Modeling and Verification of Sampled-Data Hybrid Systems

B. Izaias Silva* and Bruce H. Krogh
Dept. of Electrical and Computer Engineering,
Carnegie Mellon University
(Izaias /krogh)@cmu.edu

Abstract

We introduce sampled data hybrid automata (SDHA) as a formal model of hybrid systems that result from clock-driven computer control of continuous dynamic sytems. In contrast to standard hybrid automata, the discrete state transitions in the SDHA can occur only at valid sampling times when the guard conditions are evaluated. Sequences of valid sampling times are defined by a clock structure that specifies bounds on the possible initial phases, period variations and jitter. Approximate quotient transition systems are then defined for SDHA as a theoretical framework for performing formal verification. The concepts introduced in this paper are illustrated by a numerical example. The paper concludes with a discussion of current research aimed at developing a tool for verifying prorperties of real sampled data control systems.

Keywords: Sampled-data hybrid systems, formal verification, finite-state approximations

1 INTRODUCTION

This paper concerns the modeling and verification of hybrid systems in which the conditions for discrete-state transitions are evaluated at discrete points in time determined by an external clock. This is the typical situation in computer-based control systems.

In standard hybrid automata models, events are generated when guards are enabled by the continuous state trajectories [1]. Clock-based sampling can be modeled using the hybrid automata formalism by introducing an integrator to generate the clock events. The problem with this approach is that there can be many clock events when nothing happens other than resetting the clock integrator. Consequently, verification procedures are bogged down computing and storing information that has no real significance.

To eliminate the unnecessary details of the clock behavior that would have to be introduced in hybrid automata models of sampled-data systems, we introduce sampled-data hybrid automata (SDHA). SDHA are standard hybrid automata extended with a clock structure that defines a set of parameters specifying the ranges for the initial clock phase, the clock period, and the possible variations in the actual sampling time with respect to the clock instants, called jitter. Jitter captures the variations that occur in sample-data systems due to variations in computation times or the scheduling of multiple tasks on the control computer.

In the hybrid systems literature, the continuous dynamics in sampled-data systems are often modeled by discrete-time state-transition equations (e.g., [2], [3], [4], [5]). In contrast, the SDHA retains continuous-time differential equations for the continuous dynamics since an analytical expression for the discrete-time state transition function can not be obtained for general nonlinear continuous dynamic systems.

We describe an approach to the verification of SDHA that avoids the clock event explosion inherent in standard hybrid automata models of samped-data systems. We achieve this by logging the necessary information at clock events only when actual discrete-state transitions occur. We propose an approach to extend existing techniques for verification of hybrid automata to deal efficiently with the sampled-data behavior.

The following section describes the clock structure used to model the possible sample-time sequences for general clock-driven sampled-data systems. We then define the SDHA in section 3 and develop the concept of transition systems and quotient transition systems for SDHA in section 4. Section 5 illustrates the concepts developed in this paper with a numerical example. Section 6 describes current research on enhancing the hybrid system verification tool CheckMate for verification of sampled-data control systems. Directions for further development of the theory of SDHA are also discussed.

2 CLOCK STRUCTURES FOR SAMPLED-DATA SYSTEMS

Our objective in this paper is to develop a formal model of clock-driven computer control systems of the type illustrated in Fig. 1. In this system, a continuous-time dynamic process is being controlled by a computer that samples process data through the interface labeled *sampling device*. Control inputs from the computer to the process are also delivered through

* Supported by grant from CNPq – proc N^{0} 20.0079/92

the sampling device. The computational routines are represented by the *controller* block. The data sampling and execution of the control program are driven by the *clock* that generates clock events at the *clock times* c_0, c_1, ... The *sampling times*, that is, the actual times at which data sampling occurs, are given by the sequence τ_o, τ_1, ... In general, the sampling times are different from the clock times because sampling is typically the result of commands in the control software. Moreover, the sampling process itself can take time.

We specify the parameters defining the sets of admissible clock times and sampling times for a given sampled-data system by a *clock structure*, defined as follows.

