

Simulation of Zeno Hybrid Automata^{1 2}

Karl Henrik Johansson[†], John Lygeros[†], Shankar Sastry[†], and Magnus Egerstedt[‡]

[†]Department of Electrical Engineering and Computer Sciences
University of California, Berkeley, CA 94720-1770, U.S.A.
{johans|lygeros|sastry}@eecs.berkeley.edu

[‡]Division of Optimization and Systems Theory
Royal Institute of Technology, S-100 44 Stockholm, Sweden
magnuse@math.kth.se

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.

1 Introduction

Design evaluation through simulation remains an invaluable tool in hybrid system design, despite recent progress in the development of formal controller synthesis and analysis methods [5, 13]. 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 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 [18, 8, 12] as well as Zeno [1, 9, 4, 2, 10] have only been studied to some extent. The Zeno phenomenon can make hybrid simulation imprecise and time-consuming. Simulation packages, such as Dymola [7], Omola [16], and SHIFT [6], have been developed for hybrid systems. The sim-

ulation problem is generically difficult for these when there is fast switching between the discrete states. The reason for this is that numerical solvers divide the hybrid simulation problem into sequences of regular continuous simulations and instances of solving algebraic equations for the discrete transition times [3]. Therefore the simulation get stuck, when a large number of transitions appear in a short time interval. An extreme case is a Zeno execution.

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 that illustrate several interesting phenomena. The outline of the paper is as follows. In Section 2 the formal notation of hybrid automata is presented. Zeno hybrid automata are introduced in Section 3. Section 4 presents some convergence results for Zeno executions. We propose methods in Section 5 for extending Zeno executions beyond the Zeno time. Some conclusions are given in Section 6.

2 Hybrid Automata

The following definitions are based on [13, 10]. See [12] for notations.

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 set of discrete variables;
- X is a finite set of continuous variables;
- $\text{Init} \subseteq \mathbf{Q} \times \mathbf{X}$ is a set of initial states;
- $f : \mathbf{Q} \times \mathbf{X} \mapsto T\mathbf{X}$ is a vector field;
- $I : \mathbf{Q} \mapsto 2^{\mathbf{X}}$ is an invariant set for each $q \in \mathbf{Q}$;
- $E \subset \mathbf{Q} \times \mathbf{Q}$ is a collection of edges;
- $G : E \mapsto 2^{\mathbf{X}}$ is a guard for each edge; and
- $R : E \times \mathbf{X} \mapsto 2^{\mathbf{X}}$ is a reset map for each edge.

¹The authors would like to thank Jun Zhang for comments on the manuscript. This work was supported by the Swedish Foundation for International Cooperation in Research and Higher Education, Telefonaktiebolaget L.M. Ericsson, ONR under N00014-97-1-0946, DARPA under F33615-98-C-3614, ARO under DAAH04-96-1-0341, and by the Swedish Foundation for Strategic Research through its Centre for Autonomous Systems at KTH.

²IEEE Conference on Decision and Control, Phoenix, AZ, 1999.

We refer to $(q, x) \in \mathbf{Q} \times \mathbf{X}$ as the state of H and assume $\mathbf{X} = \mathbb{R}^n$. 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 $\text{Init}_q = \{x \in \mathbf{X} : (q, x) \in \text{Init}\}$, a vector field $f_q(x) = f(q, x)$, assumed to be globally Lipschitz, and an invariant set $I(q)$. With each edge $e \in E$, a guard $G(e)$ and a reset relation $R(e, x)$ are associated.

Definition 2 (Hybrid Time Trajectory)

A hybrid time trajectory $\tau = \{I_i\}_{i=0}^N$ is a finite or infinite sequence of intervals, such that $I_i = [\tau_i, \tau'_i]$ for $i < N$ and either $I_N = [\tau_N, \tau'_N]$ or $I_N = [\tau_N, \tau'_N)$ if $N < \infty$, where $\tau_i \leq \tau'_i = \tau_{i+1}$.

The interpretation is that the discrete transitions take place at τ_i and the continuous evolution on non-vanishing intervals $[\tau_i, \tau'_i]$. We denote by \mathcal{T} the set of all hybrid time trajectories.

