Regularization of Zeno Hybrid Automata

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

Abstract. Hybrid automata that can exhibit infinitely many discrete transitions in finite time are studied. These are called Zeno hybrid automata and it is shown that they can be regularized, so that the executions of the automata are defined also for times beyond the Zeno time. The theory is illustrated on hybrid models for a bouncing ball and a water tank control problem.

1. Introduction

There has been a great deal of recent activity in the area of hybrid systems. Despite considerable advances, however, even fundamental issues, such as conditions for existence and uniqueness of executions for classes of hybrid systems, are still the topic of intense research activity [18]. In addition to the usual technical conditions associated with existence results for conventional continuous dynamical systems, new complications due to blocking and non-determinism need to be considered in the analysis of hybrid systems. Yet another issue is the Zeno phenomenon, where the executions of the hybrid system exhibit an infinite number of discrete transitions in finite time. We denote these automata Zeno hybrid automata.

Even though the Zeno phenomenon seems like a mathematical curiosity, it turns out to be important when modeling, controlling and simulating hybrid systems. For example, Zeno executions may arise in the optimal control of even continuous systems. Relaxed controls solutions to optimal control problems involve chattering [5]. They also appear in variable control structure systems [17] and in simple relay control systems [10]. Zeno executions may also arise in controllers designed to satisfy reachability specifications, where the controller occasionally tries to prevent the system from reaching an undesirable state by forcing it to take an infinite number of transitions [16]. Timed automata with Zeno properties have been analyzed to some extent in [1, 9, 3, 2]. In more general hybrid automata, however, subtleties in the continuous dynamics make the analysis much more involved.

Fast mode switchings are important in hybrid systems, because they may deteriorate the efficiency and accuracy of simulations considerably. There exist several simulation packages for simulation of hybrid dynamical systems, e.g., SHIFT [7], OmSim [15], and Simulink [13].
The authors are not, however, aware of any packages today that takes special care of fast switching in evolutions of hybrid systems. Instead, either the simulation slows down or the accuracy decreases as the time intervals between discrete transitions get smaller. In some cases, the simulation may even give erroneous results or error messages. There are few suggestions in the literature on how to simulate hybrid systems. It was proposed in [14, 12] that for hybrid models that may exhibit sliding in the sense of Filippov, the fast switchings can be detected and the automaton can be extended with a new discrete state. The continuous flow in the new state is given by the Filippov solution. As we will see, these systems form a particular class of Zeno hybrid automata.

The main contribution of this paper is to show how Zeno executions can be analyzed using “regularization”. By the act of a regularization and taking its limit as some parameter tends to zero, executions of the automaton can be extended beyond the point of convergence of the discrete switching times. Our approach is illustrated on two particular examples: a bouncing ball automaton and a water tank automaton. Several regularization techniques are analyzed for these systems and interesting connections to averaging is pointed out. Ongoing work includes extending some of the results to a more general class of hybrid systems. The outline of the paper is as follows. Formal definitions of hybrid automata and executions of hybrid automata are given in Section 2. Results on existence and uniqueness of executions are derived and a particular property of some hybrid automata denoted transverse invariants is introduced. Some examples of Zeno hybrid systems are analyzed in Section 3 in order to highlight the various aspects of the Zeno phenomenon. Section 4 discusses regularization for Zeno hybrid automata. In particular, various properties of regularization of the examples introduced in Section 3 are derived. The paper ends with conclusions and a discussion on simulation of hybrid systems in Section 5.

2. Hybrid Automata

2.1. Notation. Consider a finite collection $V$ of variables. Let $V$ denote the set of valuations of these variables. We use lower case letters to denote both a variable and its valuation. We refer to variables whose set of valuations is finite as discrete and to variables whose set of valuations is a subset of a Euclidean space as continuous. We assume that the Euclidean space, $\mathbb{R}^n$ for $n \geq 0$, is 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 $\overline{U}$ to denote its closure, $U^o$ its interior, $\partial U$ its boundary, $U^c$ its complement, $|U|$ its cardinality, and $2^U$ the set of all subsets of $U$. We use $\land$ to denote the logical “and” and $\lor$ to denote the logical “or”.

2.2. Hybrid Automata and Executions. The following definitions are based on [11].

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 set of discrete variables;
- $X$ is a set of continuous variables;
De/#0Cnition /28Hybrid Time Tra jectory/#29 se quenc e of intervals of the r eal line/, to denote
f
H
relation de/#0Cnes a partial order on the set of executions/, and that the set of executions is pre/#0Cx
execution is maximal if it is not a strict pre/#0Cx of an y other execution/. Note that the pre/#0Cx
is a /#0Cnite sequence ending with an in terv al of the form /#5B
arbitrarily long time horizons /#28Zeno/#29/. An execution
there exists a
j
all
t
/= every
subject of
a guar d/; and

Let I_i = [\tau_i, \tau_i^1]. Then for all i, \tau_i \leq \tau_i^1 and for i > 0, \tau_i = \tau_i^1-1.

