Proceedings of the 36th
Conference on Decision & Control
San Diego, California USA • December 1997

Timed Automata and Robust Control: Can We Now Control Complex Dynamical Systems?

Michael D. Lemmon and Panos J. Antsaklis [1]
Dept. of Electrical Engineering
University of Notre Dame

Abstract

Two different approaches have emerged in recent years for the analysis and synthesis of hybrid control systems. One approach views the hybrid system as a set of concurrent computer processes whose execution is controlled by continuous variables generated by an external environment. The other approach studies hybrid systems as dynamical systems using familiar concepts of stability and robust performance. The development of a systematic framework for the analysis and synthesis of hybrid systems will require the integration of these two perspectives. The objective of this tutorial paper is to highlight some of the recent developments in computer science and control that provide insight into the integration of these two methods.

1 Introduction

In recent years there has been significant interest in hybrid systems whose signals take on values in a metric space such as $\Re^n$ as well as a finite set of symbols in which there may be no metric defined. Such systems arise frequently in the supervision of complex dynamical processes. In this case process supervision involves switching the system's structure between various operational modes so that continuous-state performance and logical behavioural specifications are satisfied. It is also convenient, however, to view such systems as switched dynamical systems where the supervisory aspect is secondary to the verification of such system theoretic properties as stability and robust performance. We therefore see that hybrid systems can be studied from two distinct viewpoints; as a supervised collection of real-time computer processes or as a switched dynamical systems.

These two approaches provide a set of tools for the analysis of hybrid systems which , in many respects, complement each other. The computer science approach focuses on high level supervision of the system and assumes highly abstracted models of the system's continuous-state dynamics. This approach means that these methods can efficiently verify whether or not the system satisfies behavioural specifications which are posed as formulae in a temporal logic. On the other hand, in abstracting away certain details of the hybrid system's continuous-state components, it becomes difficult to precisely control the process. Control theoretic methods clearly provide the tools allowing precision control of processes, but in a highly complex system, the useage of these tools may be prohibitively expensive. The development of a useful and efficient methodology for hybrid system design and analysis requires the integration of these two approaches.

The objective of this paper is to highlight some recent progress in both the computer science and control theory communities which point toward a potential integration of these two approaches. Recent results from the control systems community [Bran94] [Hou96] [Pet96] [Rant97] have provide sufficient characterizations of switched system stability which can be extremely conservative. A close examination of these results indicates that these sufficient tests for Lyapunov stability might be used in conjunction with timed automata models [Alur94] to provide a less conservative method for verifying hybrid system stability and performance. This paper identifies how the integrated use of computer science modeling methods with recent switched stability results might provide a systematic framework for the verification of hybrid system stability and performance.

The remainder of this paper is organized as follows. Section 2 discusses timed and hybrid automata. Section 3 summarizes recent results in switched system stability and identifies the way in which these results might be integrated with timed/hybrid automata. Section 4 discusses some recent results on robust bounded amplitude performance and indicates how these results might also be integrated with timed/hybrid automata.

[1] We gratefully acknowledge the partial financial support of the Army Research Office (DAAH04-95-1-0600, DAAH04-96-10285) and the National Science Foundation (NSF-ECS95-31485)

2 Timed and Hybrid Automata

In [Cla81], it was shown that formulae in the computation tree logic, CTL, could be computed as fixed points of recursive functions. A consequence of this result was the later development of efficient algorithms for verifying whether or not a finite automaton satisfies a behavioural specification posed as CTL formulae [Clar86]. This verification procedure became known as symbolic model checking [McM93] and has provided a powerful tool in the verification of VLSI digital circuits [Bur90]. Timed [Alur94] and hybrid automata [Alur93] arose out of a desire to extend symbolic model checking to the verification of real-time systems. In recent years there has been considerable progress in the development of SMC tools for timed and hybrid automata [Alur96] [TAH95]. The purpose of this section is to introduce the timed and hybrid automaton as a model of a hybrid system.

