

Copyright © 1999, by the author(s).
All rights reserved.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission.

**ON THE REGULARIZATION OF
ZENO HYBRID AUTOMATA**

by

Karl Henrik Johansson, Magnus Egerstedt, John Lygeros
and Shankar Sastry

Memorandum No. UCB/ERL M99/23

10 January 1999

Cover

**ON THE REGULARIZATION OF
ZENO HYBRID AUTOMATA**

by

Karl Henrik Johansson, Magnus Egerstedt, John Lygeros and Shankar Sastry

Memorandum No. UCB/ERL M99/23

10 January 1999

ELECTRONICS RESEARCH LABORATORY

College of Engineering
University of California, Berkeley
94720

On the Regularization of Zeno Hybrid Automata

Karl Henrik Johansson^a, Magnus Egerstedt^b, John Lygeros^a,
and Shankar Sastry^a

^a*Department of Electrical Engineering and Computer Sciences,
University of California, Berkeley, CA 94720-1770, U.S.A.,
{johans/lygeros/sastry}@eecs.berkeley.edu*

^b*Division of Optimization and Systems Theory, Royal Institute of Technology,
S-100 44 Stockholm, Sweden, magnuse@math.kth.se*

Abstract

Fundamental properties of hybrid automata, such as existence and uniqueness of executions, are studied. Particular attention is devoted to Zeno hybrid automata, which are hybrid automata that take infinitely many discrete transitions in finite time. It is shown that regularization techniques can be used to extend the Zeno executions of these automata to times beyond the Zeno time. Different types of regularization may, however, lead to different extensions. A water tank control problem and a bouncing ball system are used to illustrate the results.

Key words: Hybrid systems; Zeno automata; Regularization; Simulation.

1 Introduction

Despite considerable recent advances in the area of hybrid systems, fundamental issues, such as existence and uniqueness of executions of hybrid automata, are still the topic of intense research activity [19]. To derive local existence and uniqueness conditions for hybrid systems, one needs to consider issues such as blocking and non-determinism associated with the discrete dynamics, in addition to the usual conditions associated with the existence and uniqueness of trajectories for conventional, continuous dynamical systems. Moreover, to ensure that the executions can be extended over arbitrarily long time horizons, one also needs to show that an infinite number of discrete transitions cannot take place in a finite amount of time. Executions that fail to satisfy this property are referred to as Zeno executions, and hybrid automata that accept such executions are referred to as Zeno hybrid automata.¹

¹ The name Zeno refers to the philosopher Zeno of Elea (ca. 500–400 B.C.), whose major work consisted of a number of paradoxes, designed to support his view that the concepts of motion and evolution lead to contradictions. An example is Zeno's Second Paradox of Motion, in which Achilles is racing against a tortoise.

The Zeno phenomenon is fundamentally a hybrid phenomenon, since it requires the interaction of continuous dynamics (in the form of time) and discrete dynamics (in the form of discrete transitions). Even though it seems like a mathematical curiosity, the Zeno phenomenon turns out to be an important consideration when modeling, analyzing, controlling, and simulating hybrid systems. Zeno hybrid automata typically arise due to modeling abstractions, employed by control engineers in an attempt to derive models that are simpler to analyze and control. However, the presence of Zeno executions may cast aspersions on the validity of most techniques typically employed for the analysis of hybrid systems. Most of these techniques (including Lyapunov, model checking, and deductive methods) rely on arguments about the system behavior along an execution. Though mathematically correct, these arguments provide no guarantees about the evolution of the system beyond the limit of the transition times. If this limit is finite (as in the case of Zeno executions) the subsequent evolution may be an important part of the physical process being modeled.

Zeno executions may also arise as the result of certain control policies. Chattering and relaxed controls, common in the optimal control of continuous [20] and hybrid [5] systems, can be intuitively thought of as involving infinitely fast switching among different control actions, and can therefore be modeled by Zeno executions. Similar behavior appears in variable structure control systems [18] and in relay control systems [11]. Perhaps more importantly, Zeno executions may also arise in controllers designed to satisfy reachability specifications. Here, unless special care is taken, the controller may try to prevent the system from reaching an undesirable state by forcing it to take an infinite number of transitions in a finite amount of time [17].

Finally, Zeno type behavior may also affect the efficiency and accuracy of the simulations. Several packages have recently been developed for simulating hybrid dynamical systems, for example, Dymola [8], OmSim [16], SHIFT [7], and a Simulink toolbox [14]. None of these packages, however, makes special provisions for the case of fast switching; as the time intervals between discrete transitions get smaller, either the simulation slows down or its accuracy decreases. In some cases, the simulation may even give erroneous results or error messages.

For the purpose of analysis and controller synthesis, theoretical methods for detecting and eliminating the Zeno phenomenon may be necessary. For the purpose of simulation it may be possible to detect the Zeno phenomenon “on the fly,” and therefore circumvent it by appropriately defining the execution of the system beyond the limit time of the discrete transitions. For certain classes of hybrid systems, in cases when the switching is closely related to sliding modes, this possibility was recently explored in [15,13]. An efficient and accurate simulation method was proposed, making use of the concept of Filippov solutions. Here we propose to extend this approach to more general classes of Zeno hybrid systems.

Despite its importance, the Zeno phenomenon is still not completely understood. Timed automata with Zeno properties have been analyzed to some extent in [1,10,3,2]. For more general hybrid automata, however, subtleties in the continuous dynamics make the analysis more challenging. The main contribution of this paper is to illustrate properties of Zeno hybrid automata through examples, and to propose a method for extending Zeno executions beyond the limit of the transition times, using regularization techniques. Formal definitions of hybrid automata and their executions are given in Section 2. Based on these definitions, results on existence and uniqueness of executions are derived for a special class of hybrid automata, referred to as

automata with transverse invariants. These results are then used in Section 3, where examples of Zeno hybrid automata in this class are analyzed, to highlight the different manifestations of the Zeno phenomenon. Section 4 discusses regularization of Zeno hybrid automata. Using the examples of Section 3, it is shown that different regularizations of a Zeno execution may suggest different extensions. This indicates that, even though regularization may be used to extend Zeno executions beyond the limit of the transition times, additional information about the underlying physical process may be needed to select a meaningful extension.

2 Hybrid Automata

Notation

Consider a finite collection V of variables and let \mathbf{V} denote the set of valuations (possible assignments) 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 countable as *discrete* and to variables whose set of valuations is a subset of a Euclidean space as *continuous*. We assume that Euclidean spaces, \mathbb{R}^n for $n \geq 0$, are given the Euclidean metric topology, whereas countable and finite sets are given the discrete topology (all subsets are open). Subsets of a topological space are given the subset topology and products of topological spaces are given the product topology. For a subset U of a topological space we use 2^U to denote the set of all subsets of U . We use \wedge to denote the logical “and” and \vee to denote “or.”

Hybrid Automata and Executions

The following definitions are based on [12]. A hybrid system will involve continuous evolution as well as instantaneous transitions. To distinguish the times at which discrete transitions take place we introduce the notion of a hybrid time trajectory.