Note that hybrid time trajectories can extend to infinity if \tau 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. 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 \to K as a shorthand notation for a map assigning a value from K to each t \in \tau. We say
\tau = \{I_i\}_{i=0}^N \in \mathcal{T} is a prefix of \tau' = \{J_i\}_{i=0}^M \in \mathcal{T} and write \tau \leq \tau' if either they are identical or
\tau is finite, M \geq N, I_i = J_i for all i = 0, \ldots, N - 1 and I_N \subseteq J_N.

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

• (q(\tau_0), x(\tau_0)) \in \text{Init} (initial condition);
• for all i with \tau_i < \tau_i^1, x and q are continuous over [\tau_i, \tau_i^1] and for all t \in [\tau_i, \tau_i^1), x(t) \in
I(q(t)) and \frac{dx(t)}{dt} = f(q(t), x(t)) (continuous evolution); and
• for all i, e \in (q(\tau_i^1), q(\tau_{i+1}^1)) \in E, x(\tau_i^1) \in G(e), and x(\tau_{i+1}) \in R(e, x(\tau_i^1)) (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
\chi. We say \chi is a prefix of \chi' (write \chi \leq \chi') if \tau \leq \tau' and (q(t), x(t))(t) = (q(t), x'(t)) for
all t \in \tau. We say \chi is a strict prefix of \chi' (write \chi < \chi') if \chi \leq \chi' and \chi \neq \chi'. We say an
execution is maximal if it is not a strict prefix of any other execution. Note that the prefix
relation defines a partial order on the set of executions, and that the set of executions is prefix
closed.

Unlike conventional continuous dynamical systems, the interpretation here is that an automa-
ton \mathcal{H} accepts (as opposed to generates) an execution \chi = (\tau, q, x). This allows one to consider
hybrid automata that accepts no executions for some initial states (blocking), accept multiple
executions for the same initial state (non-deterministic), or do not accept executions over
arbitrarily long time horizons (Zeno). An execution \chi = (\tau, q, x) is called infinite, if \tau is an
infinite sequence, or \( \sum_i (\tau'_i - \tau_i) = \infty \). We use \( \mathcal{H}_{(q_0, x_0)} \) to denote the set of all infinite executions of \( H \) with initial condition \((q_0, x_0) \in \text{Init}\). An execution is \textit{admissible} if \( \sum_i (\tau'_i - \tau_i) = \infty \), and it is \textit{Zeno} if it is infinite but not admissible. For a Zeno execution \( \chi = (\tau, q, x) \) we define the Zeno time as \( \tau_\infty = \sum_i (\tau'_i - \tau_i) < \infty \).

Note that, in general, the set \( \mathcal{H}_{(q_0, x_0)} \) may be empty, or may contain multiple executions, some of which are admissible while others are Zeno. In subsequent sections we establish conditions under which infinite executions exists and are unique. We then give examples of hybrid automata that accept Zeno executions, and provide a relaxation methodology for extending Zeno executions to admissible ones.

2.3. Existence and Uniqueness of Executions.

**Definition 4** (Non-Blocking and Deterministic Automaton). A hybrid automaton is called 
\textit{non-blocking} if \( \mathcal{H}_{(q_0, x_0)} \) is non-empty for all \((q_0, x_0) \in \text{Init}\). A hybrid automaton is called 
\textit{deterministic} if \( \mathcal{H}_{(q_0, x_0)} \) contains at most one element for all \((q_0, x_0) \in \text{Init}\).

We restrict our attention to a special class of hybrid automata, where the vector field is transverse to the boundary of the invariant set. Assume \( f \) is analytic in its second argument. For a function \( \sigma : Q \times X \to \mathbb{R} \), also analytic in its second argument, inductively define the Lie derivative of \( \sigma \) along \( f \), \( L_f^m \sigma : Q \times X \to \mathbb{R} \), by

\[
L_f^0 \sigma(q, x) = \sigma(q, x) \quad \text{and} \quad L_f^m \sigma(q, x) = \left( \frac{\partial}{\partial x} L_f^{m-1} \sigma(q, x) \right) f(q, x), \quad \text{for} \quad m > 0.
\]

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

- \( I(q) = \{ x \in X : \sigma(q, x) \geq 0 \} \) for all \( q \in Q \); and
- for all \((q, x) \in Q \times X \) 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 relative degree as a function \( n : Q \times X \to \mathbb{N} \) by

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

For all \( q \in Q \) we also define the set

\[
\text{Out}(q) := \left\{ x \in X : L_f^{n(q, x)} \sigma(q, x) < 0 \right\}.
\]

**Lemma 1.** A hybrid automaton with transverse invariants is non-blocking if for all \( q \in Q \) 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 \( \tau \) is a finite sequence.
First consider the case \( \tau = \{ \tau_i \}_{i=0}^{N-1} \). Let \((q_N, x_N) = \lim_{t \to \tau} \langle q(t), x(t) \rangle\). Note that, by the definition of execution and a standard existence argument for continuous dynamical systems, \( \chi \) can be extended to \( \hat{\chi} = (\hat{\tau}, \hat{q}, \hat{x}) \) with \( \hat{\tau} = \{ [\tau_i] \}_{i=0}^{N+1} \) \( \hat{q}(\tau_N) = q_N \) and \( \hat{x}(\tau_N) = x_N \). This contradicts the maximality of \( \chi \).

