t < \tau_\infty$. The continuation $x_1(t) = x_2(t) = 0$ suggested for $t > \tau_\infty$ agrees with the one suggested by the temporal regularization in Figure 7.

- *$x \in \partial I(q) \cap \partial I(q')$ for $q, q' \in \mathbf{Q}$ and $L_f\sigma(q,x) < 0$ if and only if $(q,q') \in E$;*
- *$I(q)^o \cap I(q')^o = \emptyset$ for all $(q,q') \in E$;*
- *$L_f\sigma(q,x) \neq 0$ for all $x \in \partial I(q)$ and $q \in \mathbf{Q}$;*
- *$\partial I(q) \cap \partial I(q') \neq \emptyset$ for $(q,q') \in E$ implies the existence of an analytic function $r : \mathbf{X} \to \mathbb{R}$, such that $\{x \in \mathbf{X} : r(x) \geq 0\} = \partial I(q) \cap \partial I(q')$;*
- *$\partial I(q) \cap \partial I(q') \cap \partial I(q'') = \emptyset$ for all different $q, q', q'' \in \mathbf{Q}$;*
- *$G(q,q') = \{x \in I(q) \cap I(q') : L_f\sigma(q,x) \leq 0\}$ for all $(q,q') \in E$; and*
- *$R(q,q',x) = \{x\}$ for all $(q,q') \in E$ and $x \in I(q)$.*

Proposition 6
A Filippov automaton H is non-blocking if for all $q \in \mathbf{Q}$ and $x \in \partial I(q)$ with $L_f\sigma(q,x) < 0$, there exists $q' \in \mathbf{Q}$ such that $x \in I(q')$.

Proof: The assumption gives that for all $q \in \mathbf{Q}$ and

$$x \in \{x \in \mathbf{X} : L_f\sigma(q,x) < 0 \ \wedge \ \sigma(q,x) = 0\}, \tag{1}$$

there exists $q' \in \mathbf{Q}$ with $x \in I(q')$. By the definition of Filippov automaton, $(q,q') \in E$, $x \in G(q,q')$, and $R(q,q',x) = \{x\} \neq \emptyset$. The result now follows by applying Lemma 1 in [15] and noting that the right-hand side of (1) is equal to what is denoted Out in Lemma 1. ∎

Proposition 7
Every Filippov automaton is deterministic.

Proof: We show that the assumptions of Lemma 3 in [15] is fulfilled for all $(q,x) \in \mathbf{Q} \times \mathbf{X}$. The first assumption of Lemma 3 holds because

$$\begin{aligned} G(q,q') &= \{x \in I(q) \cap I(q') : L_f\sigma(q,x) \le 0\} \\ &= \{x \in \mathbf{X} : L_f\sigma(q,x) < 0 \ \wedge \ \sigma(q,x) = 0\}, \end{aligned}$$

for all $(q,q') \in E$. For the second assumption note that $(q,q'), (q,q'') \in E$ imply that $x \in \partial I(q) \cap \partial I(q')$ and $x \in \partial I(q) \cap \partial I(q'')$. But $\partial I(q) \cap \partial I(q') \cap \partial I(q'') = \emptyset$ by the definition of a Filippov automaton, so $x \notin G(q,q') \cap G(q,q'')$ and the second assumption is fulfilled. Finally, the third assumption follows from that $|R(q,q',x)| = |\{x\}| = 1$ for all $(q,q') \in E$ and $x \in I(q)$. ∎

Proposition 8 (Zeno Filippov Automaton)
A Filippov automaton H has a Zeno execution if and only if there exists $x \in \partial I(q) \cap \partial I(q')$ with $(q,x) \in \mathrm{Reach}(H)$ and $(q,q') \in E$, such that $L_f\sigma(q',x) < 0$.

Proof: Sufficiency follows from Proposition 3 with $\{(q_i,x_i)\}_{i=1}^2 = \{(q,x),(q',x)\}$. For necessity, first note that $L_f\sigma(q',x) \neq 0$ by the definition of Filippov automaton. Assume that $L_f\sigma(q',x) > 0$. Since σ is an analytic function in x, there exists a ball $B_\epsilon(x) \subset I(q')$ such that $L_f\sigma(q',x_0) > 0$ for all $x_0 \in B_\epsilon(x)$. Furthermore, from the analyticity of f, there exists $K > 0$ such $\|f(q',x)\| < K$ for all $x \in I(q')$. Now consider the differential equation $dy/dt = f(q',y)$ with $y(0) = x$. It has the solution

$$y(t) = x + \int_0^t f\big(q', y(\theta)\big) d\theta = x + t f\big(q', y(\theta_0)\big), \quad \theta_0 \in (0,t).$$