A finite automaton is characterized by the ordered pair, $N = (V, A)$ where V is a finite set of vertices, $A \subset V \times V$ is a set of directed arcs between vertices. The automaton, N, is marked by a function $\mu : V \rightarrow \{0,1\}$. The marking function, μ is said to be valid if and only if there is at most one $p \in V$ such that $\mu(p) = 1$. We use the vector $\bar{\mu} = [\mu(p_1), \ldots, \mu(p_n)]$ to represent the state of the automaton. A marked automaton is then represented by the ordered triple, $(V, A, \bar{\mu}_0)$ where $\bar{\mu}_0$ is the initial marking vector of the automaton. We denote the preset, $\bullet a$, of an arc $a = (p,q)$ as the place $p \in V$. Similarly the postset of the arc, $a\bullet$ is the place $q \in V$. The preset of a place, p, is denoted as $\bullet p$ and consists of all arcs of the form $(q,p) \in A$ where $q \in V$. Similarly the postset, $p\bullet$, of a vertex consists of all arcs $(p,q) \in A$ where $q \in V$.

The dynamic behaviour of the automaton is generated by the firing of arcs. An arc $(p,q) \in A$ is said to be enabled if $\mu(p) = 1$. An enabled arc is free to fire. Let μ' and μ be the marking vectors of the automaton after and before the firing of arc (q_0, q_1), respectively. The relationship between these marking vectors is given by

$$\mu'(p) = \begin{cases} 1 & \text{if } p = q_1 \\ 0 & \text{otherwise} \end{cases} \qquad (1)$$

A *hybrid automaton* arises by introducing a set, $\mathcal{X}$, of dynamical systems which we refer to generically as *clocks* and by introducing functions which label the vertices and arcs of the automaton $N = (V, A)$ with equations representing constraints on the clock state. The ith clock will be characterized by the ordered triple $\mathcal{X}_i = (f_i, x_{i0}, \tau_{i0})$ where $x_{i0} \in \Re^n$, $\tau_{i0} \in \Re$, and $f_i : \Re^n \rightarrow \Re^n$. The *local time* of the ith clock, $x_i(\tau)$, ($\tau \geq \tau_{i0}$) generated by clock $\mathcal{X}_i$ is the solution to the following initial value problem

$$\frac{dx_i(t)}{dt} = f_i(x_i(t)) \;\; ; \;\; x_i(\tau_{i0}) = x_{i0} \qquad (2)$$

The set of all local times and clock rates at time τ will be called the *clock state* and will be denoted as

$$\bar{x}(\tau) = \{(x_i(\tau), \dot{x}_i(\tau))\}_{i=1,\ldots,N} \qquad (3)$$

Let $\mathcal{P}$ be a set of formulae defined over the clock state, $\bar{x}(\tau)$. We say that the clock state $\bar{x}(\tau)$ satisfies a formula $p \in \mathcal{P}$ if the formula is true for the current state assignment at time τ. This is denoted as $\bar{x}(\tau) \models p$. A hybrid automaton is formally defined by the ordered tuple, $(N, \mathcal{X}, \ell_f, \ell_r, \ell_v)$ where $N = (V, A, \mu_0)$ is a finite automaton with initial marking μ_0. $\ell_f : A \rightarrow \mathcal{P}$, $\ell_r : A \rightarrow \mathcal{P}$, and $\ell_v : V \rightarrow \mathcal{P}$ are functions labeling the arcs and vertices of N with formula from $\mathcal{P}$. These labels have the following meaning.

- $\ell_f(a)$ is called the firing condition. If the clock state $\bar{x} \models \ell_f(a)$ for an arc $a \in A$, then the arc a is free to fire provided it is already enabled.
- $\ell_v(p)$ is called the vertex constraint. It represents a constraint on $\bar{x}$ must be satisfied while vertex p is marked. If $\bar{x} \models \ell_v(p)$ then the clock states are forced to satisfy this equation while $\mu(p) = 1$. In general, we choose $\ell_v(p)$ to be an equality constraint on the clock rates, $\dot{x}_i$.
- $\ell_r(a)$ is called the reset constraint for arc a. This label represents an equality constraint which the clock state is reset to immediately after the firing of an arc.

The preceding definition of a hybrid automaton is basically the same as that used in [Alur93]. Our description, however, follows notational conventions found in the Petri net literature. When the hybrid automaton clocks rates are all constants (i.e. integrators) then the hybrid automaton is called a timed automaton [Alur94].

