The Discrete Time Behavior of Lazy Linear Hybrid Automata

Manindra Agrawal and P. S. Thiagarajan

School of Computing,
National University of Singapore
agarwal,thiagu@comp.nus.edu.sg

Abstract. We study the class of *lazy linear hybrid automata with finite precision*. The key features of this class are:

- The observation of the continuous state and the rate changes associated with mode switchings take place with bounded delays.
- The values of the continuous variables can be observed with only finite precision.
- The guards controlling the transitions of the automaton are finite conjunctions of arbitrary linear constraints.

We show that the discrete time dynamics of this class of automata can be effectively analyzed without requiring resetting of the continuous variables during mode changes. In fact, our result holds for guard languages that go well beyond linear constraints.

1 Introduction

We present a class of linear hybrid automata and show that their discrete time behavior can be effectively computed and represented as finite state automata. A hybrid automaton of the kind we study is meant to be a model of a closed loop system consisting of a digital controller interacting with a plant whose state variables evolve in a continuous manner. The controller will sample the state of the plant at periodic *discrete time instances*. Typically, these time instances will be determined by the system clock of the processor implementing the controller. This state information will consist of the current values of the relevant plant variables as observed by the sensors. These values will be digitized with *finite precision* and reported to the controller. Using this information, the controller may decide to switch the mode of the plant by generating suitable output signals that will be transmitted to the actuators, which in turn will effect the desired mode change.

An important feature in this setting -and this will be reflected in our automata-is that the sensors will report the current values of the variables and the actuators will effect changes in the rates of evolution of the variables with *bounded delays*. More specifically, the state observed at the instant T_k is a state that held at some time in a bounded interval contained in (T_{k-1}, T_k). Further, if an instantaneous mode change has taken place at T_k from the standpoint of the digital controller, then any necessary change in the rate of a variable will not kick in immediately. Rather, it will do so at some time in a bounded interval contained in (T_k, T_{k+1}).

A restriction we impose is that each variable's allowed range of values is bounded. In addition, we focus, for simplicity, on the case where there is a single rate vector associated with each mode instead of a bounded (rectangular) region of flows as is often done [1]. Our automata are a variant of *linear* hybrid automata [2] in that the guards controlling the mode switches are assumed to be conjunctions of *linear inequalities.*

Our main result is that the discrete time behavior of such an automaton is regular and moreover, this behavior can be effectively computed and represented as a finite state automaton. Indeed, any reasonable language of constraints can be used to form the guards (for instance, conjunctions of polynomial inequalities) and the main result will continue to hold.

As is well known, hybrid automata have a great deal of expressive power. In a variety of settings, the control state reachability problem is undecidable, as reported for instance in [3, 4]. A number of undecidability results in a discrete time setting have also been reported in the literature but these results are mainly for piecewise-affine (and not, as considered here, piecewise constant) systems with infinite precision; see for instance [5, 6]. A sharp delineation of the boundary between decidable and undecidable features of hybrid automata is drawn in [7] as well as [1].

These results, as also the positive results reported for example in [8–11] suggest that except under very restrictive settings, one can not expect to get decidability if the continuous variables don't get reset during mode changes (in case their rates change as a result of the mode change). Viewed as a model of digital controllers that interact with plants through sensors and actuators, the resetting requirement severely restricts the modelling power of the automaton. Our results show that by focusing on the discrete time behavior *and* requiring finite precision, we can allow the continuous variables to retain their values during mode changes. Furthermore, we can allow a rich class of guards and cope with lazy sensors and actuators that have bounded delays associated with them.

Our work here is in a sense a generalization and in another sense specialization of the work reported in [12]. Finite precision was not demanded in [12] but the guards were required to be *rectangular.* In contrast we permit here the guards to be far more general. We do not know at present whether the finite precision assumption can be dropped for our linear hybrid automata though it is a natural one in the setting that we are considering. A closely related earlier work is [13] where the discrete time behavior of rectangular hybrid automata is studied but with the requirement that all instantaneous transitions should take place only at integer-valued instances of time. In our terms, [13] assumes that the sensors and actuators function with zero delays which considerably simplifies the analysis. On the other hand, [13] does not assume finite precision and yet establishes a decidability result for automata with triangular guards (i.e. conjunctions of constraints of the form $x - y \sim c$ where $\sim \in \{<, \leq, >, \geq\}$) and rectangular initial regions. It turns out that, as we observe in a later section, without finite precision but with zero delays sensors and actuators and *linear constraints* we can show a corresponding decidability result. As its title suggests, [13] is concerned with controller synthesis problems too. By viewing our automata as suitable open systems, we can also tackle controller synthesis problems using standard techniques.

Though not directly related to our work here, there have been a number of previous attempts to reduce the expressive power of timed and rectangular automata by taking away their ability to define trajectories with infinite precision [14–16]. Typically one demands, the set of admitted trajectories to be "fuzzy"; if a trajectory is admitted by the automaton then it should also admit trajectories that are sufficiently close to the trajectory where "closeness" is captured using a natural topology over the trajectories. This does not lead to more tractability as shown in [15] and [16]. The key difference between our work and these previous works is that in our automata, the fuzziness lies in the gap between the observed continuous state based on which a mode change takes place and the actual continuous state that holds at that instant. Further, the actual rate at which a variable may be evolving at an instant may be different from the rate demanded by the current mode of the automaton.

In the next section, we formulate the model of lazy linear hybrid automata with finite precision. In section 3 we prove our main result, namely, the language of state sequences and action sequences generated by our automaton are regular and that finite state automata representing these languages can be effectively computed. In section 4 we discuss the restrictions placed on our automata and point out that many of them can be easily relaxed. We also point how our main result can be easily extended to a much richer class of guards. In the concluding section we discuss the prospects for extending the results reported here.

2 Lazy Linear Hybrid Automata

Fix a positive integer n and one function symbol x_i for each i in $\{1, 2, \ldots, n\}$. We view each x_i as a function $x_i : \mathbb{R}_{\geq 0} \mapsto \mathbb{R}$ with $\mathbb{R}$ being the set of reals and $\mathbb{R}_{\geq 0}$, the set of non-negative reals. We let $\mathbb{Q}$ denote the set of rationals.

A *linear constraint* is an inequality of the form $a_1 \cdot x_1 + a_2 \cdot x_2 + \ldots + a_n \cdot x_n \sim c$ where $a_1, a_2, \ldots a_n$ as well as c are rational numbers and $\sim \in \{<, \leq, >, \geq\}$. A *guard* is a finite conjunction of linear constraints. We let *Grd* denote the set of guards.