Definition 1. A clock structure $C=(\Theta,\Delta,B)$ is a triple where:

$\Theta = [\theta m,\theta M]$, range of values for the clock initial phase, with $0 \le \theta m \le \theta M$;

$\Delta = [\delta m, \delta M]$, range of values for the clock period, with $0 < \delta m \le \delta M$; and

$B = [\beta m,\beta M]$, range of values for the sampling jitter, with $0 \le \beta m \le \beta M$.

The first interval in the clock structure for the clock initial phase represents the fact that the initial clock time can be offset from the initial time assumed for the process dynamics by an amount $\theta \in \Theta$. Assuming the initial time for the process is $\tau = 0$,.this implies $c_0=\theta$. The second interval specifies the uncertainty in the clock period. We assume the clock period can vary at each clock event, which implies that

$$c_n - c_{n-1} \in \Delta, \; n = 0,1, \ldots . \qquad (1)$$

Note that (1) implies the uncertainty in the clock time due to the clock period variation accumulates from sampling time to sampling time. (An alternative assumption might be that the clock period is fixed, but can be within the range Δ, in which case there would be no accumulation of uncertainty in the clock time for a given sequence of clock times.) Combining the uncertainty in the clock initial phase and the clock period uncertainty (1) gives

$$c_n \in n\Delta + \Theta, \; n = 0,1, \ldots . \qquad (2)$$

The final interval in the clock structure represents the possible range of differences between the clock times and the sampling times, which we refer to as sampling jitter. The jitter interval introduces the constraint

$$\tau_n \in c_n + B, \; n = 0,1, \ldots, \qquad (3)$$

where + is the standard addition operation for interval arithmetic. Combining the jitter constraint with the constraint (2) on c_n gives

$$\tau_n \in n\Delta + \Theta + B, \; n = 0,1, \ldots,. \qquad (4)$$

Note that in contrast to the clock period uncertainty that propagates throughout the sequence of sampling times, the jitter uncertainty does not accumulate. The constraint on the inter-sample period is given by

$$\delta_m - (\beta_M - \beta_m) \le \tau_{n+1} - \tau_n \le \delta_M + (\beta_M - \beta_m). \qquad (5)$$

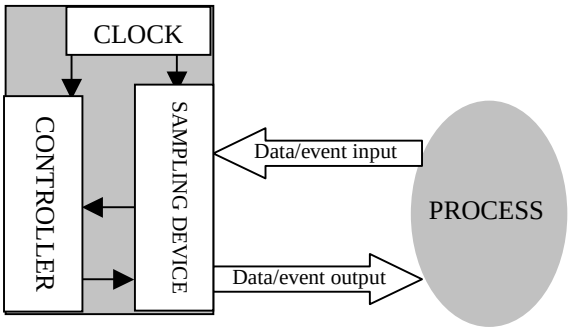

Figure 1. Sampled-data hybrid system.

Based on the above definitions and observations, given a clock structure $C = (\Theta, \Delta, B)$, we call a given sequence $\sigma = \{\tau_n\}$a *valid sampling sequence* if $\tau_n \in n\Delta + \Theta + B$, n = 0, 1, ..., and we let Σ_c denote the set of valid sampling sequences for C. We also use the following notation for sequences: given sequences σ, σ', σ-σ' denotes the sequence constructed by removing from σ all elements of the sequence σ', and $\sigma' \preceq \sigma$ means σ' is a *subsequence* of σ.

3 SAMPLE DATA HYBRID AUTOMATA (SDHA)

As a formal model of sampled-data systems, we modify the definition of the standard hybrid automaton as follows.

Definition 2. A sampled-data hybrid automaton is a pair H=(A,C), where C=(Θ,Δ,B) is a clock structure and A=(X, U, X_0, U_0, F, E, G) is a hybrid automaton where:

$X \times U$ is the set of *states* where $X = R^n$ is the *continuous state space* and U is a finite set of *discrete states*;

$X_0 \times U_0 \subseteq X \times U$ is the set of *initial states*;

$F = \{f_u : X \to X\}_{u \in U}$ are the *flows*, where $\dot{x} = f_u(x)$ determines the continuous state trajectories for discrete state u.