In a finite automaton, an arc, $a = (p,q) \in A$, can fire as long as $\mu(p) = 1$. We refer to this as a *logical* condition for firing. For a timed or hybrid automaton, however, the firing of an arc also requires that the clock state, $\bar{x}(\tau) \models \ell_f(a)$ for some time τ. We now want to examine the implication of having arc transitions enabled by such conditions. Let's assume that vertex p is currently marked. Figure 1 shows this vertex along with its input arcs and output arcs. The input arcs are denoted as $a_j^{(in)} = (q_j^{(in)}, p)$ where $q_j^{(in)} \in V$ for $j = 1, \ldots, N$. The output arcs are denoted as $a_j^{(out)} = (p, q_j^{(out)})$ for $q_j^{(out)} \in V$ for $j = 1, \ldots, M$. All of the output arcs of vertex p are logically enabled. If arc $a_j^{(out)}$ is to actually fire, then we need to ensure that the system generates a clock state, $\bar{x}(\tau)$, at some finite time τ such that $\bar{x}(\tau) \models \ell_f(a_j^{(out)})$. Whether or not this happens depends on the where the clock state was when vertex p was first marked, as well as the nature of the clock dynamics. These two conditions are determined by the reset conditions, $\ell_r(a_j^{(in)})$ on all arcs leading into p and the vertex labels $\ell_v(p)$ of vertex p. Therefore to determine the conditions under which $\bar{x} \models \ell_f(a)$, we need to examine the reset and vertex labels in more detail.

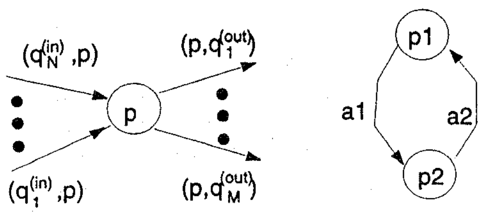

Figure 1: Vertex of a Hybrid Automaton

The reset constraint, $\ell_r(a)$ is an equality constraint on the clock states at the time arc a fires. We associate with the reset constraint a region

$$\Gamma(\ell_r(a) = \{\bar{x} : \bar{x} \models \ell_r(a)\} \tag{4}$$

This set represents the possible clock state when the vertex $p \in \bullet a$ is marked. We also associate a region with the firing condition $\ell_f(a)$. This region has the form

$$\Gamma(\ell_f(a)) = \{\bar{x} : \bar{x} \models \ell_f(a)\} \tag{5}$$

Since there are several arcs can lead into vertex, p, the total set of possible clock states when vertex p is marked will simply be

$$\Omega_{pre}(p) = \bigcup_{a \in \bullet p} \Gamma(\ell_r(a)) \tag{6}$$

We call this set the precondition for firing any output arc of p. After marking vertex p, the clock states are then constrained to satisfy $\ell_v(p)$. Recall that this equality constraint is placed on the clock rates, so we are effectively fixing the dynamics of the clocks while $\mu(p) = 1$. We denote as $R_a(x_0)$, the set of all clock states reached from initial clock state x_0 assuming that arc a has fired. Since we've already represented the set of initial clock state as $\Omega_{pre}(p)$, we can conclude that the total set of clock states that can be reached by a firing of arc a_0 will be

$$\Omega_{post}(a_0) = \bigcup_{a \in \bullet p} \{\bar{x} : (\bar{x}_0 \in \Omega_{pre}(p)) \wedge \tag{7}$$

$$(p = \bullet a_0) \wedge (\bar{x} \in R_a(x_0))\} \tag{8}$$

This set, $\Omega_{post}(a)$, is called the postcondition of arc a. Since $\Omega_{post}(a)$ represents the set of possible clock states after the firing of arc a, then a sufficient condition for the arc to fire is that this set lie in the region associated with the firing condition. In other words, a sufficient for the unconditional firing of arc a is that

$$\Omega_{post}(a) \subseteq \Gamma(\ell_f(a)) \tag{9}$$

This condition can be overly restrictive in many situations. We therefore study conditions sufficient for the repeated firing of an arc a. This requires an investigation of the cycles in the language generated by the timed automaton. A cycle is defined as a periodic sequence of arc firings. The simplest such cycle is shown in figure 1 where we have two vertices p_1 and p_2 connected by arcs $a_1 = (p_1, p_2)$ and $a_2 = (p_2, p_1)$. In this case, the sufficient conditions for firing the arcs in the cycle are that the postset of arc a_1 lie in the preset of arc a_2 and that the postset of arc a_2 lie in the prese of arc a_1. These conditions state that for the cycle, we require that