Definition 1 (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

- for all $0 \leq i < N$, $I_i = [\tau_i, \tau'_i]$ with $\tau_i \leq \tau'_i = \tau_{i+1}$;
- if $N < \infty$, either $I_N = [\tau_N, \tau'_N]$ with $\tau_N \leq \tau'_N < \infty$, or $I_N = [\tau_N, \tau'_N)$ with $\tau_N < \tau'_N \leq \infty$.

The interpretation is that τ_i are the times at which discrete transitions take place; notice that multiple transitions may take place at the same time (if $\tau_i = \tau'_i = \tau_{i+1}$). Hybrid time trajectories can extend to infinity either 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. Each $\tau \in \mathcal{T}$ is fully ordered by the relation \prec , which for $t \in [\tau_i, \tau'_i] \in \tau$ and $t' \in [\tau_j, \tau'_j] \in \tau$ is defined as $t \prec t'$ if either $i < j$ or $i = j$ and $t < t'$. For $t \in \mathbb{R}$ and $\tau \in \mathcal{T}$ we use $t \in \tau$ as a shorthand notation for “there exists a j such that $t \in [\tau_j, \tau'_j] \in \tau$ ”. For a topological space K and a $\tau \in \mathcal{T}$, we use $k : \tau \rightarrow K$ as a shorthand notation for a map

assigning values from K to all $t \in \tau$. We say $\tau = \{I_i\}_{i=0}^N \in \mathcal{T}$ is a *prefix* of $\hat{\tau} = \{J_i\}_{i=0}^M \in \mathcal{T}$ and write $\tau \leq \hat{\tau}$ if either they are identical or τ is finite, $M \geq N$, $I_i = J_i$ for all $i = 0, \dots, N-1$, and $I_N \subseteq J_N$. The prefix relation is a partial order on \mathcal{T} .

Definition 2 (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 $\mathbf{X} = \mathbb{R}^n$;
- $\text{Init} \subseteq Q \times X$ is a set of initial states;
- $f : Q \times X \rightarrow TX$ is a vector field, assumed to be Lipschitz continuous in its second argument;
- $I : Q \rightarrow 2^X$ assigns to each $q \in Q$ an invariant set;
- $E \subset Q \times Q$ is a collection of edges;
- $G : E \rightarrow 2^X$ assigns to each edge $e = (q, q') \in E$ a guard; and
- $R : E \times X \rightarrow 2^X$ assigns to each edge $e = (q, q') \in E$ and $x \in X$ a reset relation.

We refer to $(q, x) \in Q \times X$ as the state of H . Pictorially, a hybrid automaton can be represented by a directed graph, with vertices Q and edges E . With each vertex $q \in Q$, we associate a vector field $f(q, x)$ 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 3 (Execution)

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

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

For an execution $\chi = (\tau, q, x)$ we use $(q_0, x_0) = (q(\tau_0), x(\tau_0))$ to denote the initial state of χ . We say $\chi = (\tau, q, x)$ is a prefix of $\chi' = (\tau', q', x')$ (write $\chi \leq \chi'$) if $\tau \leq \tau'$ and $(q(t), x(t)) = (q'(t), x'(t))$ for all $t \in \tau$. We say χ is a strict prefix of χ' (write $\chi < \chi'$) if $\chi \leq \chi'$ and $\chi \neq \chi'$. It is easy to show that the set of executions of a hybrid automaton is prefix closed and partially ordered by the prefix relation.

Unlike conventional continuous dynamical systems, the interpretation is that an automaton H *accepts* an execution $\chi = (\tau, q, x)$ (as opposed to generates). This difference allows one to consider hybrid automata that accept no executions for some initial states, accept multiple executions for the same initial state, or do not accept executions over arbitrarily long time horizons. An execution $\chi = (\tau, q, x)$ is called *finite* if τ is a finite sequence ending with a closed interval, *infinite* if τ is an infinite sequence, or if $\sum_i(\tau'_i - \tau_i) = \infty$, *Zeno* if it is infinite but $\sum_i(\tau'_i - \tau_i) < \infty$, and *maximal* if it is not a strict prefix of any other execution of H . For an infinite execution we define the Zeno time as $\tau_\infty = \sum_i(\tau'_i - \tau_i)$. Clearly, $\tau_\infty < \infty$ if the execution is Zeno and $\tau_\infty = \infty$ otherwise. We use $\mathcal{H}_{(q_0, x_0)}$ to denote the set of all

executions of H with initial condition $(q_0, x_0) \in \text{Init}$, $\mathcal{H}_{(q_0, x_0)}^M$ to denote the corresponding maximal executions, and $\mathcal{H}_{(q_0, x_0)}^\infty$ to denote the infinite executions. For all $(q_0, x_0) \in \text{Init}$, we have $\mathcal{H}_{(q_0, x_0)}^\infty \subseteq \mathcal{H}_{(q_0, x_0)}^M \subseteq \mathcal{H}_{(q_0, x_0)}$, since infinite executions cannot be extended (so they must be maximal) and maximal executions may be blocking (so they need not be infinite). These sets may be empty or may contain multiple executions.

Definition 4 (Non-Blocking and Deterministic Automaton)

A hybrid automaton H is called non-blocking if $\mathcal{H}_{(q_0, x_0)}^\infty$ is non-empty for all $(q_0, x_0) \in \text{Init}$. It is called deterministic if $\mathcal{H}_{(q_0, x_0)}^M$ contains at most one element for all $(q_0, x_0) \in \text{Init}$.

Existence and Uniqueness of Executions

Next, we derive some simple conditions to characterize non-blocking and deterministic automata. We restrict our attention to a special class of hybrid automata, where the vector field is, in a sense, transverse to the boundary of the invariant set. Assume f is analytic in its second argument. For a function $\sigma : \mathbf{Q} \times \mathbf{X} \rightarrow \mathbb{R}$, also analytic in its second argument, recursively define the Lie derivative of σ along f , $L_f^m \sigma : \mathbf{Q} \times \mathbf{X} \rightarrow \mathbb{R}$, by

$$L_f^m \sigma(q, x) = \begin{cases} \sigma(q, x), & m = 0 \\ \left(\frac{\partial}{\partial x} L_f^{m-1} \sigma(q, x) \right) f(q, x), & m > 0. \end{cases}$$

Definition 5 (Transverse Invariants)

A hybrid automaton H is said to have transverse invariants if f is analytic in its second argument and there exists a function $\sigma : \mathbf{Q} \times \mathbf{X} \rightarrow \mathbb{R}$, also analytic in its second argument, such that

- $I(q) = \{x \in \mathbf{X} : \sigma(q, x) \geq 0\}$ for all $q \in \mathbf{Q}$; and
- for all $(q, x) \in \mathbf{Q} \times \mathbf{X}$, there exists a finite $m \in \mathbb{N}$ such that $L_f^m \sigma(q, x) \neq 0$.

For a hybrid automaton with transverse invariants we define pointwise the relative degree as a function $n : \mathbf{Q} \times \mathbf{X} \rightarrow \mathbb{N}$ with

$$n(q, x) := \min \{m \in \mathbb{N} : L_f^m \sigma(q, x) \neq 0\}.$$

For all $q \in \mathbf{Q}$ we also define

$$\text{Out}(q) := \{x \in \mathbf{X} : L_f^{n(q, x)} \sigma(q, x) < 0\}.$$

For each discrete state $q \in \mathbf{Q}$, the set $\text{Out}(q)$ contains the continuous states from which it is impossible to remain in q by continuous evolution.

