Decidability of Hybrid Systems with Rectangular Differential Inclusions*

Anuj Puri and Pravin Varaiya

Department of Electrical Engineering and Computer Science,
University of California, Berkeley, CA-94720

Abstract. A hybrid system is modeled with a finite set of locations and a differential inclusion associated with each location. We discuss a subclass of hybrid systems with constant rectangular differential inclusions. The continuous state of the system is $x \in \mathbb{R}^n$ with x_i evolving with differential inclusion $\dot{x}_i \in [L_i, U_i]$ where L_i, U_i are integers (i.e., the slope of trajectory of x_i could be changing, but is restricted to remain within $[L_i, U_i]$). A transition from one location to another can be made provided the state satisfies the enabling condition for the transition. The state can also be initialized to a new value during the transition. The differential inclusion for x_i can be changed when x_i is an integer or when x_i is initialized to a new value. We show that the verification problem for this class of hybrid systems is decidable. With this approach, systems with unsynchronized and drifting clocks can be modeled, a general differential equation can be abstracted by breaking the state space into regions with constant differential inclusions, and many previously presented hybrid system examples can be verified.

1 Introduction

Hybrid systems are modeled as automata with a finite set of locations and continuous state $x \in \mathbb{R}^n$. There is a differential inclusion at each location and the edges between locations have enabling conditions [1, 12, 13, 16]. The hybrid system starts in a specified location with an initial condition $x_0 \in \mathbb{R}^n$. The continuous trajectory evolves according to the differential inclusion associated with that location. At some time t, $x(t)$ satisfies the enabling condition for a transition and a jump is made to a new location. The state x could be initialized to a new value during the jump. At the new location, x begins evolving according to the differential inclusion associated with that location. After some time, another transition is made, and so on.

In this paper, we study hybrid systems with constant differential inclusions of the form $[L_1, U_1] \times \cdots \times [L_n, U_n]$ where L_i, U_i are integers. The continuous state of the system, $x \in \mathbb{R}^n$, evolves according to $\dot{x}_i \in [L_i, U_i], i = 1, \cdots, n$ (i.e., the slope of trajectory x_i could be changing but is restricted to remain within

* Research supported by NSF under grants ECS 9111907 and IRI 9120074, and by the PATH program, University of California, Berkeley

$[L_i, U_i]$). These are also called Bounded-Rate Automata in [4]. A transition from one location to another can be made provided the state satisfies the enabling condition for the transition. A transition to a new location with different differential inclusion can be made only when x_i is an integer value or when x_i is initialized. During the transition, the state can be initialized to a new value. We show that under these conditions, interesting verification problems for the hybrid system are decidable. In particular, we show that the languages generated by our hybrid automata are regular. Our model does not include integration graphs [10], since we permit the differential inclusion for x_i to change only when x_i is initialized, or when x_i is an integer.

Systems with clocks [6, 2, 3, 9, 8, 15], where $\dot{x}_i = 1$, are special cases of the hybrid systems we consider. With our approach, systems with unsynchronized or drifting clocks can be modeled, systems with differential equations can be abstracted by breaking the state space into regions with constant differential inclusions [16], and it follows that for many hybrid system examples [1, 13], there is a decision procedure that will terminate.

In Sect. 2, we introduce the sub-class of hybrid systems. In Sect. 3, we present the decidability results.

2 Hybrid Automata

2.1 Preliminaries

Notation $\mathbb{R}$ is the set of reals and $\mathbb{Z}$ is the set of integers. $\mathbb{R}^+$ is the set of non-negative reals and $\mathbb{Z}^+$ is the set of non-negative integers. For $X \subset \mathbb{Z}$, define $X\Delta = \{k\Delta | k \in X\}$. For an interval $[a, b]$, where $a, b \in \mathbb{Z}$ and $\frac{1}{\Delta} \in \mathbb{Z}^+$, define $[a, b]_\Delta = \{a, a + \Delta, \ldots, b\}$. For $x \in \mathbb{R}$, define $\lfloor x \rfloor_\Delta = k\Delta$ where k is the largest integer for which $k\Delta \leq x$. In this paper, we always take the floor with respect to Δ, so we write $\lfloor x \rfloor$ instead of $\lfloor x \rfloor_\Delta$.