$$\Omega_{post}(a_1) \subseteq \Omega_{pre}(\bullet a_2) \tag{10}$$

$$\Omega_{post}(a_2) \subseteq \Omega_{pre}(\bullet a_1) \tag{11}$$

Note, however that $\Omega_{post}(a_1)$ depends on $\Omega_{pre}(a_2)$ also. This means that each of these equations sets up a recursive relationship in which a function of $\Omega_{pre}(a)$ must map to within a function of this preset again. The fixed point of this recursion will be denoted as $\Omega^*(a_1)$ and $\Omega^*(a_2)$. It has been shown that if such a fixed point exists, then the arc can fire infinitely often In the hybrid system literature an arc that has the capability of always firing will be said to be viable [Desh95]. If, in addition to this we can guarantee that the sum of the firing times $\sum_j \tau_j$ is unbounded, then we say that the cycle is nonZeno. Ideally we would like our cycles to be both viable and nonZeno.

In general, it is extremely difficult to verify the liveness conditions posed above. For some important classes of hybrid automata, however, these conditions can be efficiently decided. Specific algorithmic methods have been used to verify that these conditions are computable for timed automata [Alur94] and hybrid automata whose clock rates are bounded by rectangular differential inclusions [Alur96]. The decidability of these problems has been addressed in [TAH95a] and [Puri94].

Remarks: The verification results are analysis results and do not directly address issues of supervisor or controller synthesis. The restriction to hybrid automaton modeled by rectangular differential inclusions can result in extremely conservative performance in the resulting system. While the automaton model can be used to represent concurrent events, it is not as natural a model for concurrency as Petri nets. Some initial modeling efforts in this direction will be found in [Lem98].

3 Switched System Stability

In recent years, there has been considerable interest in developing systematic frameworks for verifying whether a switched dynamical system is Lyapunov stable [Pel91] [Bran94] [Hou96]. This recent work establishes sufficient conditions for switched system Lyapunov stability. This section summarizes these recent results and argues that the results in [Hou96] provide important insights on how to use automata theoretic methods in the formal verification of switched system stability.

Consider a set of functions, $\{V_i, i = 1, \ldots, N\}$. The elements of this set are called *candidate Lyapunov functions* if they are positive definite about the origin and if they have continuous partial derivatives. Consider a switched system whose jth system is represented by the differential equation

$$\dot{x} = f_j(x) \tag{12}$$

where $x \in \Re^n$ and $f_j : \Re^n \to \Re^n$ for $j = 1, \ldots, N$. We define a *switching sequence*, Σ, assuming an initial condition $x_0 \in \Re^n$ as a sequence of ordered pairs, (i_k, τ_k) $(k = 1, \ldots, \infty)$ where $1 \leq i_k \leq N$ is the index of the i_kth system turned on and $\tau_k \in \Re$ is the time this system was turned on. The switching sequence Σ can therefore be represented as

$$\Sigma = \{(i_0, \tau_0), (i_1, \tau_1), \cdots\} \tag{13}$$

In [Bran94] it is proven if the conditions, $\dot{V}_j(x(\tau)) \leq 0$ and $V_j(x(\tau_{j+1})) \leq V_j(x(\tau_j))$ hold for all switching sequences Σ where $\tau \in \Re$ and $1 \leq j \leq N$, then the origin is stable in the sense of Lyapunov. An extended stability result was obtained in [Hou96]. Let Σ_j denote the sequence of switching times when the j system is turned on or off.

$$\Sigma_j = \{\tau_{k_1}, \tau_{k_1+1}, \tau_{k_2}, \tau_{k_2+1}, \cdots, \tau_{k_n}, \tau_{k_n+1}, \cdots\} \tag{14}$$

and let $E(\Sigma_j)$ denote the sequence of switching times when the system is turned on,

$$E(\Sigma_j) = \{\tau_{k_1}, \tau_{k_2}, \cdots, \tau_{k_n}, \cdots\} \tag{15}$$

If V_j is monotonically nonincreasing on $E(\Sigma_j)$ for each switching sequence, Σ, generated by our switched system and if $\sup(\tau_{n+1} - \tau_n) < \infty$, then the equilibrium $x = 0$ of the switched system is stable in the sense of Lyapunov.