Now consider the case \( \tau = \{ \tau_i \}_{i=0}^{N} \), and let \((q_N, x_N) = (q(\tau_N), x(\tau_N))\). If \( x_N \in \text{Out}(q_N)^c = \{ x \in X : L^N_{\tau}(q, x) > 0 \} \), then, by the assumption that \( f \) and \( \sigma \) are analytic in their second argument, there exists \( \epsilon > 0 \) such that \( \chi \) can be extended to \( \hat{\chi} = (\hat{\tau}, \hat{q}, \hat{x}) \) with \( \hat{\tau} = \tau[\tau_{N+1}, \tau_{N+1}^{-1}] \) with \( \tau_{N+1} = \tau_N \) and \( \tau_{N+1}^{-1} = \tau_{N+1} + \epsilon \) by continuous evolution. If, on the other hand \( x_N \in \text{Out}(q_N) \), then there exists \((q', x') \in Q \times X\) such that \((q_N, q') \in E\), \( x_N \in G(q_N, q') \) and \( x' \in R((q_N, q'), x_N) \). Therefore, \( \chi \) can be extended to \( \hat{\chi} = (\hat{\tau}, \hat{q}, \hat{x}) \) with \( \hat{\tau} = \{ \tau_i \}_{i=0}^{N+1} \), \( \tau_{N+1} = \tau_{N+1}^{-1} = \tau_N \), \( q(\tau_{N+1}) = q' \), \( x(\tau_{N+1}) = x' \) by a discrete transition. In both cases the maximality of \( \chi \) is contradicted. \( \square \)

The conditions of Lemma 1 indicate that a hybrid automaton is non-blocking if transitions with non-trivial reset relations are enabled along the boundary of the invariant sets.

**Lemma 2.** A hybrid automaton with transverse invariants is deterministic if for \( q, q', q'' \in Q \)

\[
\begin{align*}
\bullet \quad & \bigcup_{(q,q') \in E} G(q,q') \subseteq \text{Out}(q); \\
\bullet \quad & \text{if } (q, q') \in E \text{ and } (q, q'') \in E \text{ with } q' \neq q'' \text{ then } G(q, q') \cap G(q, q'') = \emptyset; \text{ and} \\
\bullet \quad & \text{if } (q, q') \in E \text{ then for all } x \in X, |R(q, q', x)| \leq 1.
\end{align*}
\]

**Proof.** Assume, for the sake of contradiction, that there exists an initial state \((q_0, x_0) \in \text{Init}\) and two infinite 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 \( \psi = (\rho, p, y) \) denote the maximal common prefix of \( \chi \) and \( \hat{\chi} \) (such a prefix exists as the executions start at the same initial state). Note that \( \psi \) is an execution, but not an infinite execution, as \( \chi \neq \hat{\chi} \). As in the proof of Lemma 1, \( \rho \) can be assumed to be of the form \( \rho = \{ [\rho_i, \rho'] \}_{i=0}^{N} \), otherwise the maximality of \( \psi \) 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. \( \rho'_N \) is not a time when a discrete transition takes place in either \( \chi \) 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 \( \chi \) 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 \( \psi \).

**Case 2:** \( \rho'_N \in \{ \tau_i \} \) and \( \rho'_N \notin \{ \hat{\tau}_i \} \), i.e. \( \rho'_N \) is a time when a discrete transition takes place in \( \chi \) but not in \( \hat{\chi} \). The fact that a discrete transition takes place from \((q_N, x_N)\) in \( \chi \) indicates that there exists \( q' \in Q \) such that \((q_N, q') \in E \) and \( x_N \in G(q_N, q') \). The fact that no discrete transition takes place from \((q_N, x_N)\) 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 2 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., $\rho'_N$ is a time when a discrete transition takes place in both $\chi$ and $\hat{\chi}$. The fact that a discrete transition takes place from $(q_N, x_N)$ in both $\chi$ 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 2, $x' = \hat{x}'$. Therefore, the prefixes of $\chi$ 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 $\psi$. \hfill \square

The conditions of Lemma 2 indicate that an automaton with open invariant sets is deterministic if discrete transitions have to be forced by the continuous flow exiting the invariant set, no two discrete transitions can be enabled simultaneously, and no point can be mapped onto two different points by the reset map. Summarizing Lemmas 1 and 2 give 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**

A hybrid system is Zeno if it has a solution with infinitely many discrete transitions in finite time.

**Definition 6 (Zeno Hybrid Automaton).** A hybrid automaton $H$ is called Zeno, if there exists $(q_0, x_0) \in \text{Init}$ such that $\mathcal{H}_{(q_0, x_0)}$ contains a Zeno execution.

We illustrate the Zeno property through a number of examples.

**Example 1 (Non-Analytic System).** Consider the hybrid automaton, $H$, defined by:

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

It is easy to see that the execution of $H$ with initial state $(q_1, -1)$ exhibits an infinite number of discrete transitions by $\tau_{\infty} = 1$. The reason for this is that the (non-analytic) function $x \sin(1/x)$ has an infinite number of zeros in the finite interval $(-1, 0)$.