A *valuation* is just a member of $\mathbb{R}^n$. The valuation V will be viewed as prescribing the value $V(i)$ to each variable x_i. The notion of a valuation satisfying a guard is defined in the obvious way.

A *lazy linear hybrid automaton* is a structure
$\mathcal{A} = (Q, Act, q_{in}, V_{in}, D, \epsilon, \{\rho_q\}_{q \in Q}, \mathrm{B}, \longrightarrow)$ where:

- Q is a finite set of *control states.*
- *Act* is a finite set of *actions.*
- $q_{in} \in Q$ is the initial control state.

- $V_{in} \in \mathbb{Q}^n$ is the initial valuation.
- $D = \{g, \delta_g, h, \delta_h\} \subseteq \mathbb{Q}$ is the *set of delay parameters* such that $0 < g < g + \delta_g < h < h + \delta_h < 1$.
- ϵ is the *precision of measurement*, $\epsilon \in \mathbb{Q}$.
- $\rho_q \in \mathbb{Q}^n$ is a rate vector which specifies the rate $\rho_q(i)$ at which each x_i evolves when the system is in the control state q.
- $\mathrm{B} = \{v \mid B_{min} \leq v \leq B_{max}, B_{min}, B_{max} \in \mathbb{Q}\}$ is the *allowed range.*
- $\longrightarrow \subseteq Q \times Act \times Grd \times Q$ is a transition relation such that $q \neq q'$ for every (q, a, φ, q') in $\longrightarrow$.

We shall study the discrete time behavior of our automata. At each time instant T_k, the automaton receives a measurement regarding the current values of the x_i's. However, the value of x_i that is observed at T_k is the value that held at some $t \in [T_{k-1}+h, T_{k-1}+h+\delta_h]$. Further, the value is observed with a precision of ϵ. More precisely, any value of x_i in the half-open interval $[(l-1/2)\epsilon, (l+1/2)\epsilon)$ is reported as $l\epsilon$ where l is an integer. For any real number v, we will denote this rounded-off value relative to ϵ as $\langle v \rangle$.

If at T_k, the automaton is in control state q and the observed n-tuple of values $(\langle v_1 \rangle, \langle v_2 \rangle, \ldots, \langle v_n \rangle)$ satisfies the guard φ with (q, a, φ, q') being a transition, then the automaton may perform this transition instantaneously by executing the action a and move to the control state q'. As a result, as usual, the x_i's will cease to evolve at the rates ρ_q and instead start evolving at the rates $\rho_{q'}$. However, this change in the rate of evolution will not kick in at T_k but at some time $t \in [T_k + g, T_k + g + \delta_g]$. In this sense, both the sensing of the values of the x_i's and the rate changes associated with mode switching take place in a lazy fashion but with bounded delays. We expect g to be close to 0, h to be close to 1 and both δ_g and δ_h to be small compared to 1.

Thus in the idealized setting, which we shall consider briefly later, the change in rates due mode switching would kick in immediately ($g = 0 = \delta_g$) and the value observed at T_k is the value that holds at exactly T_k ($h = 1$ and $\delta_h = 0$). In addition, assuming perfect precision would boil down to setting $\langle v \rangle = v$ for every real number v.

B specifies the range of values within which the automaton's dynamics are valid. The automaton gets stuck if any of the x_i's ever assume a value outside the allowed range $[B_{min}, B_{max}]$. A number of the restrictions that we have imposed are mainly for ease of presentation. Later, we will discuss how these restrictions can be relaxed.

Our main result is that the control state and action sequence languages generated by a lazy linear hybrid automaton are both regular. Furthermore, these languages can be computed effectively.

2.1 The Transition System Semantics

Through the rest of this section we fix a lazy linear hybrid automaton $\mathcal{A}$ as defined above and assume its associated notations and terminology. We shall often say “automaton” to mean “lazy linear hybrid automaton”. The behavior of $\mathcal{A}$ will be defined in terms of an associated transition system.

A *configuration* is a triple (q, V, q') where q, q' are control states and V is a valuation. q is the control state holding at the current time instant and q' is the control state that held at the previous time instant. V captures the *actual* values of the variables at the current instance. The configuration (q, V, q') is *feasible* iff $V(i) \in [B_{min}, B_{max}]$ for every i. The initial configuration is, by convention, the triple (q_{in}, V_{in}, q_{in}). We assume without loss of generality that the initial configuration is feasible. We let $C_{\mathcal{A}}$ denote the set of configurations. Since $\mathcal{A}$ will be clear from the context, we will often write C instead of $C_{\mathcal{A}}$.

As in the case of timed automata [17], a convenient way to understand the dynamics is to break up each move of the automaton into a time-passage move followed by an instantaneous transition. At T_0, the automaton will be in its initial configuration.

We assume that the unit of time has been fixed at some suitable level of granularity and that the rate vectors $\{\rho_q\}$ have been scaled accordingly. Suppose the automaton is in the configuration (q_k, V_k, q'_k) at T_k. Then one unit of time will pass and at time instant T_{k+1}, the automaton will make an instantaneous move by performing an action a or the silent action τ and move to a configuration $(q_{k+1}, V_{k+1}, q'_{k+1})$. The silent action will be used to record that no mode change has taken place during this move. Again, as often done in the case of timed automata, we will collapse these two sub-steps of a move (unit-time-passage followed by an instantaneous transition) into one “time-abstract” transition labelled by a member of Act or by τ.

With this scheme in mind, we now define the transition relation $\Longrightarrow \subseteq C \times (Act \cup \{\tau\}) \times C$ as follows.

- Let (q, V, q') and $(q1, V1, q1')$ be configurations and $a \in Act$. Then $(q, V, q') \overset{a}{\Longrightarrow} (q1, V1, q1')$ iff $q1' = q$ and there exists in $\mathcal{A}$ a transition of the form $q \xrightarrow{a,\varphi} q1$ and there exist $\widehat{t1} \in [g, g + \delta_g]^n$ and $\widehat{t2} \in [h,\, h + \delta_h]^n$ such that the following conditions are satisfied.
  (1) Let $v_i = V(i) + \rho_{q'}(i) \cdot \widehat{t1}(i) + \rho_q(i) \cdot (\widehat{t2}(i) - \widehat{t1}(i))$ for each i. Then $(\langle v_1 \rangle, \langle v_2 \rangle, \ldots, \langle v_n \rangle)$ satisfies φ.
  (2) $V1(i) = V(i) + \rho_{q'}(i) \cdot \widehat{t1}(i) + \rho_q(i) \cdot (1 - \widehat{t1}(i))$ for each i.
- Let (q, V, q') and $(q1, V1, q1')$ be configurations. Then

$$(q, V, q') \overset{\tau}{\Longrightarrow} (q1, V1, q1') \;\text{ iff }\; q1 = q1' = q$$

and there exists $\widehat{t1} \in [g,\, g + \delta_g]^n$ such that

$$V1(i) = V(i) + \rho_{q'}(i) \cdot \widehat{t1}(i) + \rho_q(i) \cdot (1 - \widehat{t1}(i)) \;\text{ for each } i.$$

