ELSEVIER

Systems & Control Letters 38 (1999) 141–150

SYSTEMS & CONTROL LETTERS

www.elsevier.com/locate/sysconle

On the regularization of Zeno hybrid automata

Karl Henrik Johansson[a,*], Magnus Egerstedt[b], John Lygeros[a], Shankar Sastry[a]

[a] *Department of Electrical Engineering and Computer Sciences, University of California, Berkeley, CA 94720-1770, USA*
[b] *Division of Optimization and Systems Theory, Royal Institute of Technology, S-10044 Stockholm, Sweden*

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.

Keywords: 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 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]

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

* Corresponding author.
E-mail addresses: johans@eecs.berkeley.edu (K.H. Johansson), magnuse@math.kth.se (M. Egerstedt), lygeros@eecs.berkeley.edu (J. Lygeros), sastry@eecs.berkeley.edu (S. Sastry)

[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.

0167-6911/99/$ - see front matter
PII: S0167-6911(99)00059-6

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 guarantee 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 simulations. Several packages have recently been developed for simulating hybrid dynamical systems, for example, Dymola [7], OmSim [16], SHIFT [6], 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], where an efficient and accurate simulation method was proposed that made use of the concept of Filippov solutions. Here we propose to extend the 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 [1–3,9]. 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

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 $n \geqslant 0$, are given the Euclidean metric topology, whereas countable and finite sets are given the discrete topology (all subsets are open). 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".

2.1. 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 $0 \leqslant i < N$, $I_i = [\tau_i, \tau_i']$ with $\tau_i \leqslant \tau_i' = \tau_{i+1}$;
- if $N < \infty$, either $I_N = [\tau_N, \tau_N']$ with $\tau_N \leqslant \tau_N' < \infty$, or $I_N = [\tau_N, \tau_N')$ with $\tau_N < \tau_N' \leqslant \infty$.

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 $\mathscr{T}$ the set of all hybrid time trajectories. Each $\tau \in \mathscr{T}$ 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 $\tau \in \mathscr{T}$ we use $t \in \tau$ as a shorthand notation for "there exists a j such that $t \in [\tau_i, \tau_j'] \in \tau$". For a topological space K and a $\tau \in \mathscr{T}$, we use $k : \tau \to K$ as a shorthand notation for a map assigning values from K to all $t \in \tau$. We say $\tau = \{I_i\}_{i=0}^{N} \in \mathscr{T}$ is a *prefix* of $\hat{\tau} = \{J_i\}_{i=0}^{M} \in \mathscr{T}$ and write $\tau \leqslant \hat{\tau}$ if either they are identical or τ is finite, $M \geqslant N, I_i = J_i$ for all $i = 0, \ldots, N-1$, and $I_N \subseteq J_N$. The prefix relation is a partial order on $\mathscr{T}$. Let $2^{\mathbf{X}}$ denote the set of all subsets of $\mathbf{X}$.

Definition 2 (*Hybrid automaton*). A hybrid automaton H is a collection $H = (Q, X, \mathrm{Init}, f, I, E, G, R)$, where

- Q is a finite collection of discrete variables;
- X is a finite collection of continuous variables with $\mathbf{X} = \mathbb{R}^n$;
- $\mathrm{Init} \subseteq \mathbf{Q} \times \mathbf{X}$ is a set of initial states;
- $f : \mathbf{Q} \times \mathbf{X} \to T\mathbf{X}$ is a vector field, assumed to be Lipchitz 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 $\boldsymbol{x} \in \mathbf{X}$ a reset relation.