The following lemma indicates that a hybrid automaton with transverse invariants is non-blocking if transitions with non-empty reset relations are enabled along the boundary of the invariant sets.

Lemma 1

A hybrid automaton H with transverse invariants is non-blocking, if for all $q \in \mathbf{Q}$ and for all $x \in \text{Out}(q)$ there exists $(q, q') \in E$ such that $x \in G(q, q')$ and $R((q, q'), x) \neq \emptyset$.

PROOF. Consider an arbitrary initial state $(q_0, x_0) \in \text{Init}$ and assume, for the sake of contradiction, that there does not exist an infinite execution starting at (q_0, x_0) . Let $\chi = (\tau, q, x)$ denote a maximal execution starting at (q_0, x_0) , and note that τ must be a finite sequence.

First consider the case $\tau = \{[\tau_i, \tau'_i]\}_{i=0}^{N-1}[\tau_N, \tau'_N]$. Let $(q_N, x_N) = \lim_{t \rightarrow \tau'_N} (q(t), x(t))$. Note that, by the definition of execution and a standard existence argument for continuous dynamical systems, χ can be extended to $\hat{\chi} = (\hat{\tau}, \hat{q}, \hat{x})$ with $\hat{\tau} = \{[\tau_i, \tau'_i]\}_{i=0}^N$, $\hat{q}(\tau'_N) = q_N$ and $\hat{x}(\tau'_N) = x_N$. This contradicts the maximality of χ .

Next consider the case $\tau = \{[\tau_i, \tau'_i]\}_{i=0}^N$, and let $(q_N, x_N) = (q(\tau'_N), x(\tau'_N))$. We distinguish two sub-cases. If

$$x_N \in \text{Out}(q_N)^c = \left\{ x \in \mathbf{X} : L_f^{n(q_N, x)} \sigma(q_N, x) > 0 \right\},$$

(where the strict inequality follows from the transverse invariant assumption) then χ can be extended by continuous evolution. This follows by a standard existence argument for continuous dynamical systems: since f and σ are analytic in their second argument, there exists $\epsilon > 0$ such that χ can be extended to $\hat{\chi} = (\hat{\tau}, \hat{q}, \hat{x})$ with $\hat{\tau} = \{[\tau_i, \tau'_i]\}_{i=0}^{N-1}[\tau_N, \tau'_N + \epsilon]$.

If, on the other hand, $x_N \in \text{Out}(q_N)$, then there exists $(q', x') \in \mathbf{Q} \times \mathbf{X}$ such that $(q_N, q') \in E$, $x_N \in G(q_N, q')$ and $x' \in R((q_N, q'), x_N)$. Therefore, χ can be extended by a discrete transition to $\hat{\chi} = (\hat{\tau}, \hat{q}, \hat{x})$ with $\hat{\tau} = \{[\tau_i, \tau'_i]\}_{i=0}^{N+1}$, $\tau_{N+1} = \tau'_{N+1} = \tau_N$, $q(\tau_{N+1}) = q'$, and $x(\tau_{N+1}) = x'$. In both cases the maximality of χ is contradicted. \square

The following lemma indicates that a hybrid automaton is deterministic if (1) discrete transitions are forced by the continuous flow exiting the invariant whenever they are enabled by the corresponding guard, (2) no two discrete transitions are enabled simultaneously, and (3) no point can be mapped onto two different points by the reset map.

Lemma 2

A hybrid automaton H with transverse invariants is deterministic if

- (1) $x \in \bigcup_{(q, q') \in E} G(q, q')$ implies that $x \in \text{Out}(q)$;
- (2) $(q, q') \in E$ and $(q, q'') \in E$ with $q' \neq q''$ imply $G(q, q') \cap G(q, q'') = \emptyset$; and
- (3) $(q, q') \in E$ and $x \in G(q, q')$ imply $|R(q, q', x)| \leq 1$.

PROOF. Assume, for the sake of contradiction, that there exists an initial state $(q_0, x_0) \in \text{Init}$ and two maximal executions $\chi = (\tau, q, x)$ and $\hat{\chi} = (\hat{\tau}, \hat{q}, \hat{x})$ starting at (q_0, x_0) with $\chi \neq \hat{\chi}$. Let $\psi = (\rho, p, y) \in \mathcal{H}_{(q_0, x_0)}$ denote the maximal common prefix of χ and $\hat{\chi}$. Such a prefix exists as the executions start at the same initial state. Note that ψ is not an infinite execution, as $\chi \neq \hat{\chi}$. Therefore, as in the proof of Lemma 1, ρ can be assumed to be of the form $\rho = \{[\rho_i, \rho'_i]\}_{i=0}^N$, as otherwise the maximality of ψ would be contradicted by an existence and uniqueness argument of the continuous solution along f . Let $(q_N, x_N) = (q(\rho'_N), x(\rho'_N)) = (\hat{q}(\rho'_N), \hat{x}(\rho'_N))$. We distinguish the following cases:

Case 1: $\rho'_N \notin \{\tau'_i\}$ and $\rho'_N \notin \{\hat{\tau}'_i\}$, i.e., ρ'_N is not a time when a discrete transition takes place in either χ or $\hat{\chi}$. Then, by the definition of execution and a standard existence and uniqueness argument for continuous dynamical systems, there exists $\epsilon > 0$ such that the prefixes of χ

and $\hat{\chi}$ are defined over $\hat{\rho} = \{[\rho_i, \rho'_i]\}_{i=0}^{N-1}[\rho_N, \rho'_N + \epsilon)$ and are identical. This contradicts the maximality of ψ .

Case 2: $\rho'_N \in \{\tau'_i\}$ and $\rho'_N \notin \{\hat{\tau}'_i\}$, i.e., ρ'_N is a time when a discrete transition takes place in χ but not in $\hat{\chi}$. The fact that a discrete transition takes place at (q_N, x_N) in χ indicates that there exists $q' \in \mathbf{Q}$ such that $(q_N, q') \in E$ and $x_N \in G(q_N, q')$. No discrete transition at (q_N, x_N) in $\hat{\chi}$ indicates that there exists $\epsilon > 0$ such that $\hat{\chi}$ is defined over $\hat{\rho} = \{[\rho_i, \rho'_i]\}_{i=0}^{N-1}[\rho_N, \rho'_N + \epsilon)$. A necessary condition for this is that $x_N \notin \text{Out}(q)$. This contradicts Condition 1 of the lemma.

Case 3: $\rho'_N \notin \{\tau'_i\}$ and $\rho'_N \in \{\hat{\tau}'_i\}$, symmetric to Case 2.

Case 4: $\rho'_N \in \{\tau'_i\}$ and $\rho'_N \in \{\hat{\tau}'_i\}$, i.e., ρ'_N is a time when a discrete transition takes place in both χ and $\hat{\chi}$. A discrete transition at (q_N, x_N) in both χ and $\hat{\chi}$ indicates that there exist (q', x') and (\hat{q}', \hat{x}') such that $(q_N, q') \in E$, $(q_N, \hat{q}') \in E$, $x_N \in G(q_N, q')$, $x_N \in G(q_N, \hat{q}')$, $x' \in R((q_N, q'), x_N)$, and $\hat{x}' \in R((q_N, \hat{q}'), x_N)$. Note that by Condition 2 of the lemma, $q' = \hat{q}'$, hence, by Condition 3, $x' = \hat{x}'$. Therefore, the prefixes of χ and $\hat{\chi}$ are defined over $\hat{\rho} = \{[\rho_i, \rho'_i]\}_{i=0}^N[\rho_{N+1}, \rho'_{N+1}]$, with $\rho_{N+1} = \rho'_{N+1} = \rho'_N$, and are identical. This contradicts the maximality of ψ . \square

Summarizing Lemmas 1 and 2 leads to the following result.