Basically there are four possible transition types depending on whether $q = q'$ and $\alpha \in Act$. Suppose $(q, V, q') \overset{a}{\Longrightarrow} (q1, V1, q1')$ with $a \in Act$. Assume that $q \xrightarrow{a,\varphi} q1$ in $\mathcal{A}$ and $\widehat{t1} \in [g,\, g + \delta_g]^n$ and $\widehat{t2} \in [h,\, h + \delta_h]^n$ are as specified above. We first note that $q1 \neq q$ by the definition of the transition relation of $\mathcal{A}$. The requirement $q1' = q$ follows from our convention that $q1'$ is the control state that held in the previous instant and we know this was q.

First consider the case $q \neq q'$ and let us suppose that the configuration (q, V, q') holds at T_k. We take $q \neq q'$ to mean that a change of mode from q' to q has just taken place (instantaneously) at T_k based on the observations that were made available at T_k. However, at T_k, the automaton will continue to evolve at the rate dictated by $\rho_{q'}$. Indeed, each x_i will, starting from T_k, evolve at rate $\rho_{q'}(i)$ until some $T_k + t_1$ with $t_1 \in [g,\, g + \delta_g]$. It will then start to evolve at rate $\rho_q(i)$ until T_{k+1}. Consequently, at T_{k+1}, the value of x_i will be $V1(i) = V(i) + \rho_{q'}(i) \cdot t_1 + \rho_q(i) \cdot (1 - t_1)$. On the other hand, $q1 \neq q$ implies that another instantaneous mode change has taken place at T_{k+1} based on the measurements made in the interval $[T_k + h,\, T_k + h + \delta_h]$. Suppose x_i was measured at $T_k + t_2$ with $t_2 \in [T_k + h,\, T_k + h + \delta_h]$. Then in order for the transition $q \xrightarrow{a,\varphi} q1$ of $\mathcal{A}$ to be enabled at T_{k+1}, it must be the case that the observed values of x_i's at $T_k + t_2$ satisfy the guard φ. But then these values are $\langle v_i \rangle$ with $v_i = V(i) + \rho_{q'}(i) \cdot t_1 + \rho_q(i) \cdot (t_2 - t_1)$. This explains the demands placed on the transition $(q, V, q') \overset{a}{\Longrightarrow} (q1, V1, q1')$. It is worth noting that if $q = q'$ (i.e. no mode change has taken place at T_k) then $V1(i) = V(i) + \rho_q(i) \cdot t_1 + \rho_q(i) \cdot (1 - t_1) = V(i) + \rho_q(i)$ as it should be. Furthermore, $V(i) + \rho_q(i) \cdot t_1 + \rho_q(i) \cdot (t_2 - t_1) = V(i) + \rho_q(i) \cdot t_2$.

Similar (and simpler) considerations motivate the demands placed on transitions of the form $(q, V, q') \overset{\tau}{\Longrightarrow} (q1, V1, q1')$. Here again, it is worth noting that, in case $q = q'$, $V1(i)$ is determined uniquely, namely, $V1(i) = V(i) + \rho_q(i)$.

We now define the transition system
$TS_{\mathcal{A}} = (RC_{\mathcal{A}}, (q_{in}, V_{in}, q_{in}), Act \cup \{\tau\}, \Longrightarrow_{\mathcal{A}})$ via:

- $RC_{\mathcal{A}}$, the set of *reachable configurations* of $\mathcal{A}$ is the least subset of C that contains the initial configuration (q_{in}, V_{in}, q_{in}) and satisfies:
  Suppose (q, V, q') is in $RC_{\mathcal{A}}$ and is a feasible configuration. Suppose further, $(q, V, q') \overset{\alpha}{\Longrightarrow} (q1, V1, q)$ for some $\alpha \in Act \cup \{\tau\}$. Then $(q1, V1, q) \in RC_{\mathcal{A}}$.
- $\Longrightarrow_{\mathcal{A}}$ is $\Longrightarrow$ restricted to $RC_{\mathcal{A}} \times (Act \cup \{\tau\}) \times RC_{\mathcal{A}}$.

We will often write RC instead of $RC_{\mathcal{A}}$ and write $\Longrightarrow$ instead of $\Longrightarrow_{\mathcal{A}}$. We note that a reachable configuration can be the source of a transition in $TS_{\mathcal{A}}$ only if it is feasible. Thus infeasible reachable configurations will be deadlocked in $TS_{\mathcal{A}}$.

A *run* of $TS_{\mathcal{A}}$ is a finite sequence of the form
$\sigma = (q_0, V_0, q_0')\, \alpha_0\, (q_1, V_1, q_1')\, \alpha_1\, (q_2, V_2, q_2') \ldots (q_k, V_k, q_k')$ where (q_0, V_0, q_0') is the initial configuration and $(q_m, V_m, q_m') \overset{\alpha_m}{\Longrightarrow} (q_{m+1}, V_{m+1}, q_{m+1}')$ for $0 \leq m < k$. The *st-sequence (state sequence)* induced by the run σ above is denoted as $st(\sigma)$ and it is the sequence $q_0 q_1 \ldots q_k$. On the other hand, the *act-sequence* induced by σ is denoted as $act(\sigma)$ and it is the sequence $\alpha_0 \alpha_1 \ldots \alpha_k$. We now define the languages $\mathcal{L}_{st}(\mathcal{A})$ and $\mathcal{L}_{act}(\mathcal{A})$ as :

- $\mathcal{L}_{st}(\mathcal{A}) = \{st(\sigma) \mid \sigma$ is a run of $\mathcal{A}\}$.
- $\mathcal{L}_{act}(\mathcal{A}) = \{act(\sigma) \mid \sigma$ is a run of $\mathcal{A}\}$.

Our main result is that $\mathcal{L}_{st}(\mathcal{A})$ is a regular subset of $Q^{\star}$ while $\mathcal{L}_{act}(\mathcal{A})$ is a regular subset of $(Act \cup \{\tau\})^{\star}$. Moreover, we can effectively construct finite state automata representing these languages. As a consequence, a variety of verification problems and controller synthesis problems for our automata can be effectively solved.