$E \subseteq U \times U$ is the set of *discrete transitions*;

$G: E \to 2^X$ defines the *guards* for the discrete transitions;

For simplicity, we have defined the SDHA with deterministic flows (differential equation rather than

differential inclusion) and we assume there is no reset of the continuous-state when discrete state transitions occur. We also assume discrete state transitions occur immediately when a guard condition is satisfied. Without loss of generality, we assume that there are no self loops in E, i.e., for any $u \in U$, $(u,u) \notin E$.

Definition 3. Given an SDHA H=(A,C), a function $s:R^+ \rightarrow X \times U$, denoted $s(\tau)=(s_C(\tau),s_D(\tau))$, is a *run* of H_{SD} if and only if $s(\tau)$ satisfies:

1. $s_C(0) \in X_0$, $s_D(0) \in U_0$
2. Let $\pi = \{\tau_i\}$ be the sequence of discontinuity points of $s_D(\tau)$. Then $\exists\, \sigma \in \Sigma_C$ such that:
   a) $\pi \preceq \sigma$;
   b) $\forall\, \tau_I \in \pi$, $\exists$ e= $(s_D(\tau_i^-),s_D(\tau_i^+)) \in E$ such that $s_C(\tau_i^-) \in G(e)$;
   c) $\forall\, \tau \notin \pi$, $\dot{s}_C(\tau) = F_{S_D(\tau)}(s_C(\tau))$;
   d) $\forall\, \tau' \in \sigma\text{-}\pi$ and $\forall\, u \in U$ such that e = $(s_D(\tau'),u) \in E$, $s_C(\tau') \notin G(e)$.

In words, 2.a) states that discrete state transitions only can occur at valid sampling times; 2.b) says that the guards are "forcing"; 2.c) is the standard rule that the continuous trajectory satisfies the flow constraints while the discrete state is constant. 2.d) states that if a discrete state transition does not occur at a sampling time in σ, no guard can be enabled at this time instant, reflecting the assumption that the guards force discrete-state transitions.

In the following, we denote the admissible continuous trajectory starting at x_0 under the differential equation f_u, for some $u \in U$, as $\zeta_{(u,x_0)}(.)$, where $\dot{\zeta}_{(u,x_0)}(\tau) = f_u(\zeta_{(u,x_0)}(\tau)), \forall \tau \geq 0$ and $\zeta_{(u,x_0)}(0) = x_0$. We also denote the set of reachable state from a set $X' \subseteq X$ at time interval $T=[\tau_1,\tau_2]$, under f_u by:

$$R^T(u,X) = \bigcup_{\tau \in T} \bigcup_{x_0 \in X} \{x \in X \mid \zeta_{(u,x_0)}(\tau) = x\}$$

We also define for each $u \in U$ the following set:

$$X_u = \{x \in \mathbf{X} \mid x \in \bigcap_{u \in \mathbf{U}} \overline{G((u,u'))}\}$$

4 VERIFICATION OF SDHA

In this section we show how transition systems and quotient transition systems can be defined for SDHA so that the concept of transition system simulation can be applied to perform formal verification. For standard hybrid automata, the *discrete-trace transition system* (DTS) can be defined to abstract away the continuous-time behaviors of the system [6]. The state space of the DTS is the set of continuous *entry states* for each location (discrete state). When the guard conditions force discrete state transitions, as we have assumed, the entry states are simply the boundaries of the guard conditions.

Similarly, for an SDHA a transition system can be defined, called a *sampled-trace transition system* (STTS), which records the possible states of the SDHA at times with discrete-state transitions occur. The principle difference between the DTS and the STTS is that the entry states for the discrete states in the SDHA are the states that are reached at the sampling times, which are not necessarily on the boundaries of the guard conditions. Therefore, the entry states for discrete state transitions in SDHA models cannot be restricted to these boundaries as they are in the DTS.

The STTS is defined as follows.