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

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

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

For an execution $\chi = (\tau, q, \boldsymbol{x})$ we use $(q_0, \boldsymbol{x}_0) = (q(\tau_0), \boldsymbol{x}(\tau_0))$ to denote the initial state of χ. We say $\chi = (\tau, q, \boldsymbol{x})$ is a prefix of $\chi' = (r', q', \boldsymbol{x}')$ (write $\chi \leqslant \chi'$) if $\tau \leqslant \tau'$ and $(q(t), \boldsymbol{x}(t)) = (q'(t), \boldsymbol{x}'(t))$ for all $t \in \tau$. We say χ is a strict prefix of χ' (write $\chi < \chi'$) if $\chi \leqslant \chi'$ 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 $\chi = (\tau, q, \boldsymbol{x})$ (as opposed to generates). This conceptual difference allows one to consider hybrid automata that accept no executions for some initial states, accept multiple executions for the same initial states, or do not accept executions over arbitrarily long time horizons. An execution $\chi = (\tau, q, \boldsymbol{x})$ 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 $\mathscr{H}_{(q_0, \boldsymbol{x}_0)}$ to denote the set of all executions of H with initial condition $(q_0, \boldsymbol{x}_0) \in \mathrm{Init}$, $\mathscr{H}^{M}_{(q_0, \boldsymbol{x}_0)}$ to denote the corresponding maximal executions, and $\mathscr{H}^{\infty}_{(q_0, \boldsymbol{x}_0)}$ to denote the infinite executions. For all $(q_0, \boldsymbol{x}_0) \in \mathrm{Init}$, we have $\mathscr{H}^{\infty}_{(q_0, \boldsymbol{x}_0)} \subseteq \mathscr{H}^{M}_{(q_0, \boldsymbol{x}_0)} \subseteq \mathscr{H}_{(q_0, \boldsymbol{x}_0)}$, 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 $\mathscr{H}^{\infty}_{(q_0, \boldsymbol{x}_0)}$ is non-empty for all $(q_0, \boldsymbol{x}_0) \in \mathrm{Init}$. It is called deterministic if $\mathscr{H}^{M}_{(q_0, \boldsymbol{x}_0)}$ contains at most one element for all $(q_0, \boldsymbol{x}_0) \in \mathrm{Init}$.

2.2. Existence and uniqueness of executions

Next, we derive some simple conditions to characterize deterministic and non-blocking 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 $\sigma: \mathbf{Q} \times \mathbf{X} \to \mathbb{R}$, also analytic in its second argument, recursively define the Lie derivative of σ along f, $L_f^m\sigma: \mathbf{Q} \times \mathbf{X} \to \mathbb{R}$, by

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

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 $\sigma : \mathbf{Q} \times \mathbf{X} \to \mathbb{R}$, also analytic in its second argument, such that

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

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

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

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

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

For each discrete state $q \in \mathbf{Q}$, the set $\mathrm{Out}(q)$ 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 in 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 $\boldsymbol{x} \in \mathrm{Out}(q)$ there exists $(q,q') \in E$ such that $\boldsymbol{x} \in G(q,q')$ and $R((q,q'),\boldsymbol{x}) \neq \emptyset$.*

The following lemma states 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) $\boldsymbol{x} \in \bigcup_{(q,q') \in E} G(q,q')$ *implies* $\boldsymbol{x} \in \mathrm{Out}(q)$;
(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* $\boldsymbol{x} \in G(q,q')$ *imply* $|R(q,q',\boldsymbol{x})| \leqslant 1$.

The proofs of Lemmas 1 and 2 are given in [10]. Summarizing, we have that if a hybrid automaton with transverse invariants satisfies the conditions of Lemmas 1 and 2, then it accepts a unique infinite execution for all $(q_0,\boldsymbol{x}_0) \in \mathrm{Init}$.

3. Zeno hybrid automata

Definition 6 (*Zeno hybrid automaton*)**.** A hybrid automaton H is called Zeno if there exists $(q_0,\boldsymbol{x}_0) \in \mathrm{Init}$ such that all executions in $\mathscr{H}^{\infty}_{(q_0,\boldsymbol{x}_0)}$ 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.

3.1. Non-analytic automaton

Consider the smooth, but non-analytic, scalar function, s, given by $s(\boldsymbol{x}) = \exp(-1/|\boldsymbol{x}|)\sin(1/|\boldsymbol{x}|)$ if $\boldsymbol{x} \neq 0$ and $s(0) = 0$. We define a hybrid automaton by

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

It is easy to see that the infinite execution of this automaton with initial state $(q_1,-1)$ exhibits an infi-

[2] An alternative definition is that a Zeno automaton requires *at least one* execution in $\mathscr{H}^{\infty}_{(q_0,\boldsymbol{x}_0)}$ to be Zeno. For deterministic hybrid automata, such as the ones discussed in the subsequent examples, the two definitions coincide.