3 Proof of the Main Result

The transition guards in [12] were of the form $x_i = c$ for some constant c. This fact was critical for the finite division of B^n resulting in a finite automata. If the guards are more general then quantization of the continuous state space as done in [12] is not possible. We shall address this point again in the next section.

However the extra structure provided by the finite precision of observations comes to the rescue. As we show below, it enables us to generalize the proof idea of [12].

Let $\mathcal{A} = (Q, Act, q_{in}, V_{in}, D, \epsilon, \{\rho_q\}_{q \in Q}, \mathrm{B}, F, \longrightarrow)$ be a lazy automaton. We assume for $\mathcal{A}$, the terminology and notations defined in the previous section.

We shall generalize the proof strategy of [12]. Define Δ to be the largest positive rational number that *integrally* divides every number in the finite set of rational numbers $\{g, \delta_g, h, \delta_h, 1\}$. We next define Γ to be the largest positive rational number that *integrally* divides each number in the finite set of rational numbers $\{\rho_q(i) \cdot \Delta \mid q \in Q, 1 \leq i \leq n\} \cup \{B_{min}, B_{max}\} \cup \{V_{in}(i) \mid 1 \leq i \leq n\} \cup \{\frac{\epsilon}{2}\}$.

Let $\mathbb{Z}$ denote the set of integers. We now define the map
$\|\cdot\| : \mathbb{R} \rightarrow \mathbb{Z} \times \{I, S, \perp\}$ as follows.

- If $v \in (-\infty, B_{min})$, then $\|v\| = (k_{min} - 1, \perp)$ where $k_{min} \cdot \Gamma = B_{min}$.
- If $v \in (B_{max}, \infty)$, then $\|v\| = (k_{max}, \perp)$ where $k_{max} \cdot \Gamma = B_{max}$.
- Suppose $v \in [B_{min}, B_{max}]$ and $v = k\Gamma + \widehat{v}$ with $k \in \mathbb{Z}$ and $\widehat{v} \in [0, \Gamma)$. Then $\|v\| = (k, S)$ if $\widehat{v} = 0$, and $\|v\| = (k, I)$ if $\widehat{v} \neq 0$.

This map is extended in the obvious way to points in $\mathbb{R}^n$: $\|(v_1, v_2, \ldots, v_n)\| = (\|v_1\|, \|v_2\|, \ldots, \|v_n\|)$.

The map $\|\cdot\|$ can also be extended in a natural way to configurations. Denoting this extension also as $\|\cdot\|$, we define $\|(q, v, q')\|$ to be $(q, \|v\|, q')$. Let $\mathcal{D}_{\mathcal{A}} = \{\|c\| \mid c \in C_{\mathcal{A}}\}$. Clearly $\mathcal{D}_{\mathcal{A}}$ is a finite set and we will often write $\mathcal{D}$ instead of $\mathcal{D}_{\mathcal{A}}$. Our goal is to show that the equivalence relation over the reachable configurations RC of $\mathcal{A}$ induced by the map $\|\cdot\|$ in turn induces a right congruence of finite index over $Q^{\star}$.

We are now ready to tackle the main part of the proof.

Theorem 1. *Let $c1$ and $c2$ be two reachable configurations such that $\|c1\| = \|c2\|$. Suppose $\alpha \in Act \cup \{\tau\}$ and $c1'$ is a reachable configuration such that $c1 \overset{\alpha}{\Longrightarrow}_{\mathcal{A}} c1'$. Then there exists a reachable configuration $c2'$ such that $c2 \overset{\alpha}{\Longrightarrow}_{\mathcal{A}} c2'$ and $\|c1'\| = \|c2'\|$.*

Proof. Clearly $c1$ is feasible and since $\|c1\| = \|c2\|$, it follows that $c2$ is also feasible.

Let $c1$ and $c2$ be configurations at time instant t. Let us split the unit time interval in which $c1$ moves to $c1'$ into intervals of size Δ. We refer to these smaller intervals as *basic* intervals. By the choice of Δ, there will be an integral number, say m, of basic intervals in the unit time interval. Let $c1_0 = c1$, $c1_1, \ldots$, $c1_m$ be the configurations that hold at the end of each of these basic intervals when the starting configuration is $c1$. Configuration $c1'$ is obtained by making an instantaneous state transition from $c1_m$. Let $[t+u\Delta, t+(u+1)\Delta]$ be one such basic interval. In this interval, assuming that $[u\Delta, (u+1)\Delta]$ lies in the range of $[g, g+\delta_g]$ or $[h, h+\delta_h]$, one of the two types of events may occur:

Rate Change: For some $J \subseteq \{1, 2, \ldots, n\}$ and $\{t_j\}_{j \in J}$ with $t_j \in [0, \Delta]$, the rate of variable x_j changes at $t + u\Delta + t_j$ for each $j \in J$.

Valuation: For some $J \subseteq \{1, 2, \ldots, n\}$ and $\{t_j\}_{j \in J}$ with $t_j \in [0, \Delta]$, the value of the variable x_j is recorded at time $t + u\Delta + t_j$ for each $j \in J$.

While the first event affects the current configuration immediately (by making the variables change at different rates), the affect of the second event is at the end of unit interval when an instantaneous state transition is made based on the values recorded by the event. We now prove a lemma about the behavior across basic intervals that will be crucial in proof of the theorem.

Lemma 1. *Let $c1_u$ and $c2_u$ be two configurations with $\|c1_u\| = \|c2_u\|$ at the beginning of basic interval $[t + u\Delta, t + (u + 1)\Delta]$. For every $i \in \{1, 2, \ldots, n\}$ and for every time instant $t1_i \in [0, \Delta]$, there exists another time instant $t2_i \in [0, \Delta]$ such that the following holds:*

- *Suppose, starting from $c1_u$, a rate change is affected for x_i at time $t+u\Delta+t1_i$ and the valuation of x_i at the end of interval is $V1_{u+1}(i)$. Suppose, starting from $c2_u$, a rate change is affected for x_i at time $t+u\Delta+t2_i$ and the valuation of x_i at the end of the interval is $V2_{u+1}(i)$. Then $\|V1_{u+1}(i)\| = \|V2_{u+1}(i)\|$.*
- *Suppose, starting from $c1_u$, valuation $V1(i)$ is made for variable x_i at time $t+u\Delta+t1_i$ and its valuation at the end of the interval is $V1_{u+1}(i)$. Suppose, starting from $c2_u$, valuation $V2(i)$ is made for variable x_i at time $t+u\Delta+t2_i$, and its valuation at the end of the interval is $V2_{u+1}(i)$. Then, $\|V1_{u+1}(i)\| = \|V2_{u+1}(i)\|$ and $\|V1(i)\| = \|V2(i)\|$.*

Proof. Let $c1_u = (q_t, V1_u, q_t')$ and $c2_u = (q_t, V2_u, q_t')$ with $\|V1_u\| = \|V2_u\|$. Let $V1_u(i) = k\Gamma + \alpha_1$ with $k \in \mathbb{Z}$ and $0 \leq \alpha_1 < \Gamma$.