Hence, $\|y(t) - x\| > \epsilon$ implies that $t > \epsilon/K$. Therefore, if $L_f\sigma(q',x) > 0$ for all q', the execution cannot have infinitely many discrete transitions in a finite time interval. This completes the proof. ∎

Definition 11 (Filippov Extension)
Consider a Zeno Filippov automaton and assume that $L_f\sigma(q,x) \neq L_f\sigma(q',x)$ for all $(q,q') \in E$ such that $L_f\sigma(q',x) < 0$. A Filippov extension *of a Zeno execution $\chi = (\tau, q, x)$ is an extension of χ as in Definition 9, but with the differential inclusion defining the continuous dynamics in q_∞ replaced by*

$$\frac{dy}{dt} = f(q_\infty, x),$$

where

$$f(q_\infty, x) = \alpha(x) f(q,x) + \big(1 - \alpha(x)\big) f(q',x),$$

$q, q' \in Q_\infty$ are the states whose invariants define q_∞, and

$$\alpha(x) = \frac{L_f\sigma(q',x)}{L_f\sigma(q',x) - L_f\sigma(q,x)}.$$

Filippov extension was suggested for simulation of relay systems in [18]. Zeno executions in Filippov automaton are in many cases easy to resolve, because the Zeno detection is simple and it is easy to derive the extension in Definition 11. For some simple examples, the approach has been studied in [17] and tested in simulations [?]. Related ideas of simulating a class of hybrid systems are discussed in [14]. Some Filippov systems are, however, more involved. Regularizations for those may suggest different continuation, so the continuation given by Definition 11 must not be the "correct" one, compare Example 4. Furthermore, Filippov extensions may be non-deterministic. This is illustrated by the following example.

Example 5 (Leonov [13])
Consider the hybrid automaton

- $\mathbf{Q} = \{q_1, q_2, q_3\}$ and $\mathbf{X} = \mathbb{R}^2$;
- $\mathrm{Init} = \{(q, x) \in \mathbf{Q} \times \mathbf{X} : x \in I(q)\}$;
- $f(q_1, x) = (x_2, -\sin x_1)^T$, $f(q_2, x) = (x_2, -\sin x_1 - \gamma_1)^T$, and $f(q_3, x) = (x_2, -\sin x_1 + \gamma_2)^T$;
- $I(q_1) = \{x \in \mathbf{X} : x_2 \leq 2\}$, $I(q_2) = \{x \in \mathbf{X} : (2 - x_2)\sin x_1 \leq 0\}$, and $I(q_3) = \{x \in \mathbf{X} : (2 - x_2)\sin x_1 \geq 0\}$;
- $E = \{(q_i, q_j)\}_{i \neq j}$ for all $i, j = 1, 2, 3$;
- $G(q_1, q_2) = \{x \in \mathbf{X} : \sin x_1 \leq 0 \ \wedge \ x_2 \leq 2\}$, $G(q_2, q_1) = \{x \in \mathbf{X} : x_2 \geq 2\}$, $G(q_2, q_3) = \{x \in \mathbf{X} : \sin x_1 \geq 0\}$, $G(q_3, q_2) = \{x \in \mathbf{X} : \sin x_1 \leq 0\}$, $G(q_3, q_1) = \{x \in \mathbf{X} : x_2 \leq 2\}$, and $G(q_1, q_3) = \{x \in \mathbf{X} : \sin x_1 > 0 \ \wedge \ x_2 \geq 2\}$; and
- $R(q_i, q_j, x) = \{x\}$ for all $i, j = 1, 2, 3$ and $i \neq j$.

For each initial point in $\{(x_1, x_2) \in \mathbb{R}^2 : x_1 \in \big(2k\pi, (2k+1)\pi\big), k \in \mathbb{N}, x_2 = 2\}$, the hybrid automaton accepts three different and arbitrarily short executions.

6 Conclusions

The simulation of hybrid systems poses a number of difficult theoretical and computational problems, not encountered in conventional continuous systems. In this paper, we have shown how Zeno executions may arise in hybrid systems. Zeno phenomena appear to be convenient abstract models of physical systems. Optimal control systems, variable structure systems, and switched systems are examples where they arise in the control of systems. A major motivation for our research is to increase the efficiency of simulation tools for hybrid systems. In particular, we are interested in developing methods to automatically detect Zeno hybrid automata and to extend the simulation of the automaton beyond the Zeno time. One way of doing this is by using graph theoretical algorithms taking into account the guards and invariants associated with the discrete locations, for example, see [17].

We illustrated the Zeno phenomena through some simple physical examples. It might look like these examples suggest that Zenoness can easily be avoided by the introduction of slightly more involved models. In complex systems it is not obvious that this is the case. Zeno may in these systems arise due to interconnection of non-Zeno subsystems. Future work include further classification of

Zeno hybrid automata, together with implementation of the Zeno continuations described in this paper.