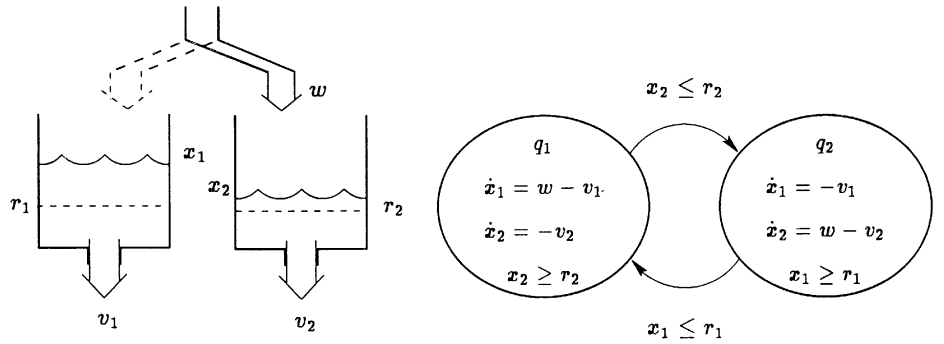

Fig. 1. Water tank system and corresponding hybrid automaton.

nite 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 $(-1, 0)$.

3.2. Water tank automaton

Consider the water tank system of Alur and Henzinger [2] shown in Fig. 1. For $i = 1, 2$, let $\boldsymbol{x}_i$ 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 $\boldsymbol{x}_1(0) > r_1$ and $\boldsymbol{x}_2(0) > r_2$). This is to be achieved by a switched control strategy that switches the inflow to Tank 1 whenever $\boldsymbol{x}_1 \leqslant r_1$ and to Tank 2 whenever $\boldsymbol{x}_2 \leqslant r_2$. More formally, the water tank automaton is a hybrid automaton, denoted by WT, with

- $\mathbf{Q} = \{q_1, q_2\}$ and $\mathbf{X} = \mathbb{R}^2$;
- $\text{Init} = \mathbf{Q} \times \{\boldsymbol{x} \in \mathbf{X}\colon (\boldsymbol{x}_1 \geqslant r_1) \wedge (\boldsymbol{x}_2 \geqslant r_2)\}, r_1, r_2 > 0$;
- $f(q_1, \boldsymbol{x}) = (w - v_1, -v_2)$ and $f(q_2, \boldsymbol{x}) = (-v_1, w - v_2)$, $v_1, v_2, w > 0$;
- $I(q_1) = \{\boldsymbol{x} \in \mathbf{X}\colon \boldsymbol{x}_2 \geqslant r_2\}$ and $I(q_2) = \{\boldsymbol{x} \in \mathbf{X}\colon \boldsymbol{x}_1 \geqslant r_1\}$;
- $E = \{(q_1, q_2), (q_2, q_1)\}$;
- $G(q_1, q_2) = \{\boldsymbol{x} \in \mathbf{X}\colon \boldsymbol{x}_2 \leqslant r_2\}$ and $G(q_2, q_1) = \{\boldsymbol{x} \in \mathbf{X}\colon \boldsymbol{x}_1 \leqslant r_1\}$; and
- $R((q_1, q_2), \boldsymbol{x}) = R((q_2, q_1), \boldsymbol{x}) = \{\boldsymbol{x}\}$.

It is straightforward to show that WT accepts a unique infinite execution for each initial state [10]. Moreover, if $\max\{v_1, v_2\} < w < v_1 + v_2$, then WT is Zeno with Zeno time $\tau_\infty = (\boldsymbol{x}_1(0) + \boldsymbol{x}_2(0) - r_1 - r_2)/(v_1 + v_2 - w)$, where $(\boldsymbol{x}_1(0), \boldsymbol{x}_2(0))$ is the continuous part of the initial state.