Differential Inclusion A differential equation is $\dot{x} = f(x)$ where $x \in \mathbb{R}^n$ and $f : \mathbb{R}^n \to \mathbb{R}^n$. A solution to the differential equation with initial condition $x_0 \in \mathbb{R}^n$ is any differentiable function $\phi(t)$, where $\phi : \mathbb{R}^+ \to \mathbb{R}^n$ such that $\phi(0) = x_0$ and $\dot{\phi}(t) = f(\phi(t))$.

A differential inclusion is written as $\dot{x} \in f(x)$ where $x \in \mathbb{R}^n$ and f is a set-valued map from $\mathbb{R}^n$ to $\mathbb{R}^n$ (i.e., $f(x) \subset \mathbb{R}^n$). A solution to the differential inclusion with initial condition $x_0 \in \mathbb{R}^n$ is any differentiable function $\phi(t)$, where $\phi : \mathbb{R}^+ \to \mathbb{R}^n$ such that $\phi(0) = x_0$ and $\dot{\phi}(t) \in f(\phi(t))$.

A differential equation with a given initial condition has a unique solution (under Lipshitz conditions), whereas a differential inclusion has a family of solutions.

In this paper we consider constant differential inclusions $\dot{x} \in \beta$, $\beta = [L_1, U_1] \times \cdots \times [L_n, U_n]$ where $L_i, U_i \in \mathbb{Z}$ (i.e., $\dot{x}_i \in [L_i, U_i]$). We define $\mathcal{B}$ to be the set of all such constant differential inclusions.

Enabling Conditions Enabling conditions will be associated with edges between locations in the hybrid automaton. Similar to [2],we define Φ inductively to be the set of all enabling conditions:

$\delta := x_i \leq c | x_i \geq c | \delta_1 \wedge \delta_2 | \delta_1 \vee \delta_2$ where $c \in \mathbb{Z}$.

Enabling conditions are closed subsets of IR^n.

Setting the Initial State During a transition, some components of the state may be initialized to a new value. We associate an initialization relation $\lambda = (\lambda_1, \cdots, \lambda_n)$ with an edge where $\lambda_i = [l_i, u_i]$ or $\lambda_i = id$, and $l_i, u_i \in \mathbb{Z}$. When $\lambda_i = id$ (the identity relation), the value of x_i does not change during the transiton. But for $\lambda_i = [l_i, u_i]$, x_i is initialized non-deterministically to a value in $[l_i, u_i]$. For $x \in \mathrm{IR}^n$, define

$\lambda[x] = \{x' \in \mathrm{IR}^n | x'_i = x_i$ for $\lambda_i = id$, and $x'_i \in [l_i, u_i]$ for $\lambda_i = [l_i, u_i]\}$.

We define $\mathcal{S}$ to be the set of all initialization relations.

2.2 Syntax

A hybrid automaton $H = (L, \Sigma, D, \mathcal{I}, \psi)$ where L is a finite set of locations, Σ is a finite set of events, $D : L \rightarrow \mathcal{B}$ associates a differential inclusion with each location, $\mathcal{I} \subset L$ is a set of initial locations, and $\psi \subset L \times L \times \Sigma \times \Phi \times \mathcal{S}$ are the edge labels ($(l, l', \sigma, \delta, \lambda) \in \psi$ labels edge (l, l') with event σ, enabling condition δ, and initialization relation λ). We further require that the differential inclusion for x_i is changed only when x_i is initialized. That is, for edge label $(l, l', \sigma, \delta, \lambda)$, $d = D(l)$, $d' = D(l')$, $d_i = d'_i$ when $\lambda_i = id$.

Fig. 1 is an example of the kind of hybrid automaton we consider in this paper. Note that the differential inclusion for y is changed when making the transition from location C to location D because y is equal to -4 at the transition. That is the same as checking y is equal to -4, followed by initializing y to -4. However, the differential inclusion for x cannot change when going from location C to location D.

2.3 Semantics

The hybrid automaton starts at an initial location with state $x = 0$. At location l, the state x moves according to the differential inclusion $D(l)$. It can make a transition from location l to location l' with edge label $(l, l', \sigma, \delta, \lambda)$ provided $x \in \delta$. After the transition, the state is $x' \in \lambda[x]$ and the new differential inclusion is $D(l')$.