Remark: The theorem in [Bran94] provides the extension of Lyapunov theory to switched systems. Related results pertain to switched linear systems will be found in [Pel91]. The result in [Hou96] represents, in our opinion, a significant extension of the earlier work in [Bran94] and [Pel91]. [Hou96] essentially says that if the candidate Lyapunov functions can be ensured to be nonincreasing at the times when the jth subsystem is switched on, then the entire system is Lyapunov stable. The sequence of events between the times when the jth system switches on is essentially a *cycle* of events and this means that analysis of system stability only requires looking at the behaviour of candidate Lyapunov functions over these cycles. The original statement of theorems in [Bran94] and [Hou96] all assume that we can test the theorem's conditions over all switching sequences. In view of the above observation, however, it should be apparent that this is not really necessary. If the underlying cycles of the switching sequences can be determined then we only need test for the monotone nonincreasing nature of the candidate Lyapunov functions over these cycles.

The cycles within all switching sequences can be identified if we know something about the switching logic. Assume, for instance, that the switching logic is generated by either a finite automaton or a bounded Petri net. In both of these cases the reachability tree of the network is bounded which means that there is a finite number of fundamental cycles from which all of the switching sequences can be constructed. In other words, rather than having to test the stability condition, we only need to test the condition over a finite number of identified cycles. As an example of this idea, let's consider recent work [Pet96] [Rant97] concerned with the computation of candidate Lyapunov functionals satisfying the stability conditions in [Bran94]. Both of these papers consider switched linear time invariant systems of the form

$$\dot{x} = A_j x \tag{16}$$

where $A_j \in \Re^{n \times n}$, $j = 1, \ldots, N$. The switching rule for this particular system assumes that the jth system is used when the state x lies in a cone characterized by the symmetric indefinite matrix, Q_j where

$$x \in \{x : x' Q_j x \leq 0\} \tag{17}$$

Let's assume that the switch between the ith and jth subsystems occurs when

$$x \in \{x : x' Q_{ij} x \leq 0\} \tag{18}$$

where $Q_{ij} = Q_{ij}'$. In [Pet96] it is shown that a set of candidate Lyapunov functions of the form

$$V_j(x) = x' P_j x \tag{19}$$

$(j = 1, \ldots, N$ and $P_j = P_j' > 0)$ satisfying the conditions in [Bran94] can be generated by solving the following linear matrix inequality (LMI).

$$\begin{aligned} A_j' P_j + P_j A_j + \alpha_j Q_j &\leq 0, \quad (j = 1, \ldots, N) \quad (20) \\ (P_j - P_k) - \alpha_{kj} Q_{kj} &\leq 0, \quad (j, k = 1, \ldots, N) \quad (21) \end{aligned}$$

The candidate Lyapunov functions obtained in the preceding LMI assume that any switch is possible between various events. This assumption can easily yield overly restrictive stability conditions. Based on our above observations concerning the stability results in [Hou96] it appears that we can greatly simplify the resulting LMI by forming these LMI's over cycles generated by the system's assumed switching logic. This approach should yield a smaller set of LMI's which are more likely to yield feasible solutions as well as providing a less conservative test for system stability. Details of this approach are currently being studied [Lem98].

Remark: Before using cycles to verifying hybrid system stability, the cycles need to be identified. A brute force approach would involve constructing the reachability tree for the network and identifying the cycles from that. For concurrent automata and Petri nets, however, the construction of a reachability tree may be impractical since it requires an exhaustive search through the network's reachability tree. In many cases, a more efficient method would rely on the use of partial order methods to reduce the computational complexity associated with exploring a network's reachable markings [McM92] [Gode97].