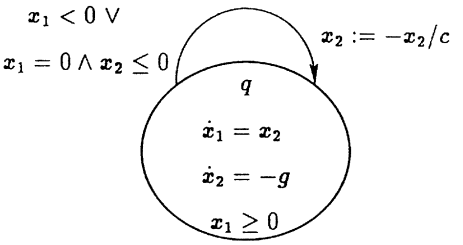

Fig. 2. Hybrid automaton for a simple model of a bouncing ball.

3.3. 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 $\boldsymbol{x}_1$ denote the altitude of the ball and $\boldsymbol{x}_2$ its vertical speed. A hybrid automaton, BB, describing this system is shown in Fig. 2 and is defined by

- $\mathbf{Q} = \{q\}$ and $\mathbf{X} = \mathbb{R}^2$;
- $\text{Init} = \{q\} \times \{\boldsymbol{x} \in \mathbf{X}\colon \boldsymbol{x}_1 \geqslant 0\}$;
- $f(q, \boldsymbol{x}) = (\boldsymbol{x}_2, -g)$ with $g > 0$;
- $I(q) = \{\boldsymbol{x} \in \mathbf{X}\colon \boldsymbol{x}_1 \geqslant 0\}$;
- $E = \{(q, q)\}$;
- $G(q, q) = \{\boldsymbol{x} \in \mathbf{X}\colon [\boldsymbol{x}_1 < 0] \vee [(\boldsymbol{x}_1 = 0) \wedge (\boldsymbol{x}_2 \leqslant 0)]\}$; and
- $R((q, q), \boldsymbol{x}) = \{(\boldsymbol{x}_1, -\boldsymbol{x}_2/c)\}$ with $c \geqslant 1$.

We can again show that BB accepts a unique execution for each initial state [10]. Moreover, if $c > 1$, then BB is Zeno with Zeno time

$$\tau_\infty = \frac{\boldsymbol{x}_2(0)}{g} + \frac{(c+1)\sqrt{\boldsymbol{x}_2(0)^2 + 2g\boldsymbol{x}_1(0)}}{g(c-1)},$$

where $(\boldsymbol{x}_1(0), \boldsymbol{x}_2(0))$ is the continuous part of the initial state.

3.4. 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 $I(q) = \{\boldsymbol{x} \in \mathbf{X}\colon \sigma(q,\boldsymbol{x}) \geqslant 0\}$, for some σ analytic in $\boldsymbol{x}$. 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 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 $(\tau_\infty - \varepsilon, \tau_\infty)$ for any $\varepsilon > 0$. There are, however, also Zeno hybrid automata for which there exists an interval $(\tau_\infty - \varepsilon, \tau_\infty)$ 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 $\mathrm{d}\boldsymbol{x}/\mathrm{d}t = -\operatorname{sgn}\boldsymbol{x}$. More generally, differential equations of the form $\mathrm{d}\boldsymbol{x}/\mathrm{d}t = F(\boldsymbol{x})$, 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 [8,18].

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 physical intuition. The regularizations are only presented graphically in this section; see [10] for formal definitions.

Consider a non-blocking and deterministic hybrid automaton H and assume that for every $(q_0, \boldsymbol{x}_0) \in \mathrm{Init}$ the execution $\chi \in \mathscr{H}^\infty_{(q_0,\boldsymbol{x}_0)}$ 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, $\varepsilon > 0$, and a continuous map, $\phi : \boldsymbol{Q}_\varepsilon \times \mathbf{X}_\varepsilon \to \mathbf{Q} \times \mathbf{X}$, relating the state of each H_ε to the state of H. Given an execution $\chi_\varepsilon = (\tau_\varepsilon, q_\varepsilon, \boldsymbol{x}_\varepsilon)$, we use $\phi(\chi_\varepsilon)$ as a shorthand notation for the collection $(\tau, q, \boldsymbol{x})$ with $\tau = \tau_\varepsilon$, and $(q(t), \boldsymbol{x}(t)) = \phi(q_\varepsilon(t), \boldsymbol{x}_\varepsilon(t))$ for all $t \in \tau$. Note that in general $\phi(\chi_\varepsilon)$ 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 zero, in the sense that if $(q_{\varepsilon_0}, \boldsymbol{x}_{\varepsilon_0}) \in \mathrm{Init}_\varepsilon$, then $\phi(q_{\varepsilon_0}, \boldsymbol{x}_{\varepsilon_0}) \in \mathrm{Init}$, and if χ_ε is the execution of H_ε with initial condition $(q_{\varepsilon_0}, \boldsymbol{x}_{\varepsilon_0})$, then $\phi(\chi_\varepsilon)$ converges to $\chi \in \mathscr{H}^\infty_{\phi(q_{\varepsilon_0},\boldsymbol{x}_{\varepsilon_0})}$ over all compact subintervals of $[\tau_0, \tau_\infty)$, where the convergence is taken in the Skorohod metric [4]. [3]