The state trajectory x moves in two phase steps [14]. In the first phase, time progresses and x changes continuously. In the second phase, a sequence of transitions is made instantaneously (Fig. 2). Formally, x is a multiple-valued function. It is defined on $[0, T_0], [T'_0, T_1], [T'_1, T_2] \ldots$ with $T'_i = T_i$. For $T'_i < T_{i+1}$, x is differentiable on $[T'_i, T_{i+1}]$. A transition is made at time T_i, with the state

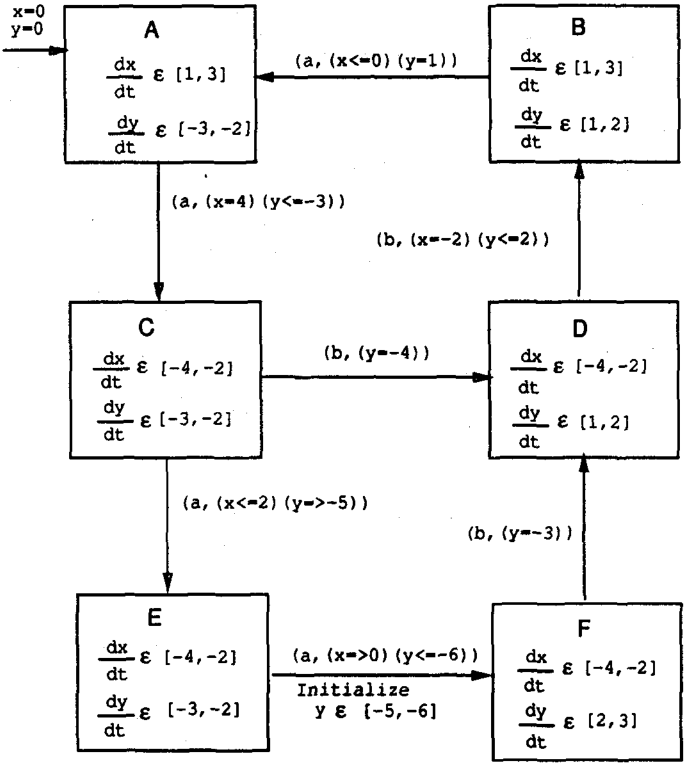

Fig. 1. Hybrid Automata with Rectangular Differential Inclusions

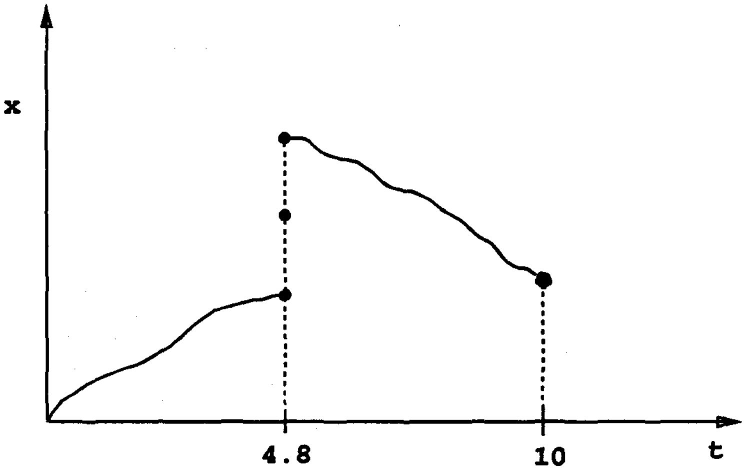

Fig. 2. Sample Trajectory

being $x(T_i)$ before the transition, and $x(T_i')$ after the transition. In Fig. 2, x is defined on $[0, 4.8], [4.8, 4.8], [4.8, 10]$, and it makes two successive transitions at time 4.8.

