Discrete Event Dynamic Systems: Theory and Applications, 8, 343–351 (1998)

Reduction of Timed Hybrid Systems

S. DI GENNARO
Dipartimento di Ingegneria Elettrica, Università di L'Aquila, 67040 Poggio di Roio, L'Aquila, Italy

C. HORN
Department of Electrical Engineering, Princeton University, Princeton, N.J., 08544

S. R. KULKARNI
Department of Electrical Engineering, Princeton University, Princeton, N.J., 08544

P. J. RAMADGE ramadge@ee.princeton.edu
Department of Electrical Engineering, Princeton University, Princeton, N.J., 08544

Abstract. We consider a class of hybrid dynamical systems and obtain conditions under which the behavior of these systems can be reduced to a finite state automaton. Specifically, we consider timed automata with more general enabling regions coupling the continuous and discrete dynamics than those previously considered. We provide a necessary condition for the existence of a finite state reduction, together with examples showing that this condition is not sufficient. We then give two sufficient conditions that provide a large class of systems with general enabling regions which admit finite reductions.

Keywords: hybrid systems, timed automata, reduction, finite state automaton

1. Introduction

A number of models for hybrid system consisting of both discrete and continuous dynamics have been proposed and analysis of the behavior of such systems has received considerable attention (see, for example, Stiver et al., 1994; Branicky et al., 1994; Brockett, 1994; Kohn and Nerode, 1993; Ramadge, 1990; and references therein). Although the dynamics of hybrid systems are generally quite complicated and the systems often difficult to analyze, certain classes of systems have been shown to have rather simple behavior.

In particular, a considerable amount of work has been done for a class of systems known as timed automata and the closely related class of systems known as hybrid automata (e.g., see Alur and Dill, 1990; Alur et al., 1993; Alur et al., 1994). Various results have been obtained on conditions under which the behavior of these systems can be reduced to a finite automaton. Using a temporal reduction technique, (Alur and Dill, 1990) showed that for integer rectangular partitions of the continuous state space, the behavior of a timed automaton can be reduced to a finite state automaton, and (Alur et al., 1990) obtained similar results under certain constraints for what they called timer region automata. (McManis and Varaiya, 1994) extended the timer region automata model and obtained reduction results for a subclass called suspension automata. (Puri and Varaiya, 1994) obtained reduction results for another class of systems with rectangular differential inclusions. These reductions are also related to the notion of finite bisimulations which has been studied by (Henzinger,

1995) and (Henzinger, 1996). A review of the information structures for a variety of setups is provided in (Deshpande and Varaiya, 1995).

In this note, we investigate a class of timed automata that allow quite general enabling regions. We obtain a necessary condition on the partitions for the existence of a finer congruence leading to a finite state reduction, and we provide two examples showing that this necessary condition is not sufficient. We also provide two sufficient conditions showing that a fairly broad class of timed automata admit a finite reduction. The approach we use is close to the reduction techniques in the works mentioned above and is also similar to other work on partition refinement such as (Kanellakis and Smolka, 1990; Paige and Tarjan, 1987; and Alur et al., 1992). Our results suggest that obtaining a full characterization of those enabling regions that admit a finite reduction in terms of geometric properties of the partition is difficult.

2. Timed Automata and Temporal Reductions

A timed automaton, as defined in (Alur and Dill, 1990), is a hybrid system $\mathcal{H}$ in which the continuous component consists of a finite set of integrators, called clocks, together with a set of projection operators, and the discrete component consists of a finite automaton. Specifically, the continuous flow is defined by the simple o.d.e. $\dot{x}_i = 1$, $i = 1, \ldots, n$. Under appropriate conditions (specified below), the continuous state can also undergo a projection (resetting of certain clocks). To this end, for $J \subseteq \{1, 2, \ldots, n\}$, let $P_J : \mathbb{R}^n \to \mathbb{R}^n$ be the linear projection with $P_J(x)_k = 0$ if $k \in J$ and $P_J(x)_k = x_k$ otherwise. The discrete component consists of an automaton $\mathcal{A} = (Q, \Sigma \cup \{\varepsilon\}, Q_i, E, Q_B)$. The set Q is a finite set of states with $Q_i \subseteq Q$ the set of initial states. Σ is a finite input alphabet not containing the null input ε. The set $E \subseteq Q \times Q \times (\Sigma \cup \{\varepsilon\})$ is the set of state transitions. An element $(q, q', \alpha) \in E$ represents a state change from q to q' with label α. The set Q_B is a subset of Q used to define the standard sequence acceptance condition for a Büchi automaton as briefly explained below.