4.1. 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 WT is Zeno.

Physically, temporal regularization represents a situation where there is a delay, $\varepsilon > 0$, 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, $\mathrm{WT}^{\mathrm{T}}_\varepsilon$, is shown in Fig. 3. It is easy to show that $\mathrm{WT}^{\mathrm{T}}_\varepsilon$ accepts a unique non-Zeno execution for each initial state. Overloading the notation somewhat, we

[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 $\phi(q_\varepsilon(\tau_i'), \boldsymbol{x}_\varepsilon(\tau_i')) = \phi(q_\varepsilon(\tau_{i+1}), \boldsymbol{x}_\varepsilon(\tau_{i+1}))$ by a single interval $[\tau_i, \tau_{i+1}']$.

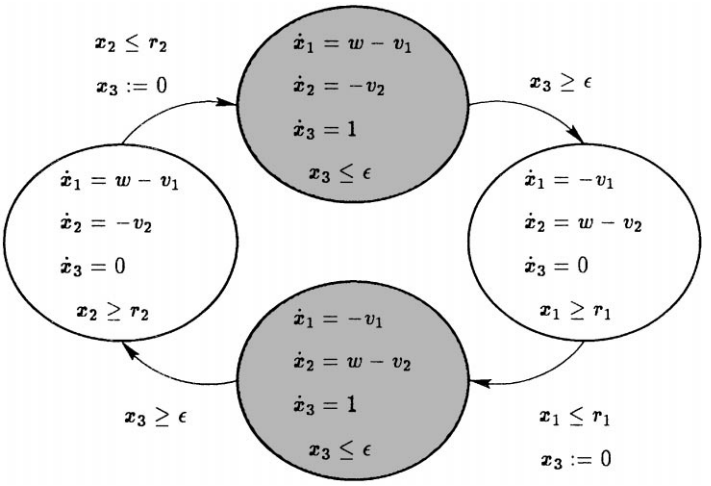

Fig. 3. Temporal regularization of the water tank automaton.

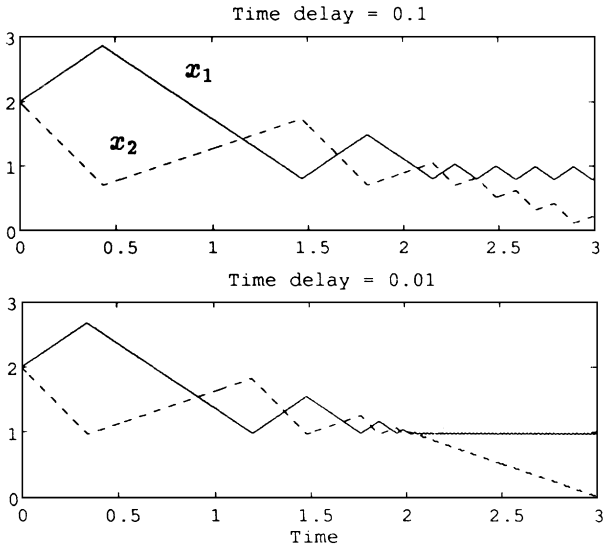

Fig. 4. Simulation of temporally regularized water tank automaton $\mathrm{WT}_{\varepsilon}^{\mathrm{T}}$.