We associate a language $\mathcal{L}(H)$ with our hybrid automaton. Given $\sigma \in \Sigma^\omega$, we say $\sigma \in \mathcal{L}(H)$ provided there is a sequence of edges $< l_0, l_1, \sigma_0, \delta_0, \lambda_0 >$ $< l_1, l_2, \sigma_1, \delta_1, \lambda_1 > \cdots$ where $l_0 \in \mathcal{I}$ and $< D(l_0), \delta_0, \lambda_0 >< D(l_1), \delta_1, \lambda_1 > \cdots$ is consistent. We say $< D(l_0), \delta_0, \lambda_0 >< D(l_1), \delta_1, \lambda_1 > \cdots$ is consistent when there is a trajectory x for which this sequence of transitions is feasible.

Definition 1. A sequence $< D(l_0), \delta_0, \lambda_0 >< D(l_1), \delta_1, \lambda_1 > \ldots$ is continuous-time consistent provided there is a multiple-valued function $x : \mathbb{R}^+ \rightarrow \mathbb{R}^n$ and a sequence of intervals $[0, T_0], [T_0', T_1], [T_1', T_2] \ldots$ with $T_i' = T_i$ such that
1) $x(0) = 0$
2) $x(T_i) \in \delta_i$
3) $x(T_i') \in \lambda_i[x(T_i)]$
4) For $T_i' < T_{i+1}$, $\dot{x}(t) \in D(l_i)$ for $t \in [T_i', T_{i+1}]$

We similarly define the hybrid system which operates in discrete time according to a difference equation. Define

$$\frac{1}{\Delta} = LCM\{L_i, U_i | D_i(l) = [L_i, U_i] \text{ for } l \in L \text{ and } 1 \leq i \leq n\}$$

where LCM is the least common multiple of the set. For the example in Fig. 1, $\frac{1}{\Delta} = LCM\{1, 2, 3, 4\} = 12$.

Definition 2. A sequence $< D(l_0), \delta_0, \lambda_0 >< D(l_1), \delta_1, \lambda_1 > \ldots$ is discrete-time consistent provided there is a multiple-valued function $x : (\mathbb{Z}^+\Delta) \rightarrow (\mathbb{Z}\Delta)^n$ and a sequence of intervals $[0, T_0], [T_0', T_1], [T_1', T_2] \ldots$ where $T_i \in (\mathbb{Z}^+\Delta)$ and $T_i' = T_i$ such that
1) $x(0) = 0$
2) $x(T_i) \in \delta_i$
3) $x(T_i') \in (\lambda_i[x(T_i))])_\Delta$ (i.e., for $(\lambda_i)_j = [a, b], x_j(T_i') \in [a, b]_\Delta$, and for $(\lambda_i)_j = id, x_j(T_i') = x_j(T_i)$)
4) For $T_i' < T_{i+1}$, $x_j((n+1)\Delta) - x_j(n\Delta) \in D(l_i)_j\Delta$ for $T_i' \leq n\Delta, (n+1)\Delta \leq T_{i+1}$

We call H_C the hybrid system which operates in continuous time and H_D the system which operates in discrete time. For $\sigma \in \mathcal{L}(H_C)$, there is a sequence $< D(l_0), \delta_0, \lambda_0 >< D(l_1), \delta_1, \lambda_1 > \ldots$ which is continuous-time consistent. Similarly for $\sigma \in \mathcal{L}(H_D)$, there is a sequence $< D(l_0), \delta_0, \lambda_0 >< D(l_1), \delta_1, \lambda_1 > \ldots$ which is discrete-time consistent.

3 Decidability Results

Our main result is that $\mathcal{L}(H_C)$ is a regular language. To prove this, we follow an approach similar to [8, 15]. We first show that $\mathcal{L}(H_C) = \mathcal{L}(H_D)$ and then prove that $\mathcal{L}(H_D)$ is regular.