We first handle a simple case: If $\|V1_u(i)\| = (k, S)$ then $V1_u(i) = V2_u(i) = k\Gamma$. So setting $t2_i = t1_i$ will do the trick.

Suppose that $\|V1_u(i)\| = \|V2_u(i)\| = (k, I)$. We first handle the case when rate changes. Let (the initial rate of x_i) $\rho_{q_t}(i) = \frac{\ell}{\Delta}\Gamma$ with $\ell \in \mathbb{Z}$ and the changed rate be $\frac{\ell'}{\Delta}\Gamma$ with $\ell' \in \mathbb{Z}$. We get the following value of x_i at the end of basic interval when starting from configuration $c1_u$:

$$\begin{aligned} V1_{u+1}(i) &= V1_u(i) + \frac{\ell}{\Delta}\Gamma \cdot t1_i + \frac{\ell'}{\Delta}\Gamma \cdot (\Delta - t1_i) \\ &= k\Gamma + \alpha_1 + \ell'\Gamma + \frac{(\ell - \ell')}{\Delta}\Gamma \cdot t1_i. \end{aligned}$$

Thus, $\|V1_{u+1}(i)\|$ lies in the range $[k+\ell, k+\ell']$. Suppose we change the rate of x_i, when starting from configuration $c2_u$ after time t_i. Then,

$$V2_{u+1}(i) = k\Gamma + \alpha_2 + \ell'\Gamma + \frac{(\ell - \ell')}{\Delta}\Gamma \cdot t_i$$

where $\alpha_2 \in [0, \Gamma)$ since $\|V1_u(i)\| = \|V2_u(i)\|$. Therefore, $\|V2_{u+1}(i)\|$ also lies in the range $[k+\ell, k+\ell']$. Also, by appropriately choosing the value of t_i, we can make $\|V2_{u+1}(i)\|$ take any value in its range. It therefore follows that there always exists a t_i such that $\|V2_{u+1}(i)\| = \|V1_{u+1}(i)\|$.

Now suppose that a valuation is made instead of rate change. It is clear that $\|V2_{u+1}(i)\| = \|V1_{u+1}(i)\|$ as this situation is same as rate changing to itself above. Moreover, as configuration $c1_u$ moves to $c1_{u+1}$, the norm of the value of x_i varies smoothly between $\|V1_u(i)\|$ and $\|V1_{u+1}(i)\|$. The same holds as $c2_u$ moves to $c2_{u+1}$. Since $\|V1_u(i)\| = \|V2_u(i)\|$ and $\|V1_{u+1}(i)\| = \|V2_{u+1}(i)\|$, it follows that there will always be a time period in the basic interval during which the norm of the value of x_i, when started on configuration $c2_u$ will be equal to $\|V1(i)\|$. Fix any such time for valuation $V2(i)$ and we get $\|V2(i)\| = \|V1(i)\|$. □

The proof of the theorem now proceeds as follows: We have that starting from $c1_0 = c1$, the configuration sequence at the end of basic intervals is $c1_1$, $c1_2$, ..., $c1_m$ and there is an instantaneous transition from $c1_m$ to $c1'$. The above lemma shows that, starting from $c2_0 = c2$ with $\|c2\| = \|c1\|$, there exist configurations $c1_1$, $c2_2$, ..., $c2_m$ at the end of basic intervals such that $\|c1_j\| = \|c2_j\|$ for each j. Further, if a rate change is made during the transition from $c1$ to $c1'$, the same change is also made during the transition from $c2$ to $c2'$ (for different variables the rate may change in different basic intervals). Also, if a valuation $V1$ is made during transition from $c1$ to $c1'$ then a valuation $V2$ is made during transition from $c2$ to $c2'$ such that $\|V1\| = \|V2\|$. Finally, note that there is complete freedom to choose a time instant to make a rate change or valuation for each variable within the specified range and that time instants of different variables can be chosen independently. Hence it is acceptable that for different variables the rate change or valuation may occur in different basic intervals.

Now consider the instantaneous transition from $c1_m$ to $c1'$. This transition depends on the current state q_t, the valuation $V1$ and some constraint φ. By our assumption about finite precision of the observations, the observed values of variable x_i from the valuations $V1$ and $V2$ are $\langle V1(i)\rangle$ and $\langle V2(i)\rangle$ respectively. We now note:

Lemma 2. *For any v, $v' \in \mathbb{R}$, If $\|v\| = \|v'\|$ then $\langle v\rangle = \langle v'\rangle$.*

Proof. Let $v = k\Gamma + v_0$ and $v' = k'\Gamma + v'_0$ for $v_0, v'_0 \in [0, \Gamma)$. Since $\|v\| = \|v'\|$, $k' = k$. Since Γ divides $\frac{\epsilon}{2}$, let $\epsilon = 2\ell\Gamma$ and $k = k_1 \cdot 2\ell + k_0$ with $k_0 \in \{0, 1, \ldots, 2\ell - 1\}$. So, $v = k_1\epsilon + k_0\Gamma + v_0$ and $v' = k_1\epsilon + k_0\Gamma + v'_0$. We have: $k_0\Gamma + v_0 < \frac{\epsilon}{2} = \ell\Gamma$ iff $k_0\Gamma < \ell\Gamma$ (since $v_0 \in [0, \Gamma)$) iff $k_0\Gamma + v'_0 < \ell\Gamma$ (since $v'_0 \in [0, \Gamma)$). Therefore, $\langle v\rangle = \langle v'\rangle$. □

Since we know that $\|V1(i)\| = \|V2(i)\|$, the above lemma gives that $\langle V1(i)\rangle = \langle V2(i)\rangle$. Therefore, $(\langle V1(1)\rangle, \ldots, \langle V1(n)\rangle)$ satisfies φ iff $(\langle V2(1)\rangle, \ldots, \langle V2(n)\rangle)$ does. This implies that if $c2'$ is the resulting configuration after making an instantaneous transition from $c2_m$, then $\|c2'\| = \|c1'\|$. □

We now define the finite state automaton
$\mathcal{Z}_{\mathcal{A}} = (\mathcal{D}, (q_{in}, \|V_{in}\|, q_{in}), Act \cup \{\tau\}, \rightsquigarrow)$ where:

- $\mathcal{D} \subseteq Q \times (\mathbb{Z} \times \{\bot, S, I\})^n \times Q$, and
- the transition relation $\rightsquigarrow \subseteq \mathcal{D} \times (Act \cup \{\tau\}) \times \mathcal{D}$ is given by: $(q, \widehat{v}, q1) \overset{\alpha}{\rightsquigarrow} (q', \widehat{v'}, q1')$ iff there exist configurations $(q, V, q1)$ and $(q', V', q1')$ such that $(q, V, q1) \overset{\alpha}{\Longrightarrow} (q', V', q1')$ and $\|V\| = \widehat{v}$ and $\|V'\| = \widehat{v'}$.