can express the relation between the states of $\mathrm{WT}_{\varepsilon}^{\mathrm{T}}$ and the states of WT through the map $\phi(q_i,(\boldsymbol{x}_1,\boldsymbol{x}_2,\boldsymbol{x}_3))=\phi(q_i',(\boldsymbol{x}_1,\boldsymbol{x}_2,\boldsymbol{x}_3))=(q_i,(\boldsymbol{x}_1,\boldsymbol{x}_2))$, 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 $\boldsymbol{x}_1(0)=\boldsymbol{x}_2(0)=2$ and $q(0)=q_1$, then $\tau_\infty=2$. Fig. 4 shows simulation results for $\mathrm{WT}_{\varepsilon}^{\mathrm{T}}$; $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$ are plotted as functions of time for two values of ε, 0.1 and 0.01. Notice that as ε decreases, the execution of $\mathrm{WT}_{\varepsilon}^{\mathrm{T}}$ 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 $\mathrm{WT}_{\varepsilon}^{\mathrm{T}}$ tends to $(\boldsymbol{x}_1(t),\boldsymbol{x}_2(t))=(1,1-(t-\tau_\infty))$.

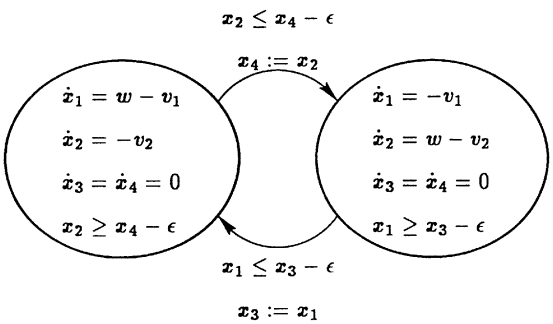

Fig. 5. Spatially regularized water tank automaton $\mathrm{WT}_{\varepsilon}^{\mathrm{S}}$.

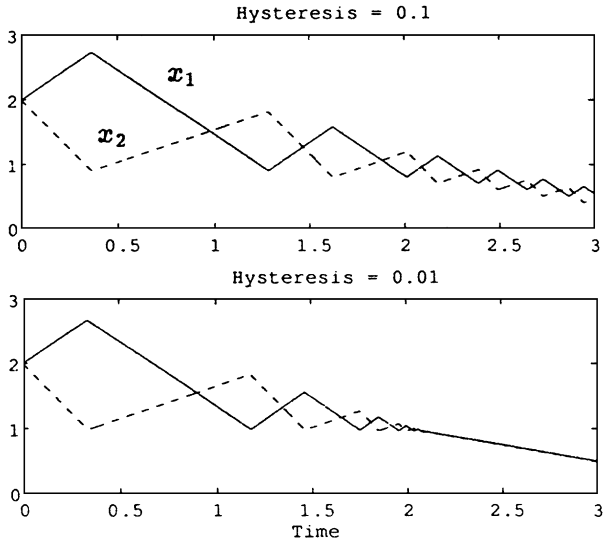

Fig. 6. Simulation of the spatially regularized water tank automaton $\mathrm{WT}_\varepsilon^{\mathrm{S}}$.

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 continuous state variables between the discrete transitions. The regularized automaton, $\mathrm{WT}_\varepsilon^{\mathrm{S}}$, is presented in Fig. 5. Again one can show that $\mathrm{WT}_\varepsilon^{\mathrm{S}}$ accepts a unique non-Zeno execution for each initial state. We can relate the state of $\mathrm{WT}_\varepsilon^{\mathrm{S}}$ to the state of WT through $\phi(q_i,(x_1,x_2,x_3,x_4))=(q_i,(x_1,x_2))$, for $i=1,2$. Fig. 6 shows simulation results for $\mathrm{WT}_\varepsilon^{\mathrm{S}}$ with $\varepsilon = 0.1$ and 0.01 and the parameters given above. As for the temporal regularization, the execution of $\mathrm{WT}_\varepsilon^{\mathrm{S}}$ converges to the execution of WT over the interval $[\tau_0,\tau_\infty)$. 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.