We show $\mathcal{L}(H_C) = \mathcal{L}(H_D)$ by proving a sequence $\rho =< D(l_0), \delta_0, \lambda_0 >< D(l_1), \delta_1, \lambda_1 > \ldots$ is discrete time-consistent iff it is continuous-time consistent. A discrete-time consistent sequence ρ has a discrete trajectory x_d. The continuous trajectory x_c obtained by linear interpolation from x_d also satisfies the constraints of ρ. Therefore, every discrete-time consistent sequence is also continuous-time consistent. The converse, that a continuous-time consistent sequence is also discrete-time consistent, is more difficult to prove. Lemma 3 states that a sequence ρ with continuous trajectory will also have a piece-wise linear trajectory satisfying it (Fig. 3). Lemma 4 shows that continuous trajectory x_i on $[t_j, t_{j+1}]$ with integer end points can be made into a trajectory on $[\lfloor t_j \rfloor, \lfloor t_{j+1} \rfloor]$ with same integer end points (Fig. 4). In Lemma 5 and Lemma 6, we show that for a sequence ρ and a continuous trajectory satisfying it at $T_0 \leq T_1 \leq T_2 \ldots$, there is another continuous trajectory which satisfies it at $\lfloor T_0 \rfloor, \lfloor T_1 \rfloor, \lfloor T_2 \rfloor, \ldots$. In Lemma 7, we finally show that there is a discrete trajectory x_d which also satisfies ρ.

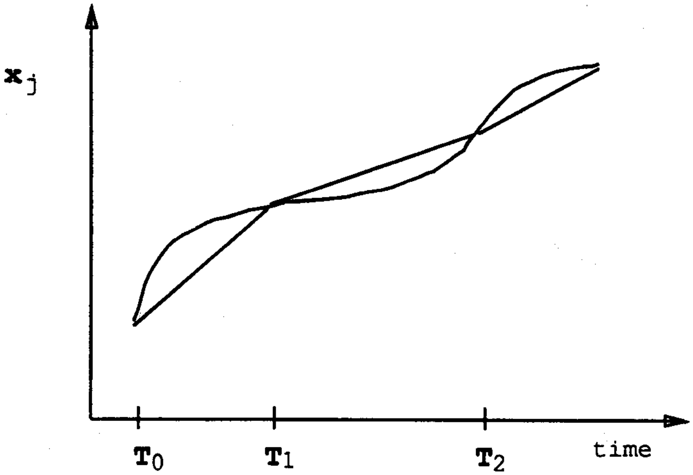

Fig. 3. Mean-Value Theorem

Lemma 3. *If a sequence $\rho =< D(l_0), \delta_0, \lambda_0 >< D(l_1), \delta_1, \lambda_1 > \ldots$ has a continuous trajectory, then it also has a piecewise linear trajectory (Fig. 3).*

Proof: Suppose x satisfies ρ at $T_0 \leq T_1 \leq T_2 \ldots$,then form x' by linear interpolation between points $x(0), x(T_0), x(T_1), \ldots$. From the Mean-Value theorem, it follows x' satisfies ρ.

Lemma 4. *Given a differentiable function x_i on $[t_j, t_{j+1}]$ with $\dot{x}_i \in [L, U]$ and $x_i(t_j) = k, x_i(t_{j+1}) = l$ where $k, l \in \mathbb{Z}$. We define x'_i on $[\lfloor t_j \rfloor, \lfloor t_{j+1} \rfloor]$ by linear interpolation between $x'_i(\lfloor t_j \rfloor) = k$ and $x'_i(\lfloor t_{j+1} \rfloor) = l$ (Fig. 4). Then $\dot{x}'_i \in [L, U]$.*

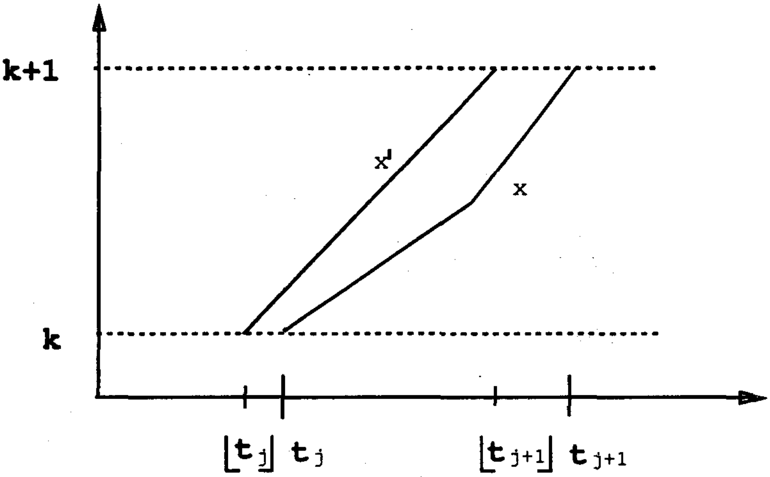

Fig. 4. Discretized Times for Trajectory with Integer End-Points

Proof: Since

$$L \leq \frac{x_i(t_{j+1}) - x_i(t_j)}{t_{j+1} - t_j} \leq U$$

for $U > 0$, we get

$$\frac{x_i(t_{j+1}) - x_i(t_j)}{U} = \frac{l-k}{U} = m\Delta \leq \lfloor t_{j+1} \rfloor - \lfloor t_j \rfloor$$