Definition 3 (Execution)

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

- $(q(\tau_0), x(\tau_0)) \in \text{Init}$;
- for all i with $\tau_i < \tau'_i$, $q(\cdot)$ is constant and $x(\cdot)$ is a solution¹ to the differential equation $dx/dt = f(q, x)$ over $[\tau_i, \tau'_i]$, and for all $t \in [\tau_i, \tau'_i)$, $x(t) \in I(q(t))$; and
- for all i , $e = (q(\tau'_i), q(\tau_{i+1})) \in E$, $x(\tau'_i) \in G(e)$, and $x(\tau_{i+1}) \in R(e, x(\tau'_i))$.

We say a hybrid automaton H accepts an execution χ , if H and χ satisfy the conditions in Definition 3. The initial state of χ is denoted $(q_0, x_0) = (q(\tau_0), x(\tau_0))$. An execution is *finite* if τ is a finite sequence ending with a closed interval, it is *infinite* if τ is either an infinite sequence or if $\sum_i (\tau'_i - \tau_i) = \infty$, and it is *maximal* if it is not a strict prefix of any other execution of H [12]. We use $\mathcal{H}_{(q_0, x_0)}$ to denote the set of all executions of H with initial condition $(q_0, x_0) \in \text{Init}$, $\mathcal{H}_{(q_0, x_0)}^M$ the set of all maximal executions, and $\mathcal{H}_{(q_0, x_0)}^\infty$ the set of all infinite executions. A hybrid automaton is called non-blocking if $\mathcal{H}_{(q_0, x_0)}^\infty$ is non-empty for all $(q_0, x_0) \in \text{Init}$, and it is called deterministic if $\mathcal{H}_{(q_0, x_0)}^M$ contains at most one element for all $(q_0, x_0) \in \text{Init}$. See [12] for some results on when a hybrid automaton is non-blocking and deterministic.

Definition 4 (Reachable State)

A state $(\hat{q}, \hat{x}) \in \mathbf{Q} \times \mathbf{X}$ is called *reachable* by H , if there exists a finite execution $\chi = (\tau, q, x) \in \mathcal{H}_{(q_0, x_0)}$ and $(q_0, x_0) \in \text{Init}$ with $\tau = \{[\tau_i, \tau'_i]\}_{i=0}^N$ and $(q(\tau'_N), x(\tau'_N)) = (\hat{q}, \hat{x})$.

¹“Solution” is interpreted in the sense of Caratheodory.

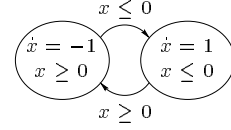


Figure 1: Chattering system automaton in Example 1.

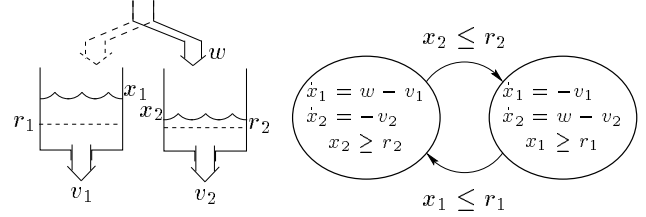


Figure 2: Water tank system and hybrid automaton.

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

3 Zeno Hybrid Automata

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

Definition 5 (Zeno Hybrid Automaton)

An execution in $\mathcal{H}_{(q_0, x_0)}^\infty$ 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*. A hybrid automaton is *Zeno* if all executions in $\mathcal{H}_{(q_0, x_0)}^\infty$ are Zeno for some $(q_0, x_0) \in \text{Init}$.

Example 1 (Chattering System)

Consider the hybrid automaton shown in Figure 1 with $\text{Init} = \mathbf{Q} \times \mathbf{X}$. 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 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 $x_2 \leq r_2$. The hybrid automaton describing this system

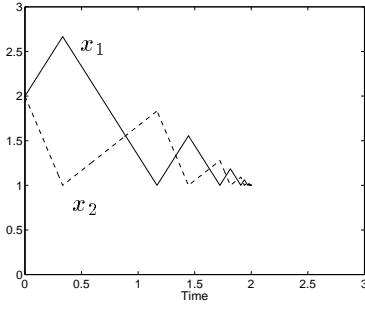


Figure 3: Simulation of Zeno water tank system. The tank levels given by the continuous states x_1 (solid) and x_2 (dashed) are shown. The simulation gets stuck at the Zeno time $\tau_\infty = 2$.