Theorem 1

If a hybrid automaton with transverse invariants satisfies the conditions of Lemma 1 and 2, then it accepts a unique infinite execution for all $(q_0, x_0) \in \text{Init}$.

3 Zeno Hybrid Automata

Definition 6 (Zeno Hybrid Automaton)

A hybrid automaton H is called Zeno if there exists $(q_0, x_0) \in \text{Init}$ such that all executions in $\mathcal{H}_{(q_0, x_0)}^\infty$ are Zeno executions.²

We illustrate the Zeno property through examples: an automaton modeling a water tank system and an automaton modeling a bouncing ball. First, however, a hybrid automaton that does not have transverse invariants is discussed.

² An alternative definition of a Zeno automaton is to require that *at least one* execution in $\mathcal{H}_{(q_0, x_0)}^\infty$ is Zeno. For deterministic hybrid automata, such as the ones discussed in the subsequent examples, the two definitions coincide.

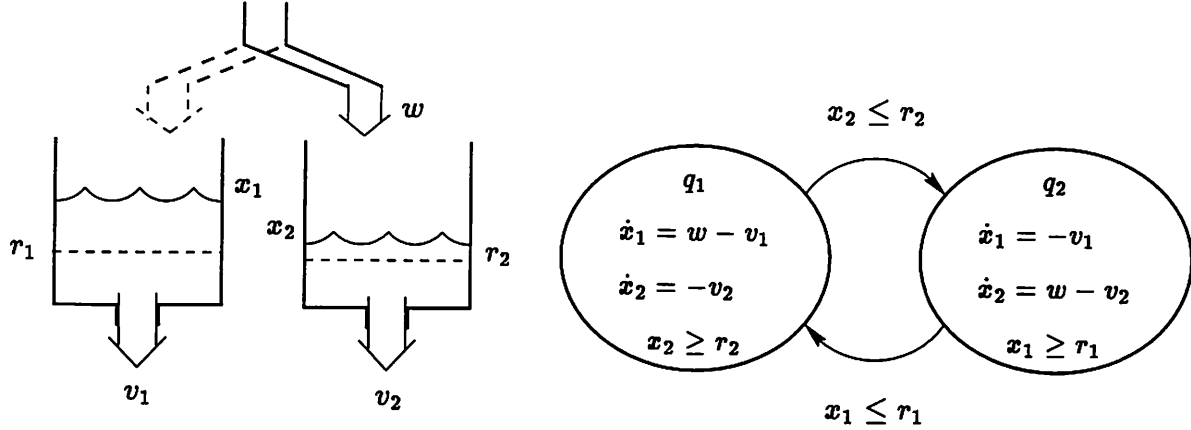


Fig. 1. Water tank system and corresponding hybrid automaton.

Non-Analytic Automaton

Consider the smooth, but non-analytic, function $s : \mathbb{R} \rightarrow \mathbb{R}$ given by

$$s(x) = \begin{cases} e^{-1/x} \sin(1/x) & \text{if } x > 0 \\ 0 & \text{if } x = 0 \\ e^{1/x} \sin(-1/x) & \text{if } x < 0. \end{cases}$$

We define a hybrid automaton, denoted NA , by

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

It is easy to see that the infinite execution of NA with initial state $(q_1, -1)$ exhibits an infinite number of discrete transitions by $\tau_\infty = 1$. The reason is that the non-analytic function s has an infinite number of zeros in the bounded interval $(-1, 0)$.

Water Tank Automaton

Consider the water tank system of [2] shown 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, directed exclusively to either Tank 1 or Tank 2 at each point in time. The objective 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, denoted WT , with

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

We show that the water tank automaton is non-blocking, deterministic, and Zeno.

Proposition 1

The water tank automaton WT accepts a unique infinite execution for each initial state.

PROOF. Let $\sigma(q_1, x) = x_2 - r_2$ and $\sigma(q_2, x) = x_1 - r_1$. Then $L_f^1 \sigma(q_1, x) = -v_2 < 0$ and $L_f^1 \sigma(q_2, x) = -v_1 < 0$. Since both f and σ are analytic functions of x and $I(q_i) = \{x \in \mathbf{X} : \sigma(q_i, x) \geq 0\}$, the water tank automaton has transverse invariants.

Note that $n(q_1, x) = 0$ if $x_2 \neq r_2$, and $n(q_1, x) = 1$ if $x_2 = r_2$, therefore $\text{Out}(q_1) = \{x \in \mathbf{X} : x_2 \leq r_2\} = G(q_1, q_2)$ (and similarly for q_2). This implies that Conditions 1 of both Lemma 1 and Lemma 2 are satisfied. Condition 2 of Lemma 2 is trivially satisfied. Moreover, since $|R((q_1, q_2), x)| = |R((q_2, q_1), x)| = 1$, Condition 2 of Lemma 1 and Condition 3 of Lemma 2 are also satisfied. The proposition follows from Theorem 1. \square

Proposition 2

If $\max\{v_1, v_2\} < w < v_1 + v_2$, then the water tank automaton WT is Zeno. The Zeno time is $\tau_\infty = (x_1(0) + x_2(0) - r_1 - r_2) / (v_1 + v_2 - w)$, where $(x_1(0), x_2(0))$ is the continuous part of the initial state.

PROOF. By straightforward calculation. \square

Bouncing Ball Automaton

Consider a simple model of an elastic ball bouncing on the ground, losing a fraction of its energy with each bounce. Let x_1 denote the altitude of the ball and x_2 its vertical speed. A hybrid automaton, BB , describing this system is shown in Figure 2 and is defined by

- $\mathbf{Q} = \{q\}$ and $\mathbf{X} = \mathbb{R}^2$;
- $\text{Init} = \{q\} \times \{x \in \mathbf{X} : x_1 \geq 0\}$;
- $f(q, x) = (x_2, -g)$ with $g > 0$;
- $I(q) = \{x \in \mathbf{X} : x_1 \geq 0\}$;
- $E = \{(q, q)\}$;
- $G(q, q) = \{x \in \mathbf{X} : [x_1 < 0] \vee [(x_1 = 0) \wedge (x_2 \leq 0)]\}$; and
- $R((q, q), x) = \{(x_1, -x_2/c)\}$ with $c \geq 1$.

We show that also BB is non-blocking, deterministic, and Zeno.

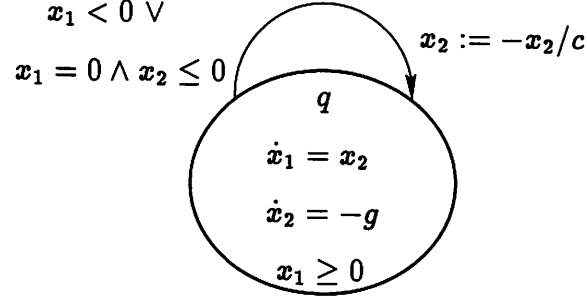


Fig. 2. Hybrid automaton for a simple model of a bouncing ball.