4 Bounded Amplitude Performance

The preceding section discussed how network models such as automata or Petri nets could be used to simplify the stability analysis of a switched system. We now examine the performance of switched systems and see if any connections to hybrid automata can be discovered. Let's assume that the subsystems are now modeled by differential equations of the form

$$\dot{x} = A_j x + Bw \tag{22}$$

where x is the plant state and w is a bounded disturbance input for $j = 1, \ldots, N$. We assume a switching rule in the form of a hybrid automaton. The performance of the entire system can be characterized using temporal logic formulae determining whether or not the system state eventually jumps outside of a given set. These performance measures can be verified (or not) using model checking methods, but the answer provides little insight into how the system can be modified to ensure better performance. Traditional control theory does provide this insight, so we begin viewing the verification problem as a control theorist. In this case, we see that system performance is measured by a *bounded amplitude* performance measure. In particular, if we introduce a performance signal,

$$z = Cx \tag{23}$$

then we'll be interested in determining if $\sup_t \|z(t)\| < \gamma$ for a specified γ for all possible switched behaviours.

In an unswitched environment, sufficient conditions for bounded amplitude performance are readily obtained [Bett97]. Given a constant $\gamma > 0$ then the system exhibits bounded amplitude performance if there exists constants $\alpha > 0$ and $\beta \geq 0$ and a positive definite matrix $P \in \Re^{n \times n}$ satisfying

$$P \geq \frac{1}{\gamma^2} C'C \tag{24}$$

and

$$A'P + PA + (\alpha + \beta)P + \frac{1}{\alpha} PBB'P \leq 0 \tag{25}$$

If w is bounded and $x'(0)Px(0) \leq 1$, then we can show that $x'(t)Px(t) \leq 1$ and $z'(t)z(t) \leq \gamma^2$ for all time. Moreover, if $x'(0)Px(0) = r_0 > 1$, then $x'(t)Px(t) \leq 1$ for all $t > t_d$ where

$$t_d = -\frac{1}{\beta} \log\left(\frac{1}{r_0}\right) \tag{26}$$

This last quantity is called the *dwell time.*

The performance level of the switched system is guaranteed provided the Riccati inequality cited above holds, and provided the switching times are not shorter than the dwell time identified above. The Riccati inequalities can be reformulated as linear matrix inequalities in a manner [Bett97] analogous to the LMI's used in [Pet96] to characterize switched system Lyapunov stability. This work ensures robust stability for systems modeled as linear parameter varying systems. In our case, however, we also need to ensure that a dwell-time constraint is satisfied. This constraint of course, is identical to the firing constraints found in timed automata. It therefore seems quite possible to use a combination of robust control methodologies and timed automata analysis to verify the bounded amplitude performance of switched systems.

5 Concluding Remarks

This paper has surveyed recent results in switched system stability, performance, and verification. Can we now design complex hybrid systems? At this point the answer is "no". Current results in the area are more concerned with analysis or rather the verification of desirable system properties. In examining the various results that have emerged, it is apparent that there are strong connections between control and computer science theoretic methods which can have a great impact in addressing the weaknesses in each methodology. Control theoretic methods provide synthesis methods which can guarantee system performance but which are ill equipped, in general, to deal with the computational complexity associated with verifying large-scale switched systems. The automata theoretic methods appear to provide a computationally attractive framework for large scale verification, but do not provide synthesis methods that can guarantee system performance. The development of a systematic method for hybrid system design will require the integration of these two viewpoints into a tool which takes the best from each method and puts them together. This paper has identified one possible way in which such a synthesis might be done for the verification of hybrid system Lyapunov stability and bounded amplitude performance. So, while it does not appear that systematic design methods for hybrid systems exist, the development of such methodologies does not appear to be far off.

References

[Alur93] R. Alur, C. Courcoubetis, T.A. Henzinger, P-H Ho, Hybrid Automata: an algorithmic approach to the specification and verification of hybrid systems, *Hybrid Systems*, Springer LNCS 736, 1993, pp. 209-229.

[Alur94] R. Alur and D.L. Dill, A Theory of Timed Automata, *Theoretical Computer Science*, 126:183-235, 1994

[Alur96] R. Alur, T. Henzinger, and P-H Ho., Automatic Symbolic Verification of Embedded Systems, *IEEE Transactions on Software Engineering*, 22:181-201, 1996.