In what follows, we will often write $\mathcal{Z}_{\mathcal{A}}$ as just $\mathcal{Z}$. Note that,we are setting all the states of $\mathcal{Z}$ to be its final states.

We define $\mathcal{L}_{st}(\mathcal{Z})$ to be the subset of $Q^\star$ as follows. A *run* of $\mathcal{Z}$ is a sequence of the form $(q_0, \widehat{v_0}, q_0')\alpha_0(q_1, \widehat{v_1}, q_1')\alpha_1 \ldots (q_m, \widehat{v_m}, q_m')$ where $(q_0, \widehat{v_0}, q_0') = (q_{in}, \|V_{in}\|, q_{in})$ and $(q_j, \widehat{v_j}, q_j') \overset{\alpha_j}{\rightsquigarrow} (q_{j+1}, \widehat{v_{j+1}}, q_{j+1}')$ for $0 \leq j < m$. Next we define $q_0 q_1 \ldots q_m \in \mathcal{L}_{st}(\mathcal{Z})$ iff there exists a run of $\mathcal{Z}$ of the form
$(q_0, \widehat{v_0}, q_0')\ \alpha_0\ (q_1, \widehat{v_1}, q_1')\ \alpha_1\ \ldots\ (q_m, \widehat{v_m}, q_m')$. Clearly $\mathcal{L}_{st}(\mathcal{Z})$ is a regular subset of $Q^\star$ and it does not involve any loss of generality to view $\mathcal{Z}_{\mathcal{A}}$ itself as a representation of this regular language.

Theorem 2. *$\mathcal{L}_{st}(\mathcal{A}) = \mathcal{L}_{st}(\mathcal{Z}_{\mathcal{A}})$ and $\mathcal{L}_{act}(\mathcal{A}) = \mathcal{L}(\mathcal{Z}_{\mathcal{A}})$ where $\mathcal{L}(\mathcal{Z}_{\mathcal{A}})$ is the regular subset of $(Act \cup \{\tau\})^\star$ accepted by $\mathcal{Z}_{\mathcal{A}}$ in the usual sense. (Note that all the states of $\mathcal{Z}_{\mathcal{A}}$ are final states.) Further, the automaton $\mathcal{Z}_{\mathcal{A}}$ can be computed in time $O(|Q|^4 \cdot 2^{2n} \cdot K^{3n} \cdot |Act|)$ where $K = \frac{(B_{max} - B_{min})}{\Gamma}$.*

Proof. To see that $\mathcal{L}_{st}(\mathcal{A}) = \mathcal{L}_{st}(\mathcal{Z})$ we first note that $\mathcal{L}_{st}(\mathcal{A}) \subseteq \mathcal{L}_{st}(\mathcal{Z})$ follows from the definition of $\mathcal{Z}_{\mathcal{A}}$. To conclude inclusion in the other direction, we will argue that for each run $(q_0, \widehat{v_0} = \|V_{in}\|, q_0')\,\alpha_0\,(q_1, \widehat{v_1}, q_1')\,\alpha_1 \ldots (q_m, \widehat{v_m}, q_m')$ of $\mathcal{Z}$ there exist $V_0, V_1 \ldots V_m \in \mathbb{R}^n$ such that
$(q_0, V_0, q_0')\ \alpha_0\ (q_1, V_1, q_1')\ \alpha_1 \ldots (q_m, V_m, q_m')$ is a run of $TS_{\mathcal{A}}$. And furthermore, $\|V_j\| = \widehat{v_j}$ for $0 \leq j \leq m$. The required inclusion will then follow at once. For $m = 1$, it is clear from the definitions and so suppose that
$(q_0, \widehat{v_0}, q_0')\ \alpha_0\ (q_1, \widehat{v_1}, q_1')\ \alpha_1 \ldots (q_m, \widehat{v_m}, q_m')\ \alpha_m\ (q_{m+1}, \widehat{v_{m+1}}, q_{m+1}')$ is a run of $\mathcal{Z}$. By the induction hypothesis, there exists a run
$(q_0, V_0, q_0')\,\alpha_0\,(q_1, V_1, q_1')\,\alpha_1 \ldots (q_m, V_m, q_m')$ of $TS_{\mathcal{A}}$ with the property, $\|V_j\| = \widehat{v_j}$ for $0 \leq j \leq m$.

Now $(q_m, \widehat{v_m}, q_m') \overset{\alpha_m}{\rightsquigarrow} (q_{m+1}, \widehat{v_{m+1}}, q_{m+1}')$ implies that there exist V_m' and V_{m+1}' such that $(q_m, V_m', q_m') \overset{\alpha_m}{\rightsquigarrow} (q_{m+1}, V_{m+1}', q_{m+1}')$ and $\|V_m'\| = \widehat{v_m}$ and $\left\|V_{m+1}'\right\| = \widehat{v_{m+1}}$. But this implies that $\|V_m'\| = \|V_m\|$. Hence by Theorem 1, there exists V_{m+1} such that $(q_m, V_m, q_m') \overset{\alpha_m}{\rightsquigarrow} (q_{m+1}, V_{m+1}, q_{m+1}')$ and moreover $\left\|V_{m+1}'\right\| = \|V_{m+1}\|$. Thus $\mathcal{L}_{st}(\mathcal{A}) = \mathcal{L}_{st}(\mathcal{Z}_{\mathcal{A}})$. It now also follows easily that $\mathcal{L}_{act}(\mathcal{A}) = \mathcal{L}(\mathcal{Z}_{\mathcal{A}})$.

Let us now analyze the complexity of constructing the automata $\mathcal{Z}_{\mathcal{A}}$. We first estimate the size of the automaton. Each state of the automata is of the form

$(q, \widehat{v}, q')$ with $q, q' \in Q$ and $\widehat{v} \in (\{k_{min}, \ldots, k_{max}\} \times \{I, S\} \cup \{k_{min} - 1, k_{max}\} \times \{\perp\})^n$. Therefore, The number of states is $O(m^2 \cdot 2^n \cdot K^n)$ where $m = |Q|$ and $K = k_{max} - k_{min}$. For constructing the transitions, we need to check if there is a transition from $(q, \widehat{v}, q')$ to $(q1, \widehat{v1}, q)$ labeled with the action α. It is clear that the most complex case is when $\alpha \in Act$ and $q \neq q'$ and we need to check for the existence of at most $O(m^4 \cdot 2^{2n} \cdot K^{2n} \cdot |Act|)$ such possible transitions.