Proposition 3

The bouncing ball automaton BB accepts a unique execution for each initial state.

PROOF. Let $\sigma(q, x) = x_1$. Then $L_f^1 \sigma(q, x) = x_2$ and $L_f^2 \sigma(q, x) = -g \neq 0$. Since both f and σ are analytic functions of x and $I(q) = \{x \in \mathbf{X} : \sigma(q, x) \geq 0\}$, the bouncing ball automaton has transverse invariants.

Note that $n(q, x) = 0$ if $x_1 \neq 0$, $n(q, x) = 1$ if $(x_1 = 0) \wedge (x_2 \neq 0)$, and $n(q, x) = 2$ if $(x_1 = 0) \wedge (x_2 = 0)$. Therefore, $\text{Out}(q) = \{x \in \mathbf{X} : [x_1 < 0] \vee [(x_1 = 0) \wedge (x_2 < 0)] \vee [(x_1 = 0) \wedge (x_2 = 0)]\} = G(q, q)$. This implies that Conditions 1 of both Lemma 1 and Lemma 2 are satisfied. Condition 2 of Lemma 2 is trivially satisfied. Moreover, since $|R((q, q), x)| = 1$, Condition 2 of Lemma 1 and Condition 3 of Lemma 2 are also satisfied. The proposition follows from Theorem 1. \square

Proposition 4

If $c > 1$, the bouncing ball automaton BB is Zeno. The Zeno time is

$$\tau_\infty = \frac{x_2(0)}{g} + \frac{(c+1)\sqrt{x_2(0)^2 + 2gx_1(0)}}{g(c-1)},$$

where $(x_1(0), x_2(0))$ is the continuous part of the initial state.

PROOF. The first bounce occurs at time $\tau_1 = \tau'_0 = (x_2(0) + \sqrt{x_2(0)^2 + 2gx_1(0)})/g$. The next bounce occurs at time $\tau_2 = \tau'_1 = \tau_1 + 2x_2(\tau_1)/g$. More generally, bounce N occurs at time $\tau_N = \tau_1 + \sum_{k=1}^N 2x_2(\tau_k)/(gc^{k-1})$. Since $c > 1$ the series on the right hand side converges to the value of τ_∞ given in the proposition. \square

Discussion

The three examples introduced above have some similar properties but shed light on different aspects of the Zeno phenomenon. The type of Zeno execution observed in the first example cannot occur in hybrid systems with transverse invariants. However, in many cases the invariants cannot be described as $I(q) = \{x \in \mathbf{X} : \sigma(q, x) \geq 0\}$, for some σ analytic in x . This is

for example the case with polygonal invariant sets. Zeno executions related to these types of systems are the topic of current research, but will not be discussed further here.

The water tank automaton and the bouncing ball automaton both have transverse invariants. In these examples, the Zeno phenomenon is due to modeling simplifications. In the water tank example the dynamics of switching the input flow from one tank to the other are abstracted away, while in the bouncing ball example the possibly complicated bounce dynamics are replaced by a simple reset map. A way of resolving the Zeno phenomenon by reintroducing some of these physical considerations through the process of regularization is discussed in the next section. The bouncing ball example is the only one with a non-trivial reset map, which leads to discontinuities in the evolution of the continuous state. The water tank automaton, on the other hand, demonstrates a situation where analysis and controller synthesis techniques may fail in the presence of Zeno executions. It is easy to show (for example, by induction) that along all executions of the water tank automaton the water in both tanks remains above the desired levels. Clearly this is not the case for the physical system the automaton is supposed to model.

In all of the above examples, an infinite number of transitions takes place in the time interval $(\tau_\infty - \epsilon, \tau_\infty)$ for any $\epsilon > 0$. There are, however, also Zeno hybrid automata for which there exists an interval $(\tau_\infty - \epsilon, \tau_\infty)$ in which no transitions take place, while an infinite number of transitions takes place at τ_∞ . One such example is the obvious hybrid automaton that describes the evolution of the discontinuous differential equation $dx/dt = -\text{sgn } x$. More generally, differential equations of the form $dx/dt = F(x)$, where F is piecewise continuous, tend to exhibit this kind of Zeno behavior. The classical way of analyzing such systems is by introducing the notion of sliding modes [9,18].

4 Regularization

Regularization is a standard technique for dealing with differential equations whose solutions are not well defined. We propose a similar approach to extend Zeno executions beyond the Zeno time, primarily for the purpose of simulation. The formal treatment of how to regularize general Zeno hybrid automata is the topic of current research. Here we limit ourselves to specific regularizations of the water tank and bouncing ball automata introduced above. All regularizations are motivated by physical considerations of the underlying systems. For the water tank automaton, it is interesting to notice that different regularizations suggest different extensions of the executions. For the bouncing ball automaton, all extensions considered here are consistent with one another and with physical intuition.

Consider a non-blocking and deterministic hybrid automaton H and assume that for every $(q_0, x_0) \in \text{Init}$ the execution $\chi \in \mathcal{H}_{(q_0, x_0)}^\infty$ is Zeno. Regularization of H involves constructing a family of deterministic, non-blocking, and non-Zeno automata H_ϵ , parameterized by a real valued parameter $\epsilon > 0$. The continuous map

$$\phi : \mathbf{Q}_\epsilon \times \mathbf{X}_\epsilon \rightarrow \mathbf{Q} \times \mathbf{X}$$

relates the state of each H_ϵ to the state of H . Given an execution $\chi_\epsilon = (\tau_\epsilon, q_\epsilon, x_\epsilon)$, we use

$\phi(\chi_\epsilon)$ as a shorthand notation for the collection (τ, q, x) with $\tau = \tau_\epsilon$, and $(q(t), x(t)) = \phi(q_\epsilon(t), x_\epsilon(t))$ for all $t \in \tau$. Note that in general $\phi(\chi_\epsilon)$ will not be an execution of H . However, the construction of the family H_ϵ should be such that H_ϵ tends to H as ϵ tends to 0, in the sense that if $(q_{\epsilon_0}, x_{\epsilon_0}) \in \text{Init}_\epsilon$, then $\phi(q_{\epsilon_0}, x_{\epsilon_0}) \in \text{Init}$, and if χ_ϵ is an execution of H_ϵ with initial condition $(q_{\epsilon_0}, x_{\epsilon_0})$, then $\phi(\chi_\epsilon)$ converges to $\chi \in \mathcal{H}_{\phi(q_{\epsilon_0}, x_{\epsilon_0})}^\infty$ over all compact subintervals of $[\tau_0, \tau_\infty)$, where the convergence is taken in the Skorohod metric [4,6].³

Water Tank Automaton

We first study temporal and spatial regularizations of the water tank automaton. Throughout, we assume that $\max\{v_1, v_2\} < w < v_1 + v_2$, so that the automaton is Zeno.