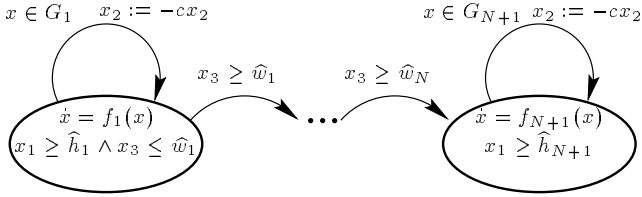


Figure 4: Bouncing ball automaton in Example 3.

is given in Figure 2, where $\text{Init} = \mathbf{Q} \times \{x \in \mathbf{X} : x_1 > r_1 \wedge x_2 > r_2\}$. 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 3 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 $\tau_\infty = 2$.

Example 3 (Ball Bouncing on a Staircase)

Consider a ball bouncing down on a N -step staircase. Assume step $k = 1, \dots, N$ has width $w_k > 0$ and height $h_k > 0$, and define $\hat{w}_m = \sum_{k=1}^m w_k$ with $\hat{w}_{N+1} = \infty$ and $\hat{h}_m = \sum_{k=m}^N h_k$. 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 for this system is shown in Figure 4. Here $f_\ell(x) = (x_2, -g, v)^T$ and $\text{Init} = \{q \in \mathbf{Q} : q = q_\ell\} \times \{x \in \mathbf{X} : x_1 > \hat{h}_{N-\ell+1} \wedge x_3 \in (\hat{w}_{\ell-1}, \hat{w}_\ell)\}$ for $\ell = 1, \dots, N+1$, $G_k = \{(x_1 < \hat{h}_k) \vee (x_1 = \hat{h}_k \wedge x_2 \leq 0 \wedge x_3 \leq \hat{w}_k)\}$ for $k = 1, \dots, N$, and $G_{N+1} = \{(x_1 < 0) \vee (x_1 = 0 \wedge x_2 \leq 0)\}$. 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 5 shows an example with $N = 4$ steps. The dimensions of the steps are given by $w_k = 1$ and $h_k = k$ for $k = 1, \dots, 4$. The other parameters are $g = 10$, $v = 1$, and $c = 0.4$. The steps are sufficiently narrow in this case to let the automaton accept Zeno executions with infinitely many discrete transitions only in the last state.

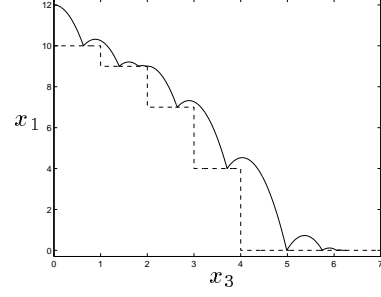


Figure 5: Simulation of hybrid automaton describing a simplistic model for a ball bouncing down a staircase. The position of the ball is given by the solid line (x_1 versus x_3). For this particular choice of parameters and initial state, the execution has Zeno time $\tau_\infty \approx 6.2$.

Examples 1–3 illustrate different aspects of Zeno executions. The first example is an instance of a piecewise smooth system [8]. For this class of systems, an infinite number of transitions typically takes place at the same point in 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 below 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 sequentially integrating the vector field in each state, 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 make two straightforward observations.

Proposition 1

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

Proposition 2

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

- $(q_1, x_1) = (q_N, x_N)$;
- $(q_i, x_i) \in \text{Reach}(H)$ for some $i = 1, \dots, N$; 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 $i = 1, \dots, N$;

then there exists a Zeno execution.

Next, we study convergence of Zeno executions and therefore introduce the notion of Zeno state.

Definition 6 (Zeno State)

A state $(\hat{q}, \hat{x}) \in \mathbf{Q} \times \mathbf{X}$ is called a Zeno state of a Zeno execution $\chi = (\tau, q, x)$, if there exists a sequence $\{\theta_i\}_{i=0}^\infty$, $\theta_i \in [\tau_i, \tau'_i]$, such that for all $N > 0$ and all $\epsilon > 0$, $q(\tau_i) = \hat{q}$ and $\|x(\theta_i) - \hat{x}\| < \epsilon$ for some $i > N$.

The discrete part of the Zeno state consists of a discrete state that is visited infinitely often by a Zeno execution. The set of all Zeno states of a Zeno execution is denoted $Z_\infty \subset \mathbf{Q} \times \mathbf{X}$. For the chattering system in Example 1, we have $Z_\infty = \{q_1, q_2\} \times \{0\}$. For the water tank automaton in Example 2, $Z_\infty = \{q_1, q_2\} \times \{(r_1, r_2)\}$. The Zeno state of the bouncing ball system is $Z_\infty = \{(q_k, \hat{h}_k)\}$, where $k \in \{1, \dots, N + 1\}$ depends on the dimensions of the steps. For the particular case shown in Figure 5, we have $k = N + 1 = 5$ and $Z_\infty = \{(q_5, 0)\}$. The next three propositions describe the structure of the Zeno state for two classes of Zeno hybrid automata. Proofs are given in [19].