[Bett97] C.J. Bett and M.D. Lemmon, Bounded amplitude control of linear parameter varying systems, In Proceedings of American Control Conference (ACC'97), Albuquerque New Mexico, June 1997.

[Bett97a] C.J. Bett and M.D. Lemmon, Bounded Amplitude Control using Multiple Linear Agents, Technical Report ISIS-97-004, Dept. of Electrical Engineering, University of Notre Dame, Notre Dame, IN 1997.

[Bran94] M. Branicky, Stability of Switched and Hybrid Systems, In *Proceedings of the 33rd Conference on Decision and Control*, pp 3498-3503, Lake Buena Vista, FL, December 1994.

[Bur90] J.R. Burch, E.M. Clarke, K.L. McMillan, D.L. Dill, and Hwang, Symbolic Model Checking: 10^{20} states and beyond, In *Proceedings of the Fifth Annual Symposium on Logic in Computer Science*, June 1990.

[Cla81] E.M. Clarke, E.A. Emerson, characterizing properties of parallel programs as fixed points, in seventh international collogquim on automata, languages and programming, volume 85, LNCS, 1981

[Clar86] E.M. Clarke, E.A. Emerson, and A.P. Sistla, Automatic verification of finite-state concurrent systems using temporal-logic specifications, *ACM Transactions on Programming Languages and Systems*, 8(2):244-263, 1986

[Desh95] A. Deshpande and P. Varaiya, Viable Control of Hybrid Systems, *Hybrid Systems II*, A. Nerode (ed.), LNCS Volume 999, Springer-Verlag, 1995.

[Gode97] P. Godefroid, On the costs and benefits of using partial order methods for the verification of concurren systems, Partial Order Methods in Verification, DIMACS workshop, July 24-26, 1996, DIMACS series in discrete mathematics and theoretical computer science volume 29, D. Peled, V Pratt, and G.J. Hozmann (eds), American Mathematical Society 1997. pp 289-303,

[Hou96] L. Hou, A.N. Michel, and H. Ye, Stability Analysis of Switched Systems, In *Proceedings of the 35th Conference on Decision and Control*, pages 1208-1212, Kobe Japan, December 1996.

[Lem98] M.D. Lemmon, K. He, and C.J. Bett, Modeling Hybrid Control Systems using Programmable Timed Petri Nets, to appear in L'Automatisation des Processus Mixtes (ADPM'98), Rheims France, March 19-20, 1998.

[McM92] K. McMillan, Using Unfoldings to Avoid the State Explosion Problem in the Verification of Asynchronous Circuits. B.V. Bochmann and D.K. Probst, editors, *Computer Aided Verification, fourth international workshop, CAV'92*, pages 164-177, Springer-Verlag, 1992, Lecture Notes in Computer Science 663.

[McM93] K. McMillan, *Symbolic Model Checking*, Kluwer Academic Press, 1993.

[Pel91] P. Peleties and R. DeCarlo, Asymptotic Stability of m-switched systems using Lyapunov-like functins. In *Proceedings of the American Control Conference*, pp 1679-1684, Bost, MA, June 1991.

[Pet96] S. Pettersson and B. Lennartson, Stability and Robustness for Hybrid Systems, In *Proceedings of the 35th Conference on Decision and Control*, Kobe Japan, December 1996.

[Puri94] A. Puri and P. Varaiya, Decidability of hybrid systems with rectangular differential inclusions, In D.L. Dill editor, CAV'94: computer aided verification, Lecture Notes in Computer Science 818, pages 81-94, Springer-Verlag, 1994.

[Rant97] M. Johansson and A. Rantzer, Computation of Piecewise Quadratic Lyapunov Functions for Hybrid Systems, to appear in *IEEE Trans. of Automatic Control*, 1997

[TAH95] T.A. Henzinger, P-H Ho, and H Wong-Toi, A user guide to Hytech, Proceedings of the First workshop on Tools and Algorithms for the Construction and Analysis of Systems, TACAS:95, LNCS 1019, Springer-Verlag, pp. 41-71.

[TAH95a] T.A. Henzinger, P.W. Kopke, A. Puri, and P. Varaiya, What's decidable about hybrid automata?, In *Proceedings of the 27th Annual Symposium on Theory of Computing*, pages 373-382, ACM Press, 1994.