Physically, temporal regularization represents a situation where there is a delay, $\epsilon > 0$, between the time the inflow is commanded to switch from one tank to the other and the time the switch actually takes place. The temporal regularization of the water tank automaton, WT_ϵ^Γ , (shown in Figure 3) is given by

- $\mathbf{Q} = \{q_1, q'_1, q_2, q'_2\}$ and $\mathbf{X} = \mathbb{R}^3$;
- $\text{Init} = \{q_1, q_2\} \times \{x \in \mathbf{X} : (x_1 \geq r_1) \wedge (x_2 \geq r_2)\}$;
- $f(q_1, x) = (w - v_1, -v_2, 0)$, $f(q'_1, x) = (w - v_1, -v_2, 1)$, $f(q_2, x) = (-v_1, w - v_2, 0)$, and $f(q'_2, x) = (-v_1, w - v_2, 1)$;
- $I(q_i) = \{x \in \mathbf{X} : x_i \geq r_i\}$ and $I(q'_i) = \{x \in \mathbf{X} : x_3 \leq \epsilon\}$ for $i = 1, 2$;
- $E = \{(q_1, q'_1), (q'_1, q_2), (q_2, q'_2), (q'_2, q_1)\}$;
- $G(q_1, q'_1) = \{x \in \mathbf{X} : x_2 \leq r_2\}$, $G(q'_1, q_2) = \{x \in \mathbf{X} : x_3 \geq \epsilon\}$, $G(q_2, q'_2) = \{x \in \mathbf{X} : x_1 \leq r_1\}$, and $G(q'_2, q_1) = \{x \in \mathbf{X} : x_3 \geq \epsilon\}$; and
- $R((q_i, q'_i), x) = \{(x_1, x_2, 0)\}$ for $i = 1, 2$ and $R((q'_i, q_2), x) = R((q'_2, q_1), x) = \{x\}$.

It is easy to show that WT_ϵ^Γ accepts a unique non-Zeno execution for each initial state. Overloading the notation somewhat, we can express the relation between the states of WT_ϵ^Γ and the states of WT through the map

$$\phi(q_i, (x_1, x_2, x_3)) = \phi(q'_i, (x_1, x_2, x_3)) = (q_i, (x_1, x_2)), \quad \text{for } i = 1, 2.$$

If we set $r_1 = r_2 = 1$, $v_1 = 2$, $v_2 = 3$, and $w = 4$, and assume that initially $x_1(0) = x_2(0) = 2$ and $q(0) = q_1$, then $\tau_\infty = 2$. Figure 4 shows simulation results for WT_ϵ^Γ ; x_1 and x_2 are plotted as functions of time for two values of ϵ , 0.1 and 0.01. Notice that as ϵ decreases, the execution of WT_ϵ^Γ converges over the interval $(\tau_0, \tau_\infty) = (0, 2)$ to the execution of WT , in the sense discussed above. For $t > \tau_\infty$, the continuous part of the execution of WT_ϵ^Γ tends to $(x_1(t), x_2(t)) = (1, 1 - (t - \tau_\infty))$.

The spatial regularization of the water tank automaton corresponds to a situation where the measurement of x_1 and x_2 is based on floats, which have to move a certain distance ϵ to register a change. It can be implemented by introducing a minimum deviation in the

³ Formally, we need to eliminate all “inert” transitions from τ , that is, replace all $[\tau_i, \tau'_i][\tau_{i+1}, \tau'_{i+1}]$ for which $\phi(q_\epsilon(\tau'_i), x_\epsilon(\tau'_i)) = \phi(q_\epsilon(\tau_{i+1}), x_\epsilon(\tau_{i+1}))$ by a single interval $[\tau_i, \tau'_{i+1}]$.

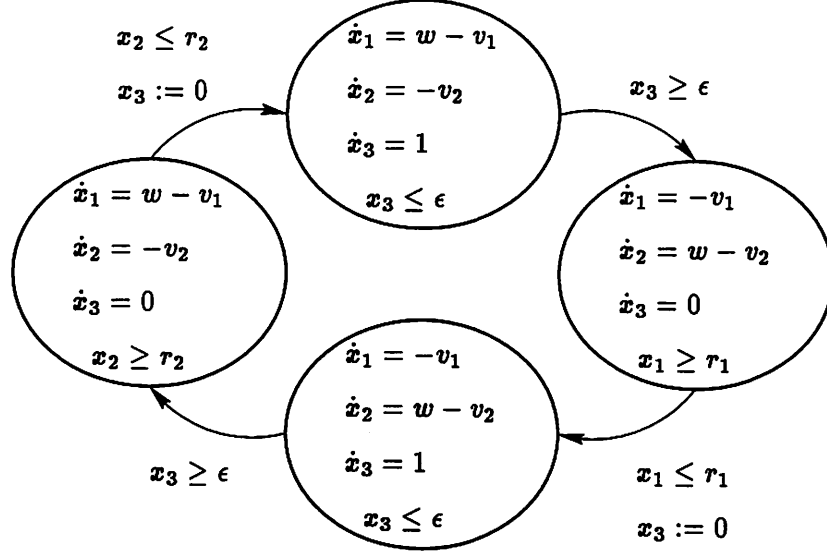


Fig. 3. Temporal regularization of the water tank automaton.

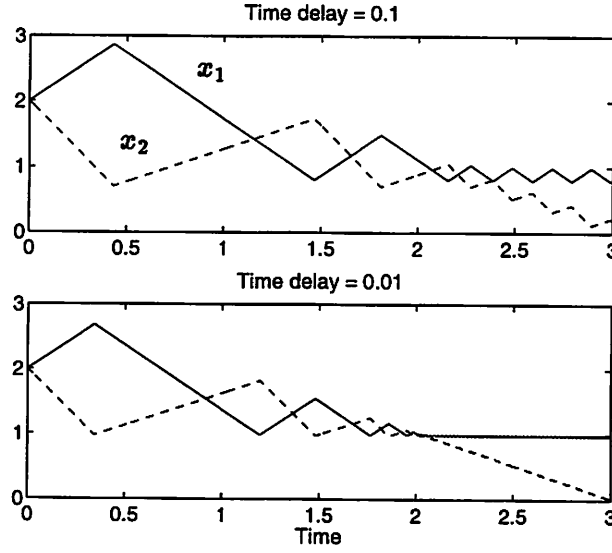


Fig. 4. Simulation of temporally regularized water tank automaton WT_ϵ^T .

continuous state variables between the discrete transitions. The regularized automaton, WT_ϵ^S , (shown in Figure 5) is given by

- $\mathbf{Q} = \{q_1, q_2\}$ and $\mathbf{X} = \mathbb{R}^4$;
- $\text{Init} = \{q_1, q_2\} \times \{x \in \mathbf{X} : (x_1 \geq r_1) \wedge (x_2 \geq r_2) \wedge (x_3 = r_1) \wedge (x_4 = r_2)\}$;
- $f(q_1, x) = (w - v_1, -v_2, 0, 0)$ and $f(q_2, x) = (-v_1, w - v_2, 0, 0)$;
- $I(q_1) = \{x \in \mathbf{X} : x_2 \geq x_4 - \epsilon\}$ and $I(q_2) = \{x \in \mathbf{X} : x_1 \geq x_3 - \epsilon\}$;
- $E = \{(q_1, q_2), (q_2, q_1)\}$;
- $G(q_1, q_2) = \{x \in \mathbf{X} : x_2 \leq x_4 - \epsilon\}$ and $G(q_2, q_1) = \{x \in \mathbf{X} : x_1 \leq x_3 - \epsilon\}$; and
- $R((q_1, q_2), x) = \{(x_1, x_2, x_3, x_2)\}$ and $R((q_2, q_1), x) = \{(x_1, x_2, x_1, x_4)\}$.