where $m \in \mathbb{Z}$. Since $x'_i(\lfloor t_{j+1} \rfloor) = x_i(t_{j+1})$ and $x'_i(\lfloor t_j \rfloor) = x_i(t_j)$, we get

$$\frac{x'_i(\lfloor t_{j+1} \rfloor) - x'_i(\lfloor t_j \rfloor)}{\lfloor t_{j+1} \rfloor - \lfloor t_j \rfloor} \leq U$$

Similar proof holds when $U \leq 0$. Similarly

$$L \leq \frac{x'_i(\lfloor t_{j+1} \rfloor) - x'_i(\lfloor t_j \rfloor)}{\lfloor t_{j+1} \rfloor - \lfloor t_j \rfloor}$$

Lemma 5. *Given a differentiable function x_i on $[b,c]$ with $\dot{x}_i \in [L,U]$, there is a function x'_i on $[\lfloor b \rfloor, \lfloor c \rfloor]$ with $\dot{x}'_i \in [L,U]$ such that for $j,k \in \mathbb{Z}$, $j \leq x_i(t) \leq k$ implies $j \leq x'_i(\lfloor t \rfloor) \leq k$ (Fig. 5).*

Proof: We look at the integer crossing points of x_i (Fig. 5). Using consecutive integer crossing points t_1, t_2, from Lemma 4, we obtain x'_i on $[\lfloor t_1 \rfloor, \lfloor t_2 \rfloor]$ where $\dot{x}'_i \in [L,U]$. For $t \in [t_1, t_2]$ and $j,k \in \mathbb{Z}$, $j \leq x_i(t) \leq k$ implies $j \leq x'_i(\lfloor t \rfloor) \leq k$. In case $x_i(b)$ is not an integer, we extend x_i to $a < b$ (keeping $\dot{x}_i \in [L,U]$) so that $x_i(a)$ is an integer and then reason as above. Similar reasoning applies if $x_i(c)$ is not an integer. After obtaining x'_i, we restrict it to $[\lfloor b \rfloor, \lfloor c \rfloor]$.

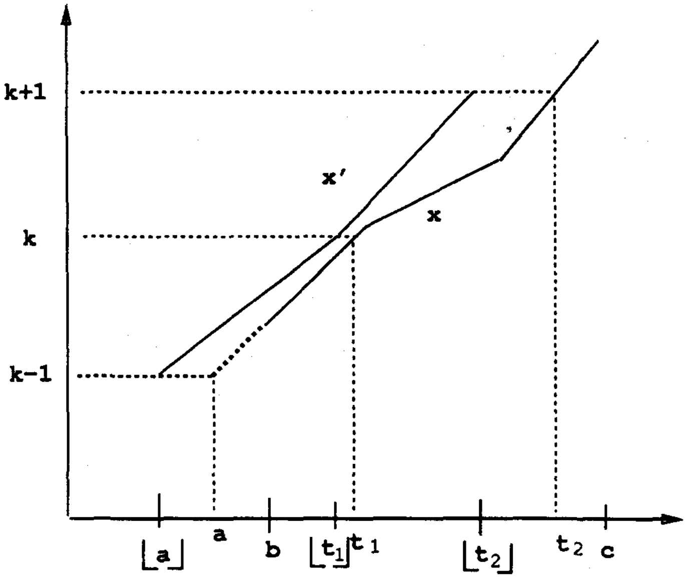

Fig. 5. Creating a Trajectory with Discretized Times

Lemma 6. *If a trajectory x satisfies ρ at times $T_0, T_1, T_2, \dots$, then there is a trajectory x' which satisfies ρ at times $\lfloor T_0 \rfloor, \lfloor T_1 \rfloor, \lfloor T_2 \rfloor, \dots$.*