Proposition 3

Consider a Zeno hybrid automaton with $R(q, q', x) = \{x\}$ for all $(q, q') \in E$. For every Zeno execution it then holds that $Z_\infty = Q_\infty \times \{\hat{x}\}$ for some $Q_\infty \subseteq \mathbf{Q}$ and $\hat{x} \in \mathbf{X}$.

This result hence give a condition for when the continuous part of the Zeno states is a singleton. Examples 1 and 2 satisfy the conditions of Proposition 3 and 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-identity reset maps can have more involved Zeno state [19]. If the reset map, however, is a contraction, then the continuous part of the Zeno state is a singleton as stated next.

Proposition 4

Consider a Zeno execution with $Z_\infty = \{\hat{q}\} \times X_\infty$ for some $\hat{q} \in \mathbf{Q}$, $X_\infty \subseteq \mathbf{X}$. If $R(\hat{q}, \hat{q}, x)$ is a contracting function, then $Z_\infty = \{\hat{q}\} \times \{\hat{x}\}$ for some $\hat{x} \in \mathbf{X}$.

Proposition 4 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. Note that in Example 3 the only cycles are loops, so by Proposition 1 it follows that if Z_∞ is non-empty it must be of the form $Z_\infty = \{\hat{q}\} \times X_\infty$. If the structure of Z_∞

is not known, but all reset maps are contracting the following result holds.

Proposition 5

Consider a Zeno hybrid automaton with $R(q, q', \cdot)$ a contracting function and $R(q, q', 0) = \{0\}$ for all $(q, q') \in E$. For every Zeno execution it then holds that $Z_\infty = Q_\infty \times \{0\}$ for some $Q_\infty \subseteq \mathbf{Q}$.

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 are discussed in this section. Three methods based on regularization, averaging, and Filippov solutions, respectively, are described.

To allow us to capture possible continuations of a Zeno execution χ 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 of the Zeno execution. We only consider Zeno hybrid automata with identity reset maps $R(\cdot, \cdot, x) = \{x\}$ and Zeno states $Z_\infty = Q_\infty \times \{\hat{x}\}$. The family of hybrid automata is a collection, denoted H_χ , similarly defined as a hybrid automaton, but with discrete states given by $\mathbf{Q} \cup q_\infty \setminus Q_\infty$, $dx/dt = f(q, x)$ replaced by a differential inclusion $dx/dt \in F(q, x)$ with

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

and suitable changes in the definitions of invariant sets, guards, reset maps etc. Figure 6 illustrates how the discrete part of Z_∞ is collected in one state denoted q_∞ . A *continuation* of a Zeno execution χ is now any execution, with initial condition (q_∞, \hat{x}) , 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 should capture 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.

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

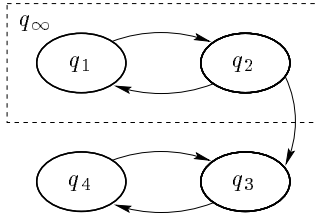


Figure 6: Zeno hybrid automaton with discrete part of the set of Zeno states Z_∞ equal to $\{q_1, q_2\}$. A Zeno execution can be continued by introducing a discrete state q_∞ with vector field belonging to the convex hull of the vector fields in q_1 and q_2 .

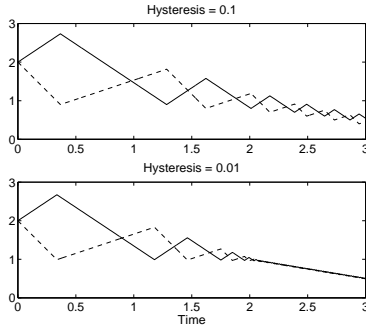


Figure 7: Simulation of spatial regularized water tanks. The upper plot corresponds to hysteresis $\epsilon = 0.1$ and the lower to $\epsilon = 0.01$. The solid line is x_1 and the dashed x_2 .

propose three different methods for doing the selection: using regularization, averaging, or 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, which imposes 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