Again one can show that WT_ϵ^S accepts a unique non-Zeno execution for each initial state. We

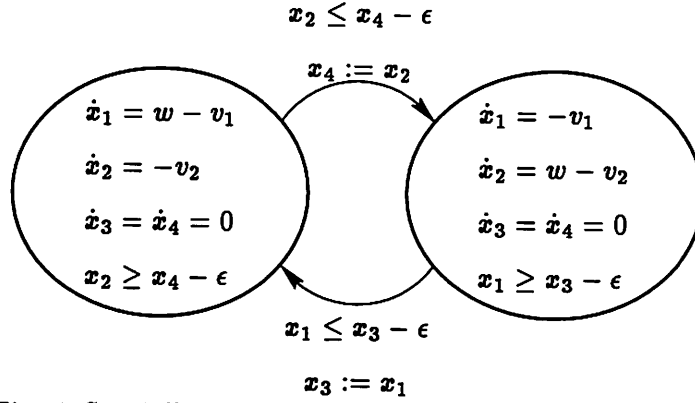


Fig. 5. Spatially regularized water tank automaton WT_ϵ^S .

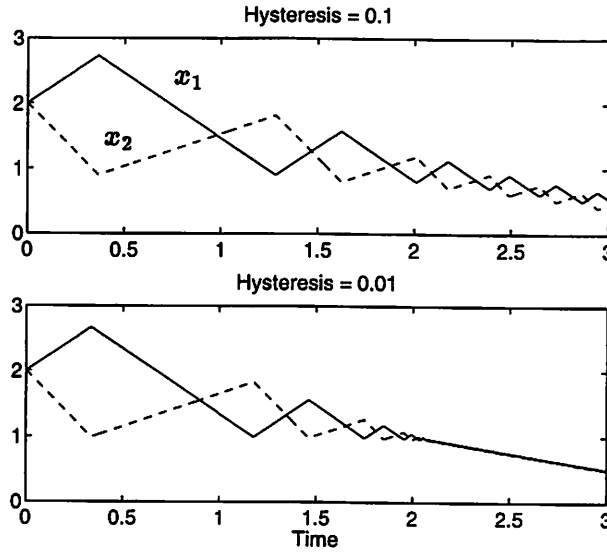


Fig. 6. Simulation of the spatially regularized water tank automaton WT_ϵ^S .

can relate the state of WT_ϵ^S to the state of WT through

$$\phi(q_i, (x_1, x_2, x_3, x_4)) = (q_i, (x_1, x_2)), \quad \text{for } i = 1, 2.$$

Figure 6 shows simulation results for WT_ϵ^S with $\epsilon = 0.1$ and 0.01 and the parameters given above. As for the temporal regularization, the execution of WT_ϵ^S converges to the execution of WT over the interval (τ_0, τ_∞) . For $t > \tau_\infty$, however, the execution converges to $x_1(t) = x_2(t) = -(t - \tau_\infty)/2 + 1$, which is different from the limit in the case of temporal regularization.

Bouncing Ball Automaton

Next, we consider temporal and dynamic regularizations of the bouncing ball automaton. Throughout we assume $c > 1$ so that BB is Zeno.

Temporal regularization corresponds to a situation where each bounce of the ball takes time $\epsilon > 0$. The temporally regularized automaton, BB_ϵ^T , (shown in Figure 7) is given by

- $\mathbf{Q} = \{q, q'\}$ and $\mathbf{X} = \mathbb{R}^3$;

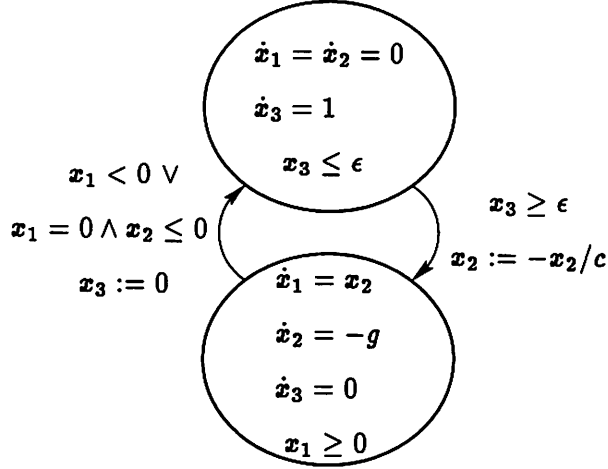


Fig. 7. Temporal regularization of the bouncing ball automaton.

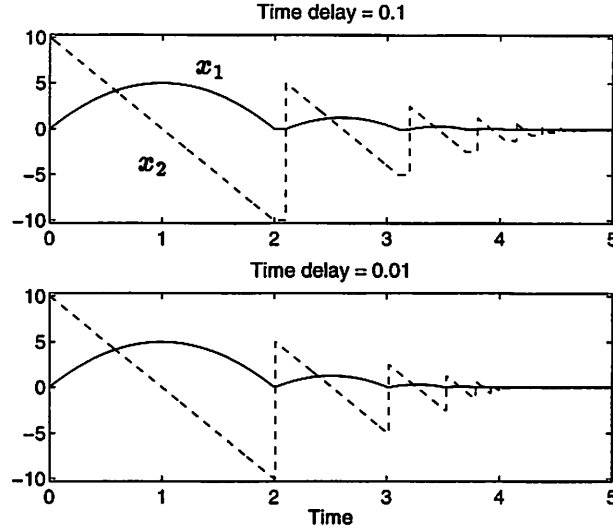


Fig. 8. Simulation of the temporal regularization of the bouncing ball.

- $\text{Init} = \{q\} \times \{x \in \mathbf{X} : x_1 \geq 0\}$;
- $f(q, x) = (x_2, -g, 0)$ and $f(q', x) = (0, 0, 1)$;
- $I(q) = \{x \in \mathbf{X} : x_1 \geq 0\}$ and $I(q') = \{x \in \mathbf{X} : x_3 \leq \epsilon\}$;
- $E = \{(q, q'), (q', q)\}$;
- $G(q, q') = \{x \in \mathbf{X} : [x_1 < 0] \vee [(x_1 = 0) \wedge (x_2 \leq 0)]\}$ and $G(q', q) = \{x \in \mathbf{X} : x_3 \geq \epsilon\}$; and
- $R((q, q'), x) = \{(x_1, x_2, 0)\}$ and $R((q', q), x) = \{(x_1, -x_2/c, x_3)\}$.

One can show that BB_ϵ^T accepts a unique non-Zeno execution for each initial state. The state of BB_ϵ^T is related to the state of BB by

$$\phi(q, (x_1, x_2, x_3)) = \phi(q', (x_1, x_2, x_3)) = (q, (x_1, x_2)).$$

If we set $g = 10$ and $c = 2$ and assume that initially $x_1(0) = 0$ and $x_2(0) = 10$, then $\tau_\infty = 4$. Figure 8 shows simulation results for BB_ϵ^T ; x_1 and x_2 are plotted as a function of time for $\epsilon = 0.1$ and 0.01 . As ϵ decreases, the execution of BB_ϵ^T converges to the execution of BB for $t \in (0, \tau_\infty)$. For $t > \tau_\infty$ the execution of BB_ϵ^T converges to the constant $x_1(t) = x_2(t) = 0$, which is physically intuitive.