Proof: Given x_i on $[0, T_{k_1}][T_{k_1}, T_{k_2}], \dots$ where T_{k_j} are times at which x_i is initialized and x_i is continuous on interval $[T_{k_j}, T_{k_{j+1}}]$ with $\dot{x}_i \in [L_j, U_j]$ (x_i maybe discontinuous because it could get initialized to a new value at T_{k_j}). The x_i' on $[0, \lfloor T_{k_1} \rfloor][\lfloor T_{k_1} \rfloor, \lfloor T_{k_2} \rfloor], \dots$ from Lemma 5 satisfies ρ at times $\lfloor T_0 \rfloor, \lfloor T_1 \rfloor, \lfloor T_2 \rfloor, \dots$. Since for $i = 1, \dots, n$, x_i' satisfies ρ at $\lfloor T_0 \rfloor, \lfloor T_1 \rfloor, \lfloor T_2 \rfloor, \dots$, we get x' satisfies ρ.

Lemma 7. *If a continuous trajectory x satisfies ρ at times $\lfloor T_0 \rfloor, \lfloor T_1 \rfloor, \lfloor T_2 \rfloor, \cdots$ then the discrete trajectory x_d where $(x_d)_i(k\Delta) = \lfloor x_i(k\Delta) \rfloor$ also satisfies ρ at times $\lfloor T_0 \rfloor, \lfloor T_1 \rfloor, \lfloor T_2 \rfloor \dots$.*

Proof: x_d satisfies the enabling conditions since $k \leq x_i(\lfloor T_j \rfloor) \leq l$ implies $k \leq \lfloor x_i(\lfloor T_j \rfloor) \rfloor \leq l$ for $k, l \in \mathbb{Z}$. Furthermore $(x_d)_i$ also satisfies the difference inclusion constraints because

$$L \leq \frac{x_i((k+1)\Delta) - x_i(k\Delta)}{\Delta} \leq U$$

implies

$$L \leq \frac{\lfloor x_i((k+1)\Delta) \rfloor - \lfloor x_i(k\Delta) \rfloor}{\Delta} \leq U$$

Theorem 8. $\mathcal{L}(H_C) = \mathcal{L}(H_D)$

Proof: $\sigma \in \mathcal{L}(H_D)$ ($\sigma \in \mathcal{L}(H_C)$) provided there is a sequence $\rho =< D(l_0), \delta_0, \lambda_0 >< D(l_1), \delta_1, \lambda_1 > \ldots$ which is discrete-time (continuous-time) consistent. As discussed at beginning of Sect. 3, every discrete-time consistent sequence is continuous-time consistent. From Lemma 3 - Lemma 7, every continuous-time consistent sequence is also discrete-time consistent. Thus $\sigma \in \mathcal{L}(H_D)$ iff $\sigma \in \mathcal{L}(H_C)$.

Theorem 9. *$\mathcal{L}(H_D)$ is a regular language.*

Proof: We will construct a finite state automaton which generates $\mathcal{L}(H_D)$. Let M_j be the largest integer with which x_j is compared or initialized, and m_j, the smallest such integer. Define $\Gamma_j = \{<\} \cup \{m_j, m_j + \Delta, \ldots, M_j\} \cup \{>\}$, $\Gamma = \Gamma_1 \times \cdots \times \Gamma_n$, and the finite set of states $Q = L \times \Gamma$. The finite state automaton $H_D = (Q, \Sigma, \rightarrow)$ where the transition relation $\rightarrow \subset Q \times \Sigma \times Q$ is defined as:

- $(l, v) \xrightarrow{\epsilon} (l, v')$ where $v'_j = "<"$ when $v_j = "<"$ or $v_j + w < m_j$ for some $w \in D(l)_j \Delta$; $v'_j = ">"$ when $v_j = ">"$ or $v_j + w > M_j$ for some $w \in D(l)_j \Delta$; $v'_j \in \{m_j, m_j + \Delta, \ldots, M_j\}$ when $v'_j - v_j \in D(l)_j \Delta$.
- $(l, v) \xrightarrow{\sigma} (l', v')$ provided $(l, l', \sigma, \delta, \lambda) \in \psi, v \in \delta$, and $v'_j \in (\lambda_j[v_j])_\Delta$.