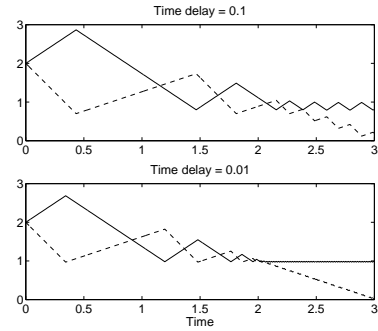


Figure 8: Simulation of temporally regularized water tank automaton. The upper plot corresponds to time delay $\epsilon = 0.1$ and the lower to $\epsilon = 0.01$. The solid line is x_1 and the dashed x_2 . This temporal regularization suggests another continuation than the spatial regularization in Figure 7.

based on floats, which have to move a certain distance corresponding to the volume $\epsilon > 0$ 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 7 and 8, respectively. Both regularizations define continuations that are admissible by the extensions of the corresponding Zeno executions. Note that 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 $t > \tau_\infty$. See [10] for further discussions on regularizations of Zeno hybrid automata.

5.2 Continuation by Averaging

Extending a Zeno execution $\chi = (\tau, q, x)$ through averaging the vector field close to the Zeno time gives a continuation of χ . Assume $Q_\infty = \bigcup_{\ell=1}^k q_\ell$ and define

$$f(q_\infty, \hat{x}) = \lim_{i \rightarrow \infty} \frac{1}{\tau_{i+k} - \tau_i} \sum_{\ell=1}^k \int_{\tau_i + \ell}^{\tau_i + \ell + 1} f(q_\ell, x(t)) dt.$$

This averaged vector field then suggests a continuation of the Zeno execution through integration of $f(q_\infty, \hat{x})$. It is easy to check that for the water tank example $f(q_\infty, \hat{x}) = (1, 1)^T (w - v_1 - v_2)/2$. Note that this agrees with the continuation suggested by the spatial regularization illustrated in Figure 7.

5.3 Continuation by Filippov Solution

For special classes of hybrid automata, the extension can be restricted in a natural way using the notion of Filippov solution [8]. Consider a hybrid automaton with invariants given by disjoint sets filling the state space, guards being the boundaries of these sets with vector fields pointing out of the sets, and iden-

tity reset maps. Such a hybrid automaton describes a piecewise smooth system. Assume that the vector field f is an analytic function in x and that the invariants are given by $I(q) = \{x \in \mathbf{X} : \sigma(q, x) \geq 0\}$, where σ is also an analytic function in x . Consider a Zeno execution with $Z_\infty = \{\hat{q}, \hat{q}'\} \times \{\hat{x}\}$. Then, we define the continuation of the Zeno execution through $dx/dt = f(q_\infty, x)$, with initial state (q_∞, \hat{x}) , $f(q_\infty, x) = \alpha(x)f(\hat{q}, x) + (1 - \alpha(x))f(\hat{q}', x)$, and $\alpha(x) = L_f \sigma(q', x) / [L_f \sigma(q', x) - L_f \sigma(q, x)]$. Such Filippov extension was suggested for simulation of relay systems in [15]. Zeno executions in Filippov automaton are in many cases easy to resolve, because the Zeno detection is relatively simple and it is easy to derive the continuation above. For some simple examples, the approach has been studied [14] and tested in simulations [17]. Related ideas of simulating a class of hybrid systems are discussed in [11].

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. Physical systems are not Zeno. But due to modeling simplification, models of real systems can be Zeno. A 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. We illustrated the Zeno phenomena through some simple physical examples. It might look like they suggest that Zenoness can easily be avoided by introducing slightly more involved models. In complex systems it is, however, not obvious that this is the case. Ongoing work includes further investigations of properties of Zeno hybrid automata, see [19].

References

[1] R. Alur and D. L. Dill. Automata for modeling real-time systems. In *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 *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. Branicky, V. Borkar, and S. Mitter. A unified framework for hybrid control. In *IEEE Conference on Decision and Control*, pages 4228–4234, 1994.

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

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

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

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

[10] K. H. Johansson, M. Egerstedt, J. Lygeros, and S. Sastry. Regularization of Zeno hybrid automata. *Systems & Control Letters*, 1999. To appear.

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

[12] J. Lygeros, K. H. Johansson, S. Sastry, and M. Egerstedt. On the existence of executions of hybrid automata. In *IEEE Conference on Decision and Control*, Phoenix, AZ, 1999.

[13] J. Lygeros, C. Tomlin, and S. Sastry. Controllers for reachability specifications for hybrid systems. *Automatica*, 35(3), March 1999.

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

[15] 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.

[16] 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.

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

[18] 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.

[19] J. Zhang, K. H. Johansson, and J. Lygeros. Properties of Zeno executions. Manuscript in preparation, 1999.