4.2. 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 $\varepsilon>0$. The temporally regularized automaton, $\mathrm{BB}_\varepsilon^{\mathrm{T}}$, is given in Fig. 7. One can show that $\mathrm{BB}_\varepsilon^{\mathrm{T}}$ accepts a unique non-Zeno execution for each initial state. The state of $\mathrm{BB}_\varepsilon^{\mathrm{T}}$ is related to the state of BB by $\phi(q,(x_1,x_2,x_3))=\phi(q',(x_1,x_2,x_3))=(q,(x_1,x_2))$.

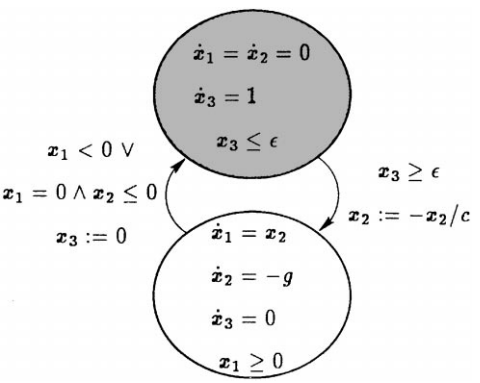

Fig. 7. Temporal regularization of the bouncing ball automaton.

If we set $g=10$ and $c=2$ and assume that initially $x_1(0)=0$ and $x_2(0)=10$, then $\tau_\infty=4$. Fig. 8 shows simulation results for $\mathrm{BB}_\varepsilon^{\mathrm{T}}$; x_1 and x_2 are plotted as a function of a time for $\varepsilon=0.1$ and 0.01. As ε decreases, the execution of $\mathrm{BB}_\varepsilon^{\mathrm{T}}$ converges to the execution of BB for $t\in[0,\tau_\infty)$. For $t>\tau_\infty$ the execution of $\mathrm{BB}_\varepsilon^{\mathrm{T}}$ 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 $1/\varepsilon$ and no damping. The dynamic regularization of the bouncing ball automaton, $\mathrm{BB}_\varepsilon^{\mathrm{D}}$, is shown in Fig. 9. One can show that $\mathrm{BB}_\varepsilon^{\mathrm{D}}$ is deterministic, non-blocking, and non-Zeno. The state of $\mathrm{BB}_\varepsilon^{\mathrm{D}}$ is related to the state of BB by $\phi(q,(x_1,x_2))=\phi(q',(x_1,x_2))=(q,(x_1,x_2))$.

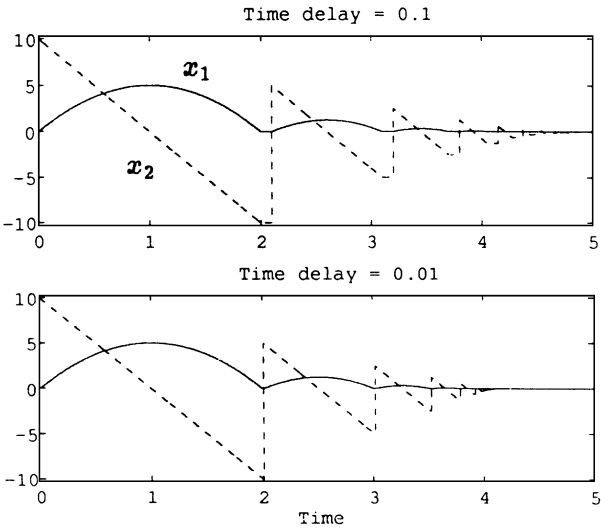

Fig. 8. Simulation of the temporal regularization of the bouncing ball.

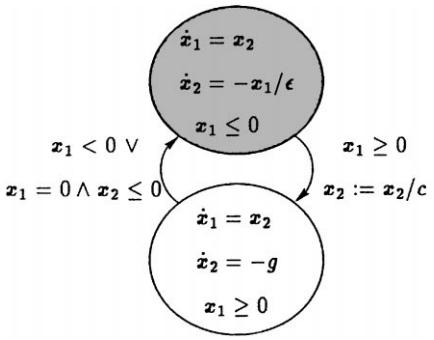

Fig. 9. Dynamic regularization of the bouncing ball.