**Example 2 (Water Tank System).** Next, 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 $r_i > 0$ denote the (constant) flow of water out of tank $i$. Let $w$ denote the constant flow of water into the system, dedicated exclusively to either Tank 1 or Tank 2 at each point in time. The control task is to keep the water volumes above $r_1$ and $r_2$, respectively (assuming that $x_1(0) > r_1$ and
$x_2(0) > r_2$. This is to be achieved by a switched control strategy that switches the inflow to Tank 1 whenever $x_1 \leq r_1$ and to Tank 2 whenever $x_2 \leq r_2$. More formally:

**Definition 7** (Water Tank Automaton). The water tank automaton is a hybrid automaton with:

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

**Proposition 1** (Existence and Uniqueness of Executions). The water tank automaton accepts a unique 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 $L_2^1 \sigma(q_1, x) = -v_2 < 0$ and $L_1^1 \sigma(q_2, x) = -v_1 < 0$. Since both $f$ and $\sigma$ are analytic functions of $x$ and $I(q_i) = \{x \in X : \sigma(q_i, x) \geq 0\}$, 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 $\text{Out}(q_1) = \{x \in X : x_2 \leq r_2\} = G(q_1, q_2)$ (and similarly for $q_2$). This implies that Condition 1 of Lemma 1 and Condition 2 of Lemma 2 are satisfied. Moreover, $|R(q_1, q_2, x)| = |R(q_2, q_1, x)| = 1$, therefore Condition 1 of Lemma 1 and Condition 2 of Lemma 2 are satisfied. Condition 2 of Lemma 2 is also trivially satisfied. The claim follows by Theorem 1. \qed

**Proposition 2** (Water Tank Automaton is Zeno). If $\max\{v_1, v_2\} < w < v_1 + v_2$ then the water tank automaton is Zeno. The Zeno time of the execution starting at $(q(0), x_1(0), x_2(0))$ is

$$r_\infty = \frac{x_1(0) + x_2(0) - r_1 - r_2}{v_1 + v_2 - w}.$$

*Proof.* By straightforward calculation. \qed
**Example 3** (Bouncing Ball System). The bouncing ball automaton models an elastic ball that subsequently is bouncing off 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.

**Definition 8** (Bouncing Ball Automaton). The bouncing ball automaton, shown in Figure 2, is a hybrid automaton with

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

**Proposition 3** (Existence and Uniqueness of Executions). The bouncing ball automaton accepts a unique execution for each initial state \( (q, x_0) \).

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

Note that \( n(q, x) = 0 \) if \( x_1 \neq 0 \), \( n(q, x) = 1 \) if \( (x_1 = 0) \land (x_2 \neq 0) \), and \( n(q, x) = 2 \) if \( (x_1 = 0) \land (x_2 = 0) \). Therefore, \( \text{Out}(q) = \{ x \in X : [x_1 < 0] \lor [(x_1 = 0) \land (x_2 < 0)] \lor [(x_1 = 0) \land (x_2 = 0)] \} = G(q, q) \). This implies that Condition 1 of Lemma 1 and Condition 2 of Lemma 2 are satisfied. Moreover, \( |R(q, q, x)| = 1 \), therefore Condition 1 of Lemma 1 and Condition 2 of Lemma 2 are satisfied. Condition 2 of Lemma 2 is also trivially satisfied. The proposition follows by Theorem 1.

**Proposition 4** (Bouncing Ball Automaton is Zeno). The bouncing ball automaton is Zeno. The Zeno time of the execution with initial state \( (q, x(0)) \) is

\[
\tau_\infty = -\frac{x_2(0) + \sqrt{x_2(0)^2 + 2gx_1(0)}}{g} + \frac{2cx_2(0)}{g(c-1)}
\]

*Proof.* Note that the first bounce occurs at time \( \tau_1 = \tau_1 = (-x_2(0) + \sqrt{x_2(0)^2 + 2gx_1(0)})/g \). The \( N^\text{th} \) bounce occurs at time \( \tau_N = \tau_1 + \sum_{k=1}^{N} 2x_2(0)/(ge^{k-1}) \). Since \( c > 1 \) the series on the right hand side converges.

**Example 4** (Modified Water Tank System). Consider the water tank automaton in Definition 7, but with the modified invariants

\[
I(q_1) = \{ x \in X : x_2 \geq x_1 \} \quad \text{and} \quad I(q_2) = \{ x \in X : x_1 \geq x_2 \}
\]

and the modified guards

\[
G(q_1, q_2) = \{ x \in X : x_2 \leq x_1 \} \quad \text{and} \quad G(q_2, q_1) = \{ x \in X : x_1 \leq x_2 \}.
\]
It is easy to show that all executions of this automaton are Zeno. In particular, an execution starting at \((x_1(0), x_2(0))\) with \(x_1(0) \geq x_2(0)\) reaches a point where \(x_1 = x_2\) at time \(\tau_\infty = (x_1(0) - x_2(0))/(w + v_1 - v_2)\) (a similar conclusion holds if \(x_1(0) \leq x_2(0)\)).