References

[1] R Alur and D L Dill. Automata for modeling real-time systems. In *Proceedings of ICALP '90*, Lecture Notes in Computer Science 443, pages 322–335. Springer-Verlag, 1990.

[2] R Alur and T A Henzinger. Modularity for timed and hybrid systems. In A. Mazurkiewicz and J. Winkowski, editors, *CONCUR 97: Concurrency Theory*, Lecture Notes in Computer Science 1243, pages 74–88. Springer-Verlag, 1997.

[3] M Andersson. *Object-Oriented Modeling and Simulation of Hybrid Systems.* PhD thesis, December 1994.

[4] B Bérard, P Gastin, and A Petit. On the power of non observable actions in timed automata. In *Actes du STACS '96*, Lecture Notes in Computer Science 1046, pages 257–268. Springer-Verlag, 1996.

[5] M di Bernardo, C J Budd, and A R Champneys. Grazing, skipping and sliding: analysis of the non-smooth dynamics of the dc/dc buck converter. *Nonlinearity*, 11(4):858–890, jul 1998.

[6] M Branicky, V Borkar, and S Mitter. A unified framework for hybrid control. In *IEEE Conference on Decision and Control*, pages 4228–4234, 1994.

[7] A Deshpande, A Göllü, and L Semenzato. The SHIFT programming language for dynamic networks fo hybrid automata. *IEEE Transactions on Automatic Control*, 43(4):584–587, April 1998.

[8] H Elmqvist. *Dymola—Dynamic Modeling Language, User's Manual.* Dynasim AB, Sweden, 1994.

[9] A F Filippov. *Differential Equations with Discontinuous Righthand Sides.* Kluwer Academic Publishers, 1988.

[10] T A Henzinger. The theory of hybrid automata. In *Proceedings of the 11th Annual Symposium on Logic in Computer Science*, pages 278–292. IEEE Computer Society Press, 1996.

[11] K H Johansson, A Barabanov, and K J Åström. Limit cycles with chattering in relay feedback systems. In *Proc. 36th IEEE Conference on Decision and Control*, San Diego, CA, 1997.

[12] K H Johansson, M Egerstedt, J Lygeros, and S Sastry. Regularization of Zeno hybrid automata. Submitted to Systems Control Letters, 1998.

[13] G A Leonov. Pendulum with positive and negative dry friction. Continuum of homoclinic orbits. *International Journal of Bifurcation and Chaos*, 5(1):251–254, 1995.

[14] Y J Lootsma, A J van der Schaft, and M K Camlibel. Uniqueness of solutions or relay systems. Technical Report 1406, Faculty of Mathematical Sciences, University of Twente, the Netherlands, 1997.

[15] J Lygeros, K H Johansson, and S Sastry M Egerstedt. On the existence of executions of hybrid automata. Submitted to CDC in Phoenix, AZ, 1999.

[16] J Lygeros, C Tomlin, and S Sastry. Controllers for reachability specifications for hybrid systems. *Automatica*, 1999. Accepted for publication, to appear in March 1999.

[17] J Malmborg. *Analysis and Design of Hybrid Control Systems*. PhD thesis, Department of Automatic Control, Lund Institute of Technology, Lund, Sweden, May 1998.

[18] S E Mattsson. On object-oriented modeling of relays and sliding mode behaviour. In *IFAC'96, Preprints 13th World Congress of IFAC*, volume F, pages 259–264, San Francisco, California, July 1996.

[19] S E Mattsson, M Andersson, and K J Åström. Object-oriented modelling and simulation. In D. A. Linkens, editor, *CAD for Control Systems*, chapter 2, pages 31–69. Marcel Dekker Inc, New York, 1993.

[20] P J Mosterman, F Zhao, and G Biswas. Sliding mode model semantics and simulation for hybrid systems. In *Hybrid System V*. Springer-Verlag, 1998.

[21] A B Nordmark. Non-periodic motion caused by grazing incidence in an impact oscillator. *Journal of Sound and Vibration*, 145(2):279–297, 1991.

[22] S S Sastry. *Nonlinear Systems: Analysis, Stability, and Control*. Springer-Verlag, New York, 1999.

[23] *Special Issue on Hybrid Systems, IEEE Transactions on Automatic Control*, 43(4), April 1998.

[24] V I Utkin. Equations of the slipping regime in discontinuous systems. *Automation and Remote Control*, (2):211–219, 1972.

[25] A J van der Schaft and J M Schumacher. Complementarity modeling of hybrid systems. *IEEE Transactions on Automatic Control*, 43(4):483–490, April 1998.