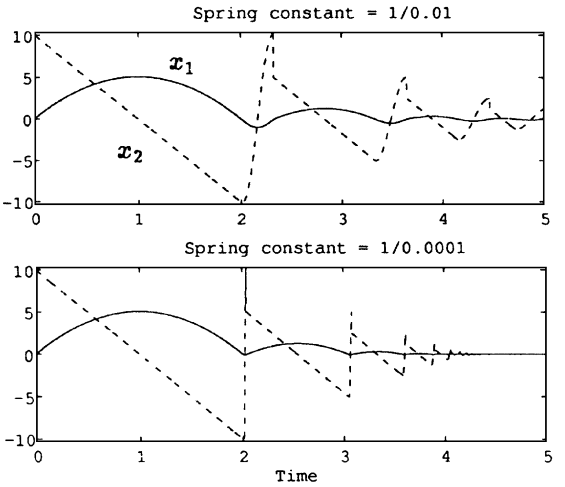

Fig. 10. Simulation of the dynamic regularization of the bouncing ball.

Fig. 10 shows simulation results for BB_ε^D with $\varepsilon = 0.01$ 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. This is, however, not surprising, since it is well-known that variable structure

systems need not have a unique (Filippov) solution [18].

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 simulation, including regularization, averaging, and Filippov solutions.

Acknowledgements

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, D.L. Dill, Automata for modeling real-time systems, in: Proceedings of ICALP'90, Lecture Notes in Computer Science, vol. 443, Springer, Berlin, 1990, pp. 322–335.

[2] R. Alur, T.A. Henzinger, Modularity for timed and hybrid systems, in: A. Mazurkiewicz, J. Winkowski (Eds.), CONCUR 97: Concurrency Theory, Lecture Notes in Computer Science, vol. 1243, Springer, Berlin, 1997, pp. 74–88.

[3] B. Bérard, P. Gastin, A. Petit, On the power of non observable actions in timed automata, in: Actes du STACS'96, Lecture Notes in Computer Science, vol. 1046, Springer, Berlin, 1996, pp. 257–268.

[4] P. Billingsley, Convergence of Probability Measures, Wiley, New York, 1968.

[5] M. Branicky, V. Borkar, S. Mitter, A unified framework for hybrid control: model and optimal control theory, IEEE Trans. Automat. Control 43 (1) (1998) 31–45.

[6] A. Deshpande, A. Göllü, L. Semenzato, The SHIFT programming language for dynamic networks of hybrid automata, IEEE Trans. Automat. Control 43 (4) (1998) 4283–4288.

[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, Dordrecht, 1988.

[9] T.A. Henzinger, The theory of hybrid automata, in: Proceedings of the 11th Annual Symposium on Logic in Computer Science, IEEE Computer Society Press, Silverspring, MD, 1996, pp. 278–292.

[10] K.H. Johansson, M. Egerstedt, J. Lygeros, S. Sastry, On the regularization of Zeno hybrid automata, Technical Report UCB/ERL M99/23, Department of Electrical Engineering and Computer Sciences, University of California, Berkeley, 1999.

[11] K.H. Johansson, A. Rantzer, K.J. Åström, Fast switches in relay feedback systems, Automatica 35 (4) (1999) 539–552.

[12] J. Lygeros, C. Tomlin, S. Sastry, Controllers for reachability specifications for hybrid systems, Automatica 35 (3) (1999) 349–370.

[13] J. Malmborg, Analysis and design of hybrid control systems, Ph.D. 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, vol. F, San Francisco, CA, July 1996, pp. 259–264.

[16] S.E. Mattsson, M. Andersson, K.J. Åström, Object-oriented modelling and simulation, in: D.A. Linkens (Ed.), CAD for Control Systems, Marcel Dekker, New York, 1993, pp. 31–69 (Chapter 2).

[17] C. Tomlin, G.J. Pappas, 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 Sciences, University of California, Berkeley, July 1996.

[18] V.I. Utkin, Sliding Modes in Control Optimization, Springer, Berlin, 1992.

[19] A.J. van der Schaft, J.M. Schumacher, Complementarity modeling of hybrid systems, IEEE Trans. Automat. Control 43 (4) (1998) 483–490.

[20] L.C. Young, Optimal Control Theory, 2nd ed., Chelsea, New York, 1980.