The first part of the definition represents passage of Δ time and the second is an instantaneous transition.

It is not necessary to keep track of x_j when it exceeds M_j. This is clear when $\dot{x}_j \in [L_j, U_j]$ where $L_j, U_j \geq 0$ because once x_j exceeds M_j, it can become less than M_j only by being initialized. But it remains true even when $L_j \leq 0$ and $U_j \geq 0$ because any trajectory which exceeds M_j and then falls below it can be replaced by another which stays at M_j. Similar reasoning applies to m_j.

Theorem 10. *$\mathcal{L}(H_C)$ is a regular language*

Proof: From Theorem 8 and Theorem 9.

References

1. R.Alur, C.Courcoubetis, T.A. Henzinger and P.-H. Ho, Hybrid automata: an algorithmic approach to the specification and analysis of hybrid systems, *Workshop on Theory of Hybrid Systems*, Lyngby, Denmark, October 1992.
2. R. Alur and D. Dill, Automata for modeling real-time systems, *Proc. 17th ICALP*, Lecture Notes in Computer Science 443, Springer-Verlag, 1990.
3. R.Alur and T.Henzinger, Logics and models of real-time: A survey, *Proc. REX Workshop Real-Time: Theory in Practice*, Lecture Notes in Computer Science 600, Springer-Verlag, The Netherlands, June 1991.
4. R. Alur, T.A. Henzinger and P.-H. Ho, Automatic symbolic verification of embedded systems, *Proc. of the 14th Annual Real-time Systems Symposium*, IEEE Computer Society Press, 1993.
5. R.Alur, A.Itai, R.Kurshan and M.Yannakakis, Timing Verification by Successive Approximation, *Proc. 4th Workshop Computer-Aided Verification*, Lecture Notes in Computer Science 663, Springer-Verlag, 1992.
6. D. Dill, Timing Assumptions and verification of finite-state concurrent systems, *Automatic Verification Methods for Finite-State Systems*, Lecture Notes in Computer Science 407, Springer-Verlag, 1989.

7. F.Balarin and A.Sangiovanni-Vincentelli, A Verification Strategy for Timing-Constrained Systems, *Proc. 4th Workshop Computer-Aided Verification*, Lecture Notes in Computer Science 663, Springer-Verlag, 1992.
8. T. Henzinger, Z. Manna and A. Pnueli, What Good are Digital Clocks ?,*Proc. 19th ICALP*, Lecture Notes in Computer Science, Springer-Verlag, 1992.
9. T.A. Henzinger, X. Nicollin, J. Sifakis and S.Yovine, Symbolic model-checking for real-time systems, *Proc. 7th Symp. on Logics in Computer Science*, IEEE Computer Society Press, 1992.
10. Y. Kesten, A. Pnueli, J. Sifakis and S. Yovine, Integration graphs: a class of decidable hybrid systems, *Workshop on Theory of Hybrid Systems*, Lyngby, Denmark, October 1992.
11. O. Maler, Z. Manna and A. Pnueli, From Timed to Hybrid Systems, *Proc. REX Workshop on Real-Time: Theory in Practice*, Lecture Notes in Computer Science 600, Springer-Verlag, The Netherlands, June 1991.
12. J. A. Mcmanis, Verification and Control of Real-Time Discrete Event Dynamical Systems, Ph.D Thesis, Department of Electrical Engineering and Computer Science, University of California, Berkeley, 1993.
13. X.Nicollin, A.Olivero, J. Sifakis and S. Yovine, An Approach to the Description and Analysis of Hybrid Systems, *Workshop on Theory of Hybrid Systems*, Lyngby, Denmark, October 1992.
14. X. Nicollin and J. Sifakis, An overview and synthesis on timed process algebras,*Proc. 3rd Workshop Computer-Aided Verification*, Denmark, July 1991.
15. A. Puri, Real-Time Systems: Discrete Time vs. Dense Time, Unpublished, May 1993.
16. A. Puri and P. Varaiya, Modeling and Verification of Hybrid Systems, Preprint.
17. *Proc. REX Workshop on Real-Time: Theory in Practice*, Lecture Notes in Computer Science 600, Springer-Verlag, The Netherlands, June 1991.