To decide if such a transition exists from $(q, \widehat{v}, q')$ to $(q1, \widehat{v1}, q)$ with $\widehat{v} = ((k_1, d_1), (k_2, d_2), \ldots, (k_n, d_n))$, $\widehat{v1} = ((k1_1, d1_1), (k1_2, d1_2), \ldots, (k1_n, d1_n))$, and a given symbolic transition in the lazy automaton of the form $(q, a, \varphi, q1)$ we need to check if there exists $\widehat{V}$ and $\widehat{t1}$ and $\widehat{t2}$ such that:

- For each i, $1 \leq i \leq n$:
$$k_i \cdot \mathit{\Gamma} < V(i) < (k_i + 1) \cdot \mathit{\Gamma},$$
- For each i, $1 \leq i \leq n$:
$$g \leq \widehat{t1}(i) \leq g + \delta_g,$$
- For each i, $1 \leq i \leq n$:
$$k1_i \cdot \mathit{\Gamma} < V(i) + \rho_{q'}(i) \cdot \widehat{t1}(i) + \rho_q(i) \cdot (1 - \widehat{t1}(i)) < (k1_i + 1) \cdot \mathit{\Gamma},$$
- For each i, $1 \leq i \leq n$:
$$h \leq \widehat{t2}(i) \leq h + \delta_h,$$
- For each i, $1 \leq i \leq n$, letting:
$$u_i = V(i) + \rho_{q'}(i) \cdot \widehat{t1}(i) + \rho_q(i) \cdot (\widehat{t2}(i) - \widehat{t1}(i)),$$
$(\langle u_1 \rangle, \langle u_2 \rangle, \ldots, \langle u_n \rangle)$ satisfies φ.

The above are $10n$ linear inequalities in $4n$ variables along with one constraint satisfaction check. The constraints themselves are linear inequalities and so one can use linear programming to check if there exists a feasible solution. However, the constraints require a "grid point" as a solution and to check if a grid point lies inside a convex region is NP-hard. So we need to spend exponential time ($= O(K^n)$) in checking if a grid point satisfies all the inequalities.

Therefore, the time complexity of the algorithm to construct the automata is $O(m^4 \cdot 2^{2n} \cdot K^{3n} \cdot |Act|)$. □

4 Limitations and Generalizations

Our construction also works for several generalizations of the problem. On the other hand, finite precision is really required in a number of settings.

4.1 The need for finite precision

The finite division of B^n in the presence of guards defined by general hyperplanes critically depends on the finite precision of the observation. For infinite precision, such a division is not possible. We defer the proof to the full version of the paper. We wish to emphasize that we *do not claim* that there is no finite state automaton recognizing the set of state-sequences in the presence of infinite precision. The only claim we make is that our way of constructing such an automaton will fail.

4.2 Infinite precision when $\boldsymbol{\delta_g = \delta_h = 0}$

There is one important case when infinite precision can be allowed: when we do not allow any uncertainty in the time at which rate change and valuation are made. In other words, when $\delta_g = \delta_h = 0$. (Note that g and h may be non-zero and so these events may still occur with non-zero delays). We sketch an argument to prove this below.

Compute $\varDelta$ and $\varGamma$ as before and split the space B^n into n-cubes of side length $\varGamma$. Call the vertices of these cubes *grid points*. It is now easy to see that the starting valuation of the system is at a grid point as well as valuation after each basic interval (of length $\varDelta$) remains at a grid point irrespective of the transitions that may occur. Therefore, there are only finitely many valuations possible and this immediately leads to a finite automaton. Notice that guards can be very general here, e.g., arbitrary polynomial surfaces.

In fact, even if the initial valuation is a rectangular region instead of a point in $\mathbb{R}^n$, the construction goes through after a slight modification. Now, instead of only grid points, we get a region around each grip point that may be a possible valuation. All these regions are identical to the initial region. However, we cannot always put points in a region into one equivalence class since a guard surface might intersect the region. We handle this in the following way.

Take a region around a grid point and suppose that the guards split it into k k disjoint pieces. Mirror this split into the copy of the region around every grid point. This increases the number of regions, however, note that now the guards do not intersect any region around the chosen grid point. Repeating this for all the grid points one-by-one splits the space into many more, but still finite, number of regions such that no region is intersected by any guard. After each basic interval, points in a single such region will evolve to points in another region of *identical* shape. So we can collect points in each region in one equivalence class and the automaton can be now be constructed as before. In fact, this argument will go through even for more complex regions.

4.3 Initial valuation region

We have assumed that the initial valuation is a point in $\mathbb{Q}^n$. A more general case is when initial valuation can be any point in an effectively presented (say, as a conjunction of linear constraints) convex region of $\mathbb{R}^n$. This is handled easily. Calculate the Γ as before (ignoring V_{in} now as there is none). The n-cubes of size length Γ will form an equivalence class as before. The initial region intersects some of these cubes. Let these be $\|V_1\|$, $\|V_2\|$, ..., $\|V_m\|$. Introduce a new initial state for $\mathcal{Z}_{\mathcal{A}}$ from which make a non-deterministic transition to each of $(q_{in}, \|V_i\|, q_{in})$. The rest of the automata remains as before. Actually we can permit the initial regions to be more complicated but what we have dealt with should do for now.

4.4 Generalizing guards

We have assumed the guards to consist of finite conjunctions of linear inequalities. However, at no point in the proof we actually made use of this property except when calculating the time taken to construct the automata $\mathcal{Z}_{\mathcal{A}}$. In fact, we can allow *any reasonable computable function* (polynomials for example) as guards and the construction goes through without any changes. The only difference is that construction of the automata $\mathcal{Z}_{\mathcal{A}}$ may take more time since one needs to check if a specific grid point satisfies a guard. Let T be the upper bound on the time needed to evaluate any guard. Then the time complexity of the construction will be $O(|Q|^4 \cdot 2^{2n} \cdot K^{3n} \cdot |Act| \cdot T)$.