The examples introduced above have some similar properties but shed light on different aspects of the Zeno phenomenon. The first example is more of a mathematical curiosity; it can not be observed in non-blocking, deterministic automata with transverse invariants for example, since the vector fields and boundaries of the invariant sets are analytic. The second example is the only one with non-trivial reset relations, which result in discontinuities in the state execution. Finally, the last example is the only one for which there exists an interval \((\tau_\infty - \epsilon, \tau_\infty)\) with \(\epsilon > 0\) with no discrete transitions; an infinite number of transitions takes place at \(\tau_\infty\), whereas in the remaining examples there are infinitely many transitions on any interval \((\tau_\infty - \epsilon, \tau_\infty)\). More generally, differential equations of the form \(\dot{x} = F(x)\) where \(F\) is piecewise analytic (which can easily be modeled as hybrid automata) tend to exhibit sliding modes. This class of systems will not be studied further here, the reader is referred to [8] for a thorough treatment. We only notice that for system like the modified water tank system in Example 4, any imperfections in the switching mechanism will give a chattering execution. Analysis of this type of behavior in general hybrid automata seems to be possible using generalizations of Filippov’s definition of solution for differential equations with discontinuous right-hand side.

4. Regularization of Hybrid Automata

Regularization is a standard technique in the differential equations literature for dealing with systems that are not well defined. We suggest that a similar approach can be used to continue the execution for Zeno hybrid automata beyond the Zeno time. A hybrid automaton can be regularized in many different ways. In this section we analyze temporal, spatial, and dynamical regularizations based on physical considerations. By regularization of an automaton \(H\), we mean adding a small perturbation \(\epsilon > 0\) to \(H\) to make the system non-Zeno. We denote the regularized automaton \(H_\epsilon\) and assume that \(H_\epsilon \to H_0\) as \(\epsilon \downarrow 0\). The continuous state of the regularized system is assumed to converge to the continuous state of the Zeno system.

**Definition 9** (Regularization). A hybrid automaton \(H_\epsilon\) is a proper regularization of a non-blocking and deterministic Zeno hybrid automaton \(H\) if

\[ x_1 \leq 0 \]

\[ x_2 := -x_2/\epsilon \]

\[ x_1 = x_2 \]

\[ x_2 = -g \]

\[ x_1 \geq 0 \]
• $H_\epsilon$ accepts a unique and non-Zeno execution for all $\epsilon \in (0, \varepsilon)$;
• $x_\epsilon(t) \in I_\epsilon(q_\epsilon(t))$ for all executions $(\tau, q_\epsilon, x_\epsilon)$; and
• there exists a partition $x_\epsilon = (x_\epsilon^T, x_\epsilon^T)^T$, such that $\lim_{\epsilon \to 0} x_\epsilon(t) = x(t)$ for all $t$ in every closed subinterval of $(\tau, \infty)$. The convergence is in the sense of the Skorohod metric.2

4.1. Water Tank Automaton. Spatial regularization of the water tank automaton is defined by introducing a minimum deviation in the continuous state variables between the discrete transitions. A physical interpretation of the regularization is to assume that the measurement devices for $x_1$ and $x_2$ are based on floats, which have to move a certain distance corresponding to the volume $\epsilon$ to respond.

**Definition 10** (Spatial Regularization of Water Tank System). The spatially regularized water tank automaton $H_\epsilon$ is defined as an extension of the water tank automaton in Definition 7 through the following equations:

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

**Proposition 5.** Assume $\max\{v_1, v_2\} < w < v_1 + v_2$. The spatially regularized water tank automaton $H_\epsilon$ is a proper regularization of the water tank automaton $H$ for every $\epsilon > 0$. The first two entries of the continuous state $x_\epsilon = (x_\epsilon^T, x_\epsilon^T)^T$ of $H_\epsilon$ satisfies

$$\lim_{\epsilon \to 0} x_\epsilon(t) = \begin{bmatrix} r_1 + (w - v_1 - v_2)(t - \tau_\infty)/2 \\ r_2 + (w - v_1 - v_2)(t - \tau_\infty)/2 \end{bmatrix}$$

for all $t$ in every closed subinterval contained in $[\tau_\infty, \infty)$ and every Zeno execution of $H$.

**Proof.** The proof of that $H_\epsilon$ accept a unique execution for each initial state is similar to the proof of Proposition 1. It is easy to show that the minimal time between three consecutive discrete transitions is $2\epsilon/(v_1 + v_2 - w) > 0$. Hence, $H_\epsilon$ is non-Zeno.

To show that $x_\epsilon(t) \in I_\epsilon(q_\epsilon(t))$ for all executions $(\tau, q_\epsilon, x_\epsilon)$, first note that $x_\epsilon(\tau_0) \in I_\epsilon(q_\epsilon(\tau_0))$. The continuous transition gives $x_\epsilon(t) \in I_\epsilon(q_\epsilon(t))$ for all $t \in [\tau_0, \tau_0']$ and the discrete transition gives $x_\epsilon(\tau_i) \in I_\epsilon(q_\epsilon(\tau_i))$. Induction gives the result.