The coupling of the continuous and discrete components is brought about by two additional constraints. First, a finite partition π on the continuous state space $\mathbb{R}^n$ is assumed to be given. Each transition of $\mathcal{A}$ is only allowed, or enabled, if the continuous state belongs to specified elements of the partition. Second, a set of projections is associated to each transition. When a transition in $\mathcal{A}$ is enabled by π and occurs, the continuous state x is projected using some P_J.

Let $x(t) = (x_1(t), \ldots, x_n(t))$ and let $D = (1, \ldots, 1)$ be a vector of length n. Then $x(t)$ evolves in $\mathbb{R}^n$ along the straight line $x(t) = x(0) + tD$ until when in the enabling region of a transition, that transition occurs and a projection may result.

Given an automaton $\mathcal{A}$ with accepting set Q_B, the language of $\mathcal{A}$ defined by the Büchi acceptance condition is the set of infinite input strings $\{\alpha_i\}$ such that the corresponding sequence of states $\{q_i\}$ satisfies $q_i \in Q_B$ for infinitely many i. For a timed automaton $\mathcal{H}$, a run of $\mathcal{H}$ is a finite or infinite sequence $(x_0, q_0, t_0, \alpha_0, J_0), (x_1, q_1, t_1, \alpha_1, J_1), \ldots$ subject to the obvious consistency constraints imposed by the coupled continuous and discrete dynamics of $\mathcal{H}$, where x_i and q_i denote the continuous and discrete states respectively, t_i denotes the time of a discrete state transition under input α_i, and J_i denotes the corresponding projection of the continuous state at the transition. The untimed language of $\mathcal{H}$ is defined to be the set of all infinite sequences $\{\alpha_i\}$ for which there exists an infinite run of $\mathcal{H}$ satisfying the Büchi acceptance condition—namely, $q_i \in Q_B$ for infinitely many i.

The untimed language of $\mathcal{H}$ characterizes the set of acceptable discrete inputs/controls to the system. Hence, questions regarding the existence of acceptable runs, verification of behavior, design of control strategies, etc., depend crucially on the structure of the untimed language. It is therefore of interest to determine when the untimed language possesses a simple structure, e.g., when it corresponds to the language of some finite state automaton.

The main idea in reducing a timed automaton $\mathcal{H}$ to a finite state automaton (i.e., showing that the untimed language of $\mathcal{H}$ is regular) is to introduce an equivalence relation on $\mathbb{R}^n$, which groups together states equivalent under time flow and projections, and then couple this to the underlying automaton $\mathcal{A}$. In (Alur and Dill, 1990), such a temporal reduction was carried out for the case in which the partition π is an integer rectangular lattice., but this idea can be extended to general finite partitions π.

Let π be a finite partition on the state space $\mathbb{R}^n$ of a generalized timed automaton. We say that a partition ω on $\mathbb{R}^n$ is finer than π if for each $x, y \in \mathbb{R}^n$, $x \equiv y \bmod(\omega)$ implies $x \equiv y \bmod(\pi)$. For any finite partition ω of $\mathbb{R}^n$, let Z_ω denote the quotient of $\mathbb{R}^n$ mod ω (a finite set) and let $z = [x]_\omega$ denote the equivalence class of $x \in \mathbb{R}^n$. For each $x \in \mathbb{R}^n$ the current class of x mod ω is $[x]_\omega$ and the next class of x mod ω is defined to be:

$$\begin{aligned} N(x) \;=\; z \in Z_\omega \text{ s.t. } \exists t > 0 \text{ with } [x + tD]_\omega = z \neq [x]_\omega \\ \text{and } \forall 0 \le t^{'} \le t \quad [x + tD]_\omega \in \{z, [x]_\omega\}. \end{aligned}$$

Intuitively, $N(x)$ is the next class of ω that the state trajectory of the integrator system enters under time evolution without projection. There is at least one coset of ω that will not have a successor.

The partition ω of $\mathbb{R}^n$ is said to be a congruence of the integrator system with resetting if for each $x, y \in \mathbb{R}^n$, $x \equiv y \bmod(\omega)$ implies $P_J(x) \equiv P_J(y) \bmod(\omega)$ for each projection P_J and $N(x) = N(y)$. In this case, for each $z \in Z_\omega$ with $z = [x]_\omega$ define $P_J(z) = [P_J(x)]_\omega$ and $\tau(z) = N(x)$. We say that a congruence is finite if it is a finite partition. The following proposition can be stated as "bisimulation implies language equivalence" which is a well known result (e.g., see (Henzinger, 1995; Henzinger, 1996). It also follows straightforwardly from the argument in (Alur and Dill, 1990).