4.5 Uncertainty in rounding off

We have assumed no uncertainty in rounding off valuations. In other words, we assumed that every number in the interval $[(k - \frac{1}{2})\epsilon, (k + \frac{1}{2})\epsilon)$ is observed as $k\epsilon$. A more realistic situation would be to assume some uncertainty even here. For example, every number in $((k - \frac{1}{2})\epsilon, (k + \frac{1}{2})\epsilon)$ is observed as $k\epsilon$ while $(k + \frac{1}{2})\epsilon$ can be observed as either $k\epsilon$ or $(k + 1)\epsilon$.

This can be handled with no change in the construction. The crucial observation is that numbers of the form $(k + \frac{1}{2})\epsilon$ have norm $(2k\ell + \ell, I)$ (assuming $\epsilon = 2\ell\Gamma$) since Γ integrally divides $\frac{\epsilon}{2}$. And as observed in the proof of Theorem 1, if $\|V1(i)\| = (k', I) = \|V2(i)\|$ for any two valuations $V1$ and $V2$, then $V1(i) = V2(i)$. Therefore, if there is uncertainty in rounding off $V1(i)$, there will be uncertainty in rounding off $V2(i)$ as well.

In fact, we can handle the more general case when there is a whole interval of uncertainty. Specifically, let η with $0 < \eta \leq \frac{1}{2}$ be such that any number in $((k - \eta)\epsilon, (k + \eta)\epsilon)$ is observed as $k\epsilon$ while any number in $[(k + \eta)\epsilon, (k + 1 - \eta)\epsilon]$ can be observed to be either $k\epsilon$ or $(k + 1)\epsilon$. For this case, we choose Γ such that it divides both ϵ and $\eta\epsilon$. Now if $\|V1(i)\| = \|V2(i)\|$ then either $V1(i) = V2(i) = (k \pm \eta)\epsilon$ (both uncertain), or $V1(i), V2(i) \in ((k+\eta)\epsilon, (k+1-\eta)\epsilon)$ (both uncertain), or $V1(i), V2(i) \in ((k - \eta)\epsilon, (k + \eta)\epsilon)$ (both certain). In each of the cases, the required equivalence holds.

4.6 Additional Relaxations

We can also permit the rates of evolution at a control location to range over a *rectangular region* instead of associating a single rate vector with each location. The proof of the main result will go through with minor modifications.

We have not studied carefully the effect of more complex flow constraints while being mindful of the undecidability result presented in [13] for triangular flow constraints. We can easily handle state invariants formulated using linear constraints and the delay parameters can be permitted to spill across more than one unit time interval. Finally, our construction will extend to a network of automata that synchronize on common actions.

5 Conclusion

We have formulated here a class of lazy linear automata. These are basically linear hybrid automata but where each automaton is accompanied by the delay parameters $\{g, \delta_g, h, \delta_h\}$ and a finite precision of measurement parameter ϵ. Our main result is that the discrete time behavior of these automata can be effectively computed if the allowed ranges of values for the variables are bounded.

We have not detailed the verification problems that can be settled effectively for these automata. It is however clear that we can model-check the discrete time behavior of our automata against a variety of linear time and branching time temporal logic specifications. We can also view (a subset of) the transitions of the automaton to be controllable and solve the problem of devising a switching strategy that can win against a given specification; again, these specifications can be a variety of linear time and branching time specifications.

We believe that associating non-zero bounded delays with the sensors and actuators and demanding that the values of the plant variables be observed with only finite precision are natural requirements. We also believe that it is useful to focus on the discrete time behavior of hybrid automata. As our results show, the pay-off is the ability to effectively solve a host of verification and controller synthesis problems for a rich class of hybrid automata.

Finally, based on the results reported here, there is some hope that a even larger class of lazy hybrid automata will turn out to be tractable in terms of their discrete time behaviors. We have in mind automata in which the dynamics of each mode is given by a simultaneous system of linear differential equations.

A related interesting problem which is open is to determine the border between the decidable and undecidable in the context of laziness, finite precision and discrete time semantics. Here the undecidability results reported in [5, 6] may provide the required technical tools.

References

1. Alur, R., Henzinger, T., Lafferriere, G., Pappas, G.: Discrete abstractions of hybrid systems. Proc. of the IEEE **88** (2000) 971–984
2. Henzinger, T.: The theory of hybrid automata. In: 11th LICS, IEEE Press (1996) 278–292
3. Asarin, E., Maler, O.: Achilles and the tortoise climbing up the arithmetical hierarchy. J. of Comp. and Sys. Sci. **57** (1998) 389–398
4. Asarin, E., Maler, O., Pnueli, A.: Reachability analysis of dynamical systems having piecewise-constant derivatives. Theoretical Comp. Sci. **138** (1995) 35–65

5. Blondel, V., Tsitsiklis, J.: Complexity of stability and controllability of elementary hybrid systems. Automatica **35** (1999) 479–489
6. Blondel, V., Bournez, O., Koiran, P. Papdamitrou, C., Tsitsiklis, J.: Deciding stability and mortality of piecewise affine dynamical systems. Theoretical Comp. Sci. **255** (2001) 687–696
7. Henzinger, T., Kopke, P., Puri, A., Varaiya, P.: What's decidable about hybrid automata? J. of Comp. and Sys. Sci. **57** (1998) 94–124
8. Henzinger, T.: Hybrid automata with finite bisimulations. In: 22nd ICALP, LNCS 944, Springer (1995) 324–335
9. Kesten, Y., Pnueli, A., Sifakis, J., Yovine, S.: Integration graphs: A class of decidable hybrid systems. In: Hybrid Systems, LNCS 736, Springer (1993) 179–208
10. McManis, J., Varaiya, P.: Suspension automata: A decidable class of hybrid automata. In: 6th CAV, LNCS 818, Springer (1994) 105–117
11. Puri, A., Varaiya, P.: Decidability of hybrid systems with rectangular differential inclusions. In: 6th CAV, LNCS 818, Springer (1994) 95–104
12. Agrawal, M., Thiagarajan, P.: Lazy rectangular hybrid automat. In: 7th HSCC, LNCS 2993, Springer (2003) 1–15
13. Henzinger, T., Kopke, P.: Discrete-time control for rectangular hybrid automata. Theoretical Comp. Sci. **221** (1999) 369–392
14. Gupta, V., Henzinger, T., Jagadeesan, R.: Robust timed automata. In: HART '97, LNCS 1201, Springer (1997) 331–345
15. Henzinger, T., Raskin, J.F.: Robust undecidability of timed and hybrid systems. In: HSCC '00, LNCS 1790, Springer (2000) 145–159
16. Ouaknine, J., Worrell, J.: Revisiting digitization, robustness and decidability for timed automata. In: 25th LICS, IEEE Press (2003) 198–207
17. Alur, R., Dill, D.: A theory of timed automata. Theoretical Comp. Sci. **126** (1994) 183–235