Both the limit $\lim_{\epsilon \to 0} x_\epsilon(t) = x(t)$ for all $t$ in every closed subinterval of $(\tau_\infty, \infty)$ and the limit in the proposition follows from straightforward integration of the vector field of the automaton. \hfill $\square$

---

2The Skorohod metric has the following interpretation for the continuous state of an execution [6]. The limit $\lim_{\epsilon \to 0} x_\epsilon(t) = x(t)$, for all $t \in [t_0, t_1]$, means that there exists a strictly increasing, continuous, surjective function $\kappa : [t_0, t_1] \to [t_0, t_1]$, such that $max_{t \in [t_0, t_1]} |s(t) - t| \to 0$ as $\epsilon \to 0$ and $max_{t \in [t_0, t_1]} \|x_\epsilon(s(t)) - x(t)\| \to 0$ as $\epsilon \to 0$. The function $\kappa$ can be interpreted as a deformation of the time axis. The Skorohod metric was originally used for stochastic point processes [4].
Consider the water tank system with parameters $r_1 = r_2 = 1$, $v_1 = 2$, $v_2 = 3$, and $w = 4$. Let the initial condition be $x_1(0) = x_2(0) = 2$ and $q(0) = q_1$. It follows then from Proposition 2 that the original water tank automaton in Definition 7 is Zeno with $\tau_\infty = 2$. Simulations of the water tank automaton with spatial regularization are shown in Figure 3 for $\epsilon = 0.1$ and 0.01. Note that the averaged vector field prior to the Zeno time for the non-regularized system is equal to the limiting vector field for the regularized system after the Zeno time. (Connect the crossings between $x_1$ and $x_2$ in Figure 3 with a straight line.)

The following temporal regularization of the water tank automaton corresponds to taking finite time $\epsilon > 0$ to change the inflow from Tank 1 to Tank 2 and vice versa.

**Definition 11 (Temporal Regularization of Water Tank System).** The **temporally regularized water tank automaton** $H_\epsilon$ is defined as an extension of the water tank automaton in Definition 7 through the following equations:

- $Q = \{q_1, q_1', q_2, q_2'\}$ and $X = \mathbb{R}^3$;
- $\text{Init} = \{q_1\} \times \{x \in X : x_i > r_i, i = 1, 2\}$, $r_1, r_2 > 0$;
- $f(q_1, x) = (w - v_1, -v_2, 0)^T$, $f(q_1', x) = (w - v_1, -v_2, 1)^T$, $f(q_2, x) = (-v_1, w - v_2, 0)^T$, and $f(q_2', x) = (-v_1, w - v_2, 1)^T$;
- $I_i(q_1) = \{x \in X : x_i \geq r_i\}$ and $I_i(q_1') = \{x \in X : x_3 \leq \epsilon\}$ for $i = 1, 2$;
- $E = \{(q_1, q_1'), (q_1', q_2), (q_2, q_2'), (q_2', q_1)\}$;
- $G(q_1, q_1') = \{x \in X : x_2 \leq r_2\}$, $G(q_1', q_2) = \{x \in X : x_3 \geq \epsilon\}$, $G(q_2, q_2') = \{x \in X : x_1 \leq r_1\}$, and $G(q_2', q_1) = \{x \in X : x_3 \geq \epsilon\}$; and
- $R(q_1, q_1', x) = R(q_2', q_2, x) = (x_1, x_2, 0)^T$ and $R(q_1, q_2, x) = R(q_2', q_1, x) = x$.

**Proposition 6.** Assume $\max\{v_1, v_2\} < w < v_1 + v_2$. The temporally regularized water tank automaton $H_\epsilon$ is not a proper regularization of the water tank automaton $H$.
Time delay = 0.1

Time delay = 0.01

Figure 4. 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 $x_2$.

Proof. We show that there is a time $t'$ such that $x_e(t') \notin I_\epsilon(q_e(t'))$. Consider the initial state $x_e(0) = (r_1 + \delta, r_2 + \delta, 0)^T$, $\delta > 0$, and $q_e(0) = q_1$. Then $x_{e2}(t) = r_2$ at $t = \delta/v_2$, so a discrete transition to $q_1'$ takes place. It is followed by a discrete transition to $q_2$ at $t = \delta/v_2 + \epsilon$. However, $x_{e2}(\delta/v_2 + \epsilon) < r_2$ so $x_e \notin I_\epsilon(q_2)$. Hence, $t' = \delta/v_2 + \epsilon$ proves the result.

The vector fields in the delay states $q_1'$ and $q_2'$ can be defined in other ways than in Definition 11. Two natural alternatives are $f(q_1', x) = (-v_1, -v_2, 1)^T$ and $f(q_2', x) = (0, 0, 1)^T$. Neither of them, however, define proper regularizations.

Figure 4 shows simulations of the water tank automaton with temporal regularization for $\epsilon = 0.1$ and 0.01. Note that Figures 3 and 4 illustrate that the regularized automata have different limiting solutions for $t > \tau_\infty$. The execution tends to $x_1(t) = -(t - \tau_\infty)/2 + 1$ and $x_2(t) = -(t - \tau_\infty)/2 + 1$, $t > \tau_\infty$, for the automaton with the spatial regularization, while the execution tends to $x_1(t) = 1$ and $x_2(t) = -(t - \tau_\infty) + 1$, $t > \tau_\infty$, for the temporal regularization. Only the spatial regularization is a proper regularization as introduced in Definition 9.