PROPOSITION 1 *Given a timed automaton $\mathcal{H}$, if there exists a finite congruence ω that is finer than π then there is a finite automaton $\mathcal{D}$ such that the the language of $\mathcal{D}$ is equal to the untimed language of $\mathcal{H}$ (in which case we say that $\mathcal{H}$ admits a reduction to a finite state automaton).*

The extension to non-rectangular grid partitions allows treating some problems that might arise naturally such as enabling conditions based on norm constraints. In the next two sections, we present some conditions under which $\mathcal{H}$ admits a reduction to a finite state automaton in terms of the structure of the partition π.

3. A Necessary Condition

If a partition ω is not a congruence then either P_J or N does not preserve ω. We now define the notion of critical points to characterize this lack of preservation under projections and time flow, respectively.

DEFINITION 1 *Let $J \subset \{1, \ldots, n\}$. A point $(x_1, \ldots, x_n)$ such that $x_j = 0$ for $j \in J$ is a P_J*-**critical point** *w.r.t partition ω if $\exists y_j$, $j \in J$ s.t. for some $\epsilon > 0$ the ϵ-ball centered at z, where $z_j = x_j$, $j \notin J$; $z_j = y_j$, $j \in J$, is in the same equivalence class and is mapped under P_J to different equivalence classes.*

DEFINITION 2 *A line segment $\{(x_1 - t, \ldots, x_n - t) : t \in (0, T)\}$ is an N-critical line segment if for all points y on the line segment and for all $\epsilon > 0$ sufficiently small, the ϵ-ball centered at y is in the same equivalence class and is mapped under operation N to different equivalence classes. x is an N*-**critical point** *w.r.t partition ω if $x - tD$ is contained in an N-critical line segment $\forall t > 0$ sufficiently small and x is not contained in any N-critical line segment.*

Critical points induce finer partitions which in turn may induce new critical points. If at some stage no critical points are introduced, it means that the partition at that stage is a congruence. Also note that if $|J| < n-1$ then critical points will actually make up surfaces. For $J = \{1, \ldots, n\} - \{j\}$, we shall refer to a P_J-critical point also as a P_j-critical point.

PROPOSITION 2 *If at any stage during a reduction, there is a P_i-critical point $(0, \ldots, 0, a_i, 0, \ldots, 0)$ and a P_j-critical point $(0, \ldots, 0, a_j, 0, \ldots, 0)$ such that a_i and a_j are not rationally related then there is no finite congruence.*

Proof: Suppose a_i and a_j are not rationally related and without loss of generality assume that $a_i > a_j$. We consider projections onto the $i - j$ plane. Since $(a_i, 0)$ and $(0, a_j)$ are P_i- and P_j-critical points, respectively, after inverse projections it follows that (a_i, a_j) is an N-critical point. Consider the rectangle formed by the points $(0, 0)$, $(a_i, 0)$, (a_i, a_j), $(0, a_j)$. The N-critical point at (a_i, a_j) induces a refinement of the rectangle through an inverse flow. That is, a partition boundary is formed by the line joining (a_i, a_j) to $(a_i - a_j, 0)$ since $a_i > a_j$. This results in a P_i-critical point at $(a_i - a_j, 0)$, which requires a further refinement of the boundary by an inverse P_i-projection from this point. This in turn results in another N-critical point at $(a_i - a_j, a_j)$. This procedure is continued by alternating inverse flows (resulting in either P_i- or P_j-critical points) followed by inverse projections (resulting in N-critical points).

Since $a_i > a_j$, an inverse flow from an N-critical point at (a_i, α) will induce a P_i-critical point at $(a_i - \alpha, 0)$. An N-critical point at (β, a_j) will induce a P_j-critical point at $(0, a_j - \beta)$ if $\beta < a_j$ and will induce a P_i-critical point at $(\beta - a_j, 0)$ if $\beta > a_j$. Under each of these operations, a critical point of the form $(m_1 a_i + m_2 a_j, n_1 a_i + n_2 a_j)$ with m_1, m_2, n_1, n_2 integers produces another critical point of this type. That is, since the initial N-critical point (a_i, a_j) has integer coefficients, every induced P_i-critical point is of the form $(m_1 a_i + m_2 a_j, 0)$ and every P_j-critical point is of the form $(0, n_1 a_i + n_2 a_j)$.