Finally, consider a dynamic regularization of the bouncing ball automaton, where the ground is modeled as a stiff spring with spring constant $1/\epsilon$ and no damping. The dynamic regularization of the bouncing ball automaton, BB_ϵ^D , is shown in Figure 9 and is given by

- $Q = \{q, q'\}$ and $X = \mathbb{R}^2$;
- $\text{Init} = \{q\} \times \{x \in X : x_1 \geq 0\}$;
- $f(q, x) = (x_2, -g)$ and $f(q', x) = (x_2, -x_1/\epsilon)$;
- $I(q) = \{x \in X : x_1 \geq 0\}$ and $I(q') = \{x \in X : x_1 \leq 0\}$;
- $E = \{(q, q'), (q', q)\}$;
- $G(q, q') = \{x \in X : [x_1 < 0] \vee [(x_1 = 0) \wedge (x_2 \leq 0)]\}$ and $G(q', q) = \{x \in X : x_1 \geq 0\}$; and
- $R((q, q'), x) = \{x\}$ and $R((q', q), x) = \{(x_1, -x_2/c)\}$.

One can show that BB_ϵ^D is deterministic, non-blocking, and non-Zeno. The state of BB_ϵ^D is related to the state of BB by

$$\phi(q, (x_1, x_2)) = \phi(q', (x_1, x_2)) = (q, (x_1, x_2)).$$

Figure 10 shows simulation results for BB_ϵ^D with $\epsilon = 0.01$ and 0.0001 , and the parameter values given above. As ϵ decreases, the execution of BB_ϵ^D converges to the execution of BB before τ_∞ , and to $(x_1(t), x_2(t)) = (0, 0)$ after τ_∞ . Notice that the limiting behaviors of the temporal and dynamic regularizations for the bouncing ball are consistent with one another and with physical intuition.

5 Conclusions

We gave an introductory discussion on Zeno hybrid automata. We showed how Zeno executions can arise as a result of modeling over-abstraction, and discussed their importance for the analysis, controller synthesis, and simulation of hybrid automata.

In some cases (for example, in simulation) it may be desirable to extend a Zeno execution beyond the Zeno time. If the Zeno execution is a result of modeling over-abstraction, the extension should be motivated by intuition about what a more detailed model of the underlying physical process may involve. We proposed a method for performing such extensions using regularization techniques. Unfortunately, our examples indicated that in some cases the extension obtained through regularization may be non-unique, and may depend on the specific assumptions made about the detailed model.

The work presented here is just a first step towards a more complete understanding of the Zeno phenomenon. Current research focuses on deriving conditions to determine when an automaton is Zeno, and classifying different types of Zeno executions. In parallel we are working towards formalizing the notion of an extension and investigating different approaches to perform extensions for simulations, including regularization, averaging and Filippov solutions.

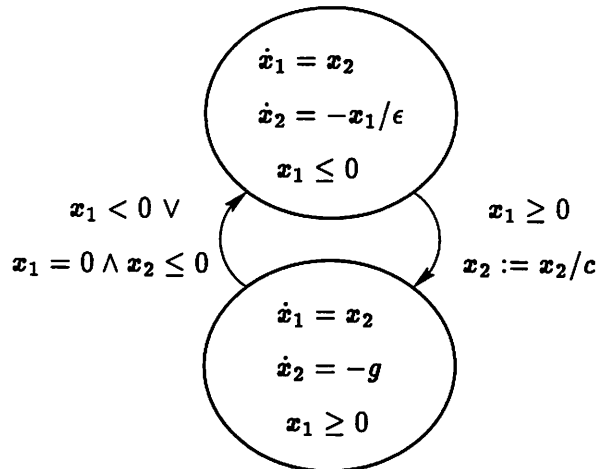


Fig. 9. Dynamic regularization of the bouncing ball.

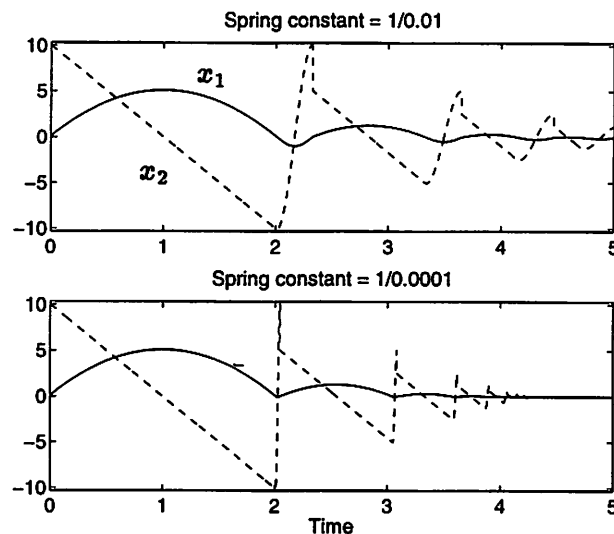


Fig. 10. Simulation of the dynamic regularization of the bouncing ball.

Acknowledgment

The authors would like to thank Gautam Biswas, Pieter Mosterman, and George Pappas for helpful discussions. The work by Karl Henrik Johansson was supported by the Swedish Foundation for International Cooperation in Research and Higher Education and Telefonaktiebolaget LM Ericsson's Foundation. The work by Magnus Egerstedt was sponsored by the Swedish Foundation for Strategic Research through its Centre for Autonomous Systems at KTH. The work by John Lygeros and Shankar Sastry was supported by ONR under grant N00014-97-1-0946 and by DARPA under contract F33615-98-C-3614.

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] 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.
- [4] P. Billingsley. *Convergence of probability measures*. Wiley & Sons, New York, 1968.
- [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] M. Broucke. Regularity of solutions and homotopy equivalence for hybrid systems. In *IEEE Conference on Decision and Control*, Tampa, FL, 1998.
- [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. Rantzer, and K. J. Åström. Fast switches in relay feedback systems. *Automatica*, 35(4), April 1999.
- [12] J. Lygeros, C. Tomlin, and S. Sastry. Controllers for reachability specifications for hybrid systems. *Automatica*, pages 349–370, March 1999.
- [13] J. Malmberg. *Analysis and Design of Hybrid Control Systems*. PhD thesis, Department of Automatic Control, Lund Institute of Technology, Lund, Sweden, May 1998.
- [14] The MathWorks, Inc., Natick, MA. *Simulink: dynamic system simulation for Matlab*, 1997. Version 2.
- [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] C. Tomlin, G. J. Pappas, and S. Sastry. Conflict resolution for air traffic management: a case study in multi-agent hybrid systems. Technical Report UCB/ERL M96/38, Department of Electrical Engineering and Computer Science, UC Berkeley, CA, July 1996.
- [18] V. I. Utkin. *Sliding Modes in Control Optimization*. Springer-Verlag, Berlin, 1992.
- [19] 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.
- [20] L. C. Young. *Optimal Control Theory*. Chelsea, second edition, 1980.