4.2. Bouncing Ball Automaton. Consider the bouncing ball, but assume that each bounce takes a certain amount of time $\epsilon > 0$. This corresponds to one form of temporal regularization.

Definition 12 (Temporal Regularization of Bouncing Ball). The temporal regularized bouncing ball automaton $H_e$ is defined as an extension of the bouncing ball automaton in Definition 8 through the following equations:

- $Q = \{q, q'\}$ and $X = \mathbb{R}^3$;
- $\text{Init} = \{q\} \times \{x \in X : x_1 = 0, x_2 \geq 0\}$;
\[ f(q, x) = (x_2, -g, 0)^T \text{ with } g > 0 \text{ and } f(q', x) = (0, 0, 1); \]
\[ I(q) = \{ x \in \mathbf{X} : x_1 \geq 0 \} \text{ and } I(q') = \{ x \in \mathbf{X} : x_3 \leq \epsilon \}; \]
\[ E = \{(q, q'), (q', q)\}; \]
\[ G(q, q') = \{ x \in \mathbf{X} : x_1 < 0 \} \text{ and } G(q', q) = \{ x \in \mathbf{X} : x_3 \geq \epsilon \}; \text{ and} \]
\[ R(q, q', x) = (x_1, x_2, 0)^T \text{ and } R(q', q, \tilde{x}) = (x_1, -x_2/c, 0)^T, \quad c > 1. \]

**Proposition 7.** The temporal regularized bouncing ball automaton \( H_c \) in Definition 12 is a proper regularization of the bouncing ball automaton \( H \) for every \( c > 0 \). The continuous state \( x_c \) of \( H \) satisfies \( \lim_{t \to 0} x_c(t) = 0 \) for all \( t \) in every closed subinterval contained in \([\tau_\infty, \infty)\) and every Zeno execution of \( H \).

**Proof.** The proof of that \( H_c \) accept a unique execution for each initial state is similar to the proof of Proposition 3. \( H_c \) is non-Zeno because every pair of consecutive discrete transitions takes a time greater than \( \epsilon \). The condition \( x_c(t) \in I_c(q_c(t)) \) for all executions \((\tau_\epsilon, q_\epsilon, x_c)\) can be shown similar to Proposition 5. Finally, the limits follow from straightforward integration of the vector field.

Consider the bouncing ball automaton with parameters \( g = 10 \) and \( c = 2 \). Initial conditions \( x_1(0) = 0 \) and \( x_2(0) = 10 \) give then, according to Proposition 4, that the automaton is Zeno with Zeno time \( \tau_\infty = 4 \). Figure 5 shows simulations of the temporal regularized automaton \( H_c \) with \( \delta = 0.1 \) and 0.01. Note that the execution of the regularized system converges to the execution for the original system for \( t \in (0, \tau_\infty) \). For \( t > \tau_\infty \) the execution of the regularized system converges zero. In particular, the averaged vector field for \( x_1 \) prior to the Zeno time for the non-regularized system is equal to the limiting vector field for the regularized system after the Zeno time.

Similar to the water tank automaton, there exist non-proper regularizations also for the bouncing ball automaton. Consider the following modified temporal regularization.

**Proposition 8** (Modified Temporal Regularization of Bouncing Ball). Define a modified temporal regularized bouncing ball automaton \( H_c \) as the regularized automaton in Definition 12 but with \( f(q', x) = (x_2, -g, 1) \). Then, \( H_c \) is not a proper regularization of the bouncing ball automaton \( H \).

**Proof.** We show that there is a time \( t' \), such that \( x_c(t') \notin I_c(q_c(t')) \). Consider the initial state \((q_0(0), x_c(0)) = (q_1, (0, 0, 0)^T) \in Q_0 \times X_0 \). Then \( x_c(\epsilon) = (-g\epsilon^2/2, -g\epsilon, \epsilon)^T \). A discrete transition \( q' \rightarrow q \) is then enforced, but \( x_c(\epsilon) \notin I(q) = I(q_c(\epsilon)) \). Hence, for \( t' = \epsilon \) we have that \( x_c(t') \notin I_c(q_c(t')) \), which proves the result.

The execution for \( t > \tau_\infty \) converges to the physical intuitive solution \( x_c(t) = 0 \) as \( \epsilon \downarrow 0 \) for the temporal regularized automaton, see Figure 5. The execution for the automaton with the modified regularization will not converge to zero for \( t > \tau_\infty \).

Next consider dynamical regularization of the bouncing ball automaton. Model the ground as a stiff spring with spring constant \( 1/\epsilon \) and no damping.
Figure 5. Simulation of bouncing ball with temporal regularization. The upper plot corresponds to $H_{0.1}$ and the lower to $H_{0.01}$. For the non-regularized system, we have the Zeno time at $\tau_\infty = 4$. The executions of the regularized system tend to the execution of the Zeno system as $\epsilon \downarrow 0$ for $t \in [0, \tau_\infty)$. The continuous state tends to zero for $t$ in every closed interval $[\tau_\infty, t_1]$.