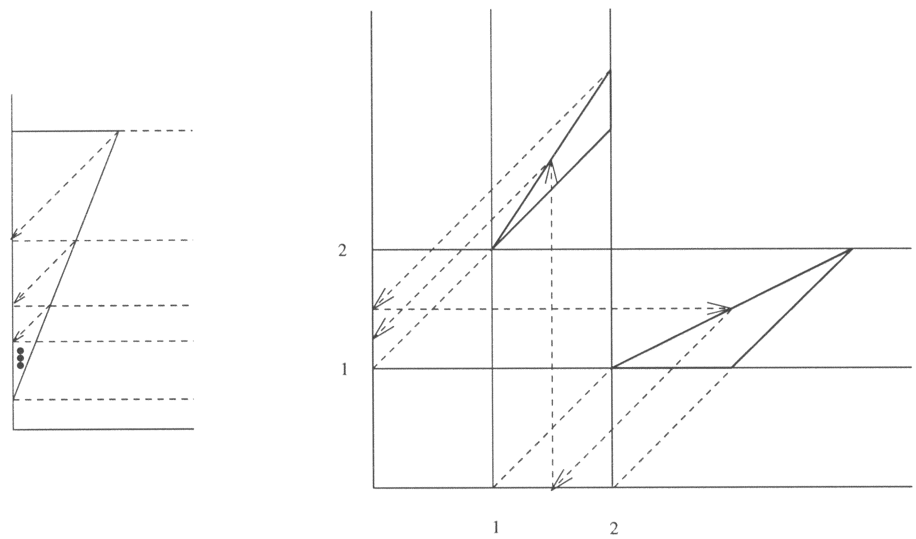

Figure 1. Examples with no finite congruence.

Now, suppose the congruence is finite. Then at some stage, the procedure above fails to produce critical points. Hence for some integers m_1, m_2 and $\tilde{m}_1, \tilde{m}_2$ with $(m_1, m_2) \neq (\tilde{m}_1, \tilde{m}_2)$ we have $m_1 a_i + m_2 a_j = \tilde{m}_1 a_i + \tilde{m}_2 a_j$. This contradicts the fact that a_i and a_j are not rationally related. ■

Proposition 2 shows that if a timed automaton admits a finite reduction, then all the critical points generated during the process of refining π must be rationally related. However, the following two examples in Figures (1a) and (1b) show that such a condition is not sufficient for the existence of a finite reduction. In Figure (1a) the partition consists of the outer triangle (with the slope of the hypotenuse not equal to one) which is taken to have rational corner points. In Figure (1b), the partition consists of the two bold faced triangles. The congruence in each case is constructed as shown in the figures. It can be seen that in both cases the procedures do not terminate in a finite number of steps, and yet by construction all of the critical points generated at each stage are rationally related.

4. Two Sufficient Conditions

4.1. Extended Rational Grids

Given a rectangular grid, in defining a partition it is possible to allow appropriate surfaces passing through two adjacent grid points and still have a congruence. We call such structures extended rational grid partitions.

DEFINITION 3 (Extended Rational Grids) *A finite partition π on $\mathbb{R}^n$, with cosets R_1, R_2, ..., R_m, is said to be an* extended rational grid *if it can be constructed through the following steps:*

1. *A rational rectangular grid is placed on $\mathbb{R}^n$ and each grid hypercube is divided into n! hyperpyramids having the D-directional line in common. Call this partition ω_g.*
2. *In each hyperpyramid of a grid hypercube, nonintersecting surfaces passing through the grid points and points on the D-directional line can be added. The surfaces divide the hyperpyramid portion of the hypercube into separate regions. The surfaces can be arbitrary except with the constraint that, under time flow, no point in any region can exit and then reenter the same region.*
3. *The boundaries of the cosets are constructed from hyperplanes perpendicular to an axis, hyperpyramids connected to a D-directional line, and the surfaces introduced in Step 2. (The surfaces from Steps 1 and 2 which are not explicitly used in the boundaries of the cosets can be removed.)*

PROPOSITION 4.1 *If π is an extended rational grid partition then there is a finite congruence ω that is finer than π.*

Proof: Consider the refinement, ω, of π given by $\omega = \omega_g \cap \pi$. We claim that ω is a finite congruence. To see this, we check that projections and time flow preserve ω. By construction, each of the points on the hyperplane perpendicular to the x_1 axis and between grid points is in the same equivalence class and similarly for each axis. Therefore, the projections preserve ω. As well, by Step 2 of the construction, time flow preserves ω. Therefore, both projections and time flow preserve ω and thus ω is a finite congruence that is finer than π. ■

Figure (2) provides examples of 2-d and 3-d extended rational grid partitions. The partitions have borders which consist of very general surfaces as long as they can be embedded on a grid and satisfy the constraint of the definition. These constraints allow a wide variety of enabling regions including, for example, norm constraints (which correspond to circular/spherical enabling regions).