**Definition 13** (Dynamical Regularization of Bouncing Ball). The dynamically regularized bouncing ball automaton $H_\epsilon$ is defined as an extension of the bouncing ball automaton in Definition 8 through the following equations:

- $Q = \{q, q'\}$ and $X = \mathbb{R}^2$;
- $\text{Init} = \{q\} \times \{x \in X : x_1 = 0, x_2 \geq 0\}$;
- $f(q, x) = (x_2, -g)^T$ with $g > 0$ and $f(q', x) = (x_2, -x_1/\epsilon)$;
- $I_r(q) = \{x \in X : x_1 \geq 0\}$ and $I_s(q) = \{x \in X : x_1 \leq 0\}$;
- $E = \{(q, q'), (q', q)\}$;
- $G(q, q') = \{x \in X : x_1 \leq 0\}$ and $G(q', q) = \{x \in X : x_1 \geq 0\}$; and
- $R(q, q', x) = x$ and $R(q', q, x) = (x_1, -x_2/\epsilon)^T$, $\epsilon > 1$.

**Proposition 9.** The dynamically regularized bouncing ball automaton $H_\epsilon$ is a proper regularization of the bouncing ball automaton $H$ for every $\epsilon > 0$. The continuous state $x_\epsilon$ of $H_\epsilon$ satisfies $\lim_{\epsilon \downarrow 0} x_\epsilon(t) = 0$ for all $t$ in every closed subinterval contained in $[\tau_\infty, \infty)$ and every Zeno execution of $H$.

*Proof.* The proof is similar to Proposition 7. Note that $H_\epsilon$ is non-Zeno because the time spent in $q'$ each period is $\pi \sqrt{\epsilon}$.

Figure 6 shows simulations of the dynamically regularized bouncing ball automaton with $\epsilon = 0.01$ and $0.0001$. The execution of the regularized system converges as $\epsilon$ tends to zero, compare Figure 5. Note that the bouncing ball automaton with proper regularization do not
only have the same limiting solution for $t < \tau_{\infty}$, but also for $t > \tau_{\infty}$. This agrees with the simulations in Figures 5 and 6.

It was previously pointed out for the spatially regularized water tank automaton and for the temporally and dynamically regularized bouncing ball automata that the averaged vector field prior to the Zeno time for the non-regularized systems were equal to the limiting vector fields for the regularized systems after the Zeno time. Note that this was the case only for automata with proper regularizations. It is believed that there is a close relation between the continuation of an execution derived from a proper regularization and from averaging. This topic will be further explored in future reports.

5. Conclusions

In this paper, we have shown how Zeno hybrid automata can be regularized to extend their execution beyond the Zeno time. The work presented is the first set of results on modeling and simulation of hybrid systems. Zeno phenomena appear to be convenient abstract models of physical systems. Optimal control, relay systems, and controller designs for hybrid systems are examples of how they arise in the control of systems. A major motivation for this 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.

The detection of Zeno automata can be done statically or dynamically. Static detection would be to investigate the hybrid automaton off-line to find Zeno loops. One way of doing this...
could be by using graph theoretical algorithms taking into account the guards and invariants associated with the discrete locations (see for example [12]). Dynamical detection would be based on the study of the numerical solution on-line, i.e., during the course of simulation. For example, if the number of discrete transitions during a time interval is greater than a certain bound, the system is defined to be Zeno. The details of these tests and their characterization is the subject of future work.

On the other hand, after the Zeno phenomenon has been detected, extension of the execution beyond the Zeno time is motivated by concerns of simulation efficiency, since when the number of discrete transitions per time unit becomes large, the simulation time increases. As was discussed in this paper, some hybrid automata are Zeno because the vector fields are pointing towards a common switching surface. One example is the modified water tank system in Example 4. In this case Filippov solutions can be used to extend the solution beyond the Zeno point. The hybrid automaton can be extended with an extra discrete state, which represents a sliding mode, see [14, 12]. In other situations, it may be possible to reduce the number of discrete states by replacing the fast switching between two states with a single state representing the average dynamics. A complete classification of the regularized Zeno automata is in progress.

Acknowledgment. The authors would like to thank 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. 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


Department of Electrical Engineering and Computer Science, University of California at Berkeley, 333 Cory Hall # 1770, Berkeley, CA 94720-1770, U.S.A., Tel. +1 510 642-1906, Fax +1 510 642-1341

E-mail address: johans@eecs.berkeley.edu

Division of Optimization and Systems Theory, Royal Institute of Technology, S-100 44 Stockholm, Sweden, Tel. +46 8 7907220, Fax +46 8 225320

E-mail address: magnuse@math.kth.se

Department of Electrical Engineering and Computer Science, University of California at Berkeley, 333 Cory Hall # 1770, Berkeley, CA 94720-1770, U.S.A., Tel. +1 510 643-5806, Fax +1 510 642-1341

E-mail address: lygeros@eecs.berkeley.edu

Department of Electrical Engineering and Computer Science, University of California at Berkeley, 333 Cory Hall # 1770, Berkeley, CA 94720-1770, U.S.A., Tel. +1 510 642-1857, Fax +1 510 642-1341

E-mail address: sastry@eecs.berkeley.edu