4.2. Nested Clocks

Another class of partitions admitting a finite reduction allow even more general enabling regions subject to an ordering type of constraint on the clocks. This is formalized in the following definition.

DEFINITION 4 (Nested Clocks) *A finite partition π on $\mathbb{R}^n$ with cosets $R_1, R_2, \ldots, R_m$, is said to be a* nested clock partition *if $R_1, \ldots, R_m$ are polytopes, such that*
(i) $\{(x_1, \ldots, x_n) \mid x_1 \leq x_2 \leq \cdots \leq x_n\}^C \subset R_1$, *where* $\{\cdot\}^C$ *denotes complement, and*

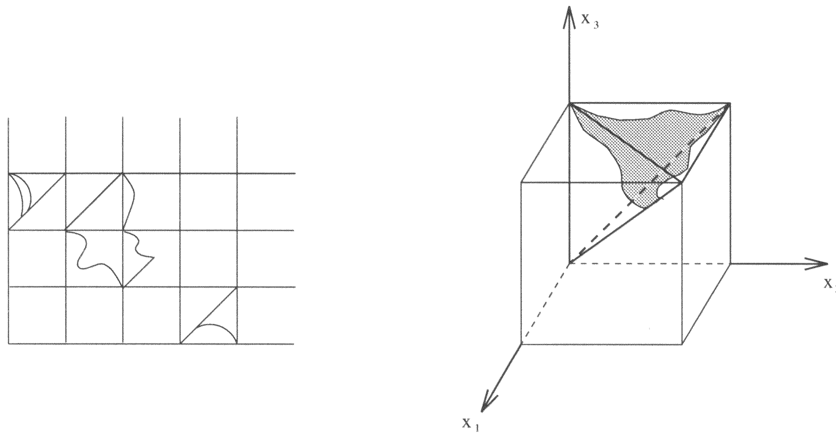

Figure 2. Examples of extended rational grid partitions.

(ii) for some $\delta > 0$ *we have* $\{(x_1, \ldots, x_n) \mid x_1 < \delta, x_1 < x_2 < \cdots < x_n\} \subset R_2$.

In the 2-d case, the restriction that the cosets $R_1, \ldots, R_m$ be polytopes can be dropped as long as we impose a regularity condition such as assuming the regions in π only contain boundaries whose intersection with any hyperplane consists of a finite number of connected components. The key property we need is that at each stage only a finite number of new N-critical points are created. Hence, in 2-d the nested clock condition amounts to having only one coset below the diagonal line $x_1 = x_2$ and having the boundaries of cosets above the diagonal line bounded away from the x_2 axis (e.g., see Figure (3)). In the multidimensional case, one could also generalize to non-polytope cosets as long as suitable regularity conditions are satisfied so that the key property mentioned above and in the proof below is satisfied.

PROPOSITION 4.2 *If* π *is a nested clock partition then there is a finite congruence* ω *that is finer than* π.

Proof: For simplicity, we restrict ourselves to the 2-d case, but the proof extends easily to multidimensions. The key point is that only a finite number of new critical points are created after each round of inverse flow followed by inverse projection, which is obtained through the restriction that the cosets $R_1, \ldots, R_m$ be polytopes (or by a suitable regularity assumption as mentioned above).

From conditions (i) and (ii), initially there are no P-critical points, and all N-critical points satisfy $x_2 > x_1$. The condition that $R_1, \ldots, R_m$ are polytopes results in only finitely many N-critical points and let b_0 denote the maximum value of the second component of the N-critical points. These N-critical points will induce only P_2-critical points by an

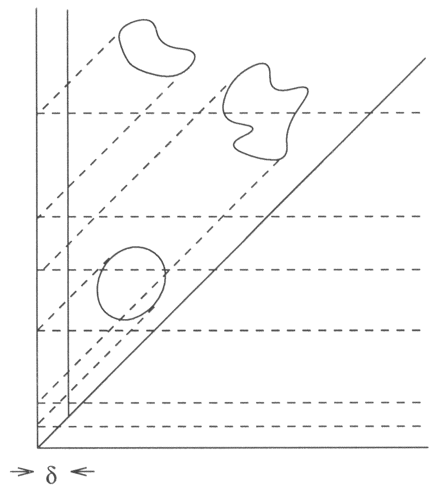

Figure 3. A nested clock partition in 2-D.

inverse flow. These in turn may induce a number of new N-critical points by an inverse projection. However, since $R_1, \ldots, R_m$ are polytopes there will only be a finite number of new N-critical points, and from condition (i) all the new N-critical points will also satisfy $x_2 > x_1$. Moreover, from condition (ii), the x_2 components of the critical points induced by the inverse flow and inverse projection will be at most $b_0 - \delta$. By repeating the argument, we see that after at most b_0/δ rounds of inverse flow followed by inverse projection, no critical points are induced. This results in a finite congruence finer than π.

■

Acknowledgements

Research partially supported by the National Science Foundation under grants ECS-9022634, ECS-9216450, IRI-9209577, and IRI-9457645, by the U.S. Army Research Office under grant DAAL03-92-G-0320, by EPRI under grant RP8030-18, and in the case of the first author by Italian ASI RS157/92 funds. Please address correspondence to the last author.

References

Alur, R., *et al.* 1992. Minimization of timed transition systems. *Lecture Notes in Computer Science 630,* R. J. Cleaveland, ed., Springer-Verlag, 340–354.

Alur, R., Courcoubetis, C., Henzinger, T. A., and Ho, P.-H. 1993. Hybrid automata: an algorithmic approach to the specification and verification of hybrid systems. *Hybrid Systems I*, Grossman *et al*., ed., Springer-Verlag, 209–229.

Alur, R., *et al*. 1994. The algorithmic analysis of hybrid systems. *Lecture Notes in Control and Information Sciences 199*, G. Cohen and J.-P. Quadrat, Springer-Verlag, 331–351.
Alur, R., and Dill, D. L. 1990. Automata for modeling real-time systems. *Proceedings of the 17th International Colloquium: Automata, Languages and Programming*, pp. 322–335.
Branicky, M. S., Borkar, V. S., and Mitter, S. K. 1994. A unified framework for hybrid control. *Proc. 33rd IEEE Conference on Decision and Control*, pp. 4228-4234.
Brockett, R. 1994. Language driven hybrid systems. *Proc. 33rd IEEE Conference on Decision and Control*, pp. 4210–4214.
Deshpande, A., and Varaiya, P. 1995. Information structures for control and verification of hybrid systems. *Proc. American Control Conference*, pp. 2642–2647.
Henzinger, T. A. 1995. Hybrid automata with finite bisimulations. *Lecture Notes in Computer Science 944*, Z. Fülöp and F. Gécseg, eds., Springer-Verlag, 324–335.
Henzinger, T. A. 1996. The theory of hybrid automata. *Proc. 11th Annual Symp. on Logic in Computer Science*, IEEE Computer Society Press, pp. 278–292.
Kanellakis, P. C., and Smolka, S. A. 1990. CSS expressions, finite-state processes, and three problems of equivalence, *Information and Computation* 86: 43–68.
Kohn, W., and Nerode, A. 1993. Models for hybrid systems: automata, topologies, controllability, observability. *Hybrid Systems*, R. Grossman *et al*., eds., Lecture Notes in Computer Science 736, Springer-Verlag, 297–316.
McManis, J., and Varaiya, P. 1994. Suspension automata: a decidable class of hybrid automata. *Proc. 6th Workshop on Computer Aided Verification*.
Paige, R., and Tarjan, R. E. 1987. Three partition-refinement algorithms. *SIAM J. Computing* 16-6: 973–989.
Puri, A., and Varaiya, P. 1994. Decidability of hybrid systems with rectangular differential equations. *Proc. 6th Workshop on Computer Aided Verification*.
Ramadge, P. J. 1990. On the periodicity of symbolic observations of piecewise smooth discrete-Time systems. *IEEE Tran. Automatic Control* 35: 807–813.
Stiver, J. A., Antsaklis, P. J., and Lemmon, M. D. 1994. Digital control from a hybrid perspective. *Proc. 33rd IEEE Conference on Decision and Control*, pp. 4241–4246.