Definition 4. (STTS) Given a SDHA H=(A,C), its sampled-trace transition system is given by $T_H = (Q_H, \rightarrow_H, Q_0)$ with $Q_0 = X_0 \times U_0 \times \Theta$, $Q_H = (X \times U) \cup Q_0$ and the transition relation $\rightarrow_H$ is defined by following conditions:

1. Initial transitions. $(x_0,u_0,\theta_0) \rightarrow_H (x,u)$ Iff**:**
   a) $(x_0,u_0,\theta_0) \in X_0 \times U_0 \times [\theta_m,\theta_M]$;
   b) $x \in R^{B+\theta_0}(x_0,u_0)$; and
   c) either $u = u_0$ and $x \notin G(u_0,u)\, \forall u \neq u_0$, or $x \in G(u_0,u)$.
2. Discrete transitions. $(x,u) \rightarrow_H (x',u')$ iff**:**
   a) $e = (u,u') \in E$, $u' \neq u$;
   b) $\exists\, (\tau_1,..., \tau_N)$ s.t. all next conditions hold:
      i) $\tau_n \in n\Delta+B$, $1 \leq n \leq N$.
      ii) $x' = \zeta_{(u,x)}(\tau_N)$
      iii) $x' \in G(e)$;
      iv) $\forall\ u'' \neq u$, $\zeta_{(u,x)}(\tau_n) \notin G((u,u''))$,

   $1 \leq n < N$.

The definition of the STTS captures the sampled-data behaviors for all possible sampling sequences in Σ_C. The initial transitions record the continuous state at the time of the first sample, $\tau_o = c_0 + \beta_o$, where $c_0 = \theta \in \Theta$, the initial clock phase, and $\beta_o \in B$, the initial variation in the sampling time due to jitter. The subsequent discrete transitions from each state lead to all states that could be reached for some valid sampling sequence starting at that state. A sequence of transitions in the STTS starting from some initial state corresponds to a valid run in for the hybrid system H for a particular valid clock sequence. It is also true that for any valid run for H, there will be a corresponding sequence of transitions in the STTS

that records the values of the states along the run at the discrete state transitions in the run.

Having constructed a transition system for the SDHA, formal verification of SDHA can be performed by creating a finite partition of the STTS state space and constructing a *quotient transition system* (QTS).

Definition *5*. Given a transition system T = (Q,→, Q_0), and a P of Q, the quotient transition system of T is defined as T/P = (P,→$_{\mathsf{P}}$, Q_0/P) where

1. For all *P,P'* ∈ P, *P*→$_{\mathsf{P}}$*P'* iff there exist q ∈ *P* and q' ∈ *P'* such that *q*→ *q'* .
2. Q_0/P = {*P*∈P | P ⊆ Q_0}.

The quotient transition system *T*/P *simulates* T and it can be used for verification of properties of T [7].

To create a QTS for an STTS, separate partitions of the continuous state space are required for each discrete state, u ∈ U. We denote the partition of X for discrete state u by P_u. Finite partitions are also required for the initial state set X_o for each u ∈ U_o, denoted by $\mathsf{P}_{o,u}$, as well as for the interval of possible clock initial phases, denoted by P_θ. Letting P denote the complete partition, the QTS T_H/P for an STTS T_H can be constructed, *provided the reachability mapping could be computed for the continuous flows of the SDHA*.

In general, it is not possible to represent, much less compute, the sets of reachable states required to construct the QTS for a transition system defined for a hybrid system. Nevertheless, if one can compute a conservative approximation to the reachable state mapping, an *approximate* QTS (AQTS) can be created that simulates the transition system and the AQTS can be used for effective verification in many cases [6]. If the specification is not satisfied for the AQTS, the approximation can be improved by refining the partition of the STTS state space.

Given an SDHA H=(A,C), the procedure in figure computes the transition relation →$_{\mathsf{P}}$ for an AQTS T_H/P for a given partition P = ({P_u},{$\mathsf{P}_{o,u}$},P_θ) of the state space for T_H. This procedure creates transitions between elements (*P*,u), (*P'*,u') of the state space of T_H when for some n:

C1. $R^{[n\Delta+B]}(u,P) \cap G((u,u')) \neq \phi$; and

C2. $R^{[n\Delta+B]}(u,P) \cap P' \neq \phi$.

To show this leads to an AQTS, suppose there are continuous states x ∈*P* and x' ∈*P'* such that $(x,u) \rightarrow_H (x',u')$ in the STTS T_H. From the definition of T_H, this means that there is a sampling sequence (τ_1,..., τ_N) such that $\tau_n \in n\Delta+B$, $1 \le n \le N$, and $x' = \zeta_{(u,x)}(\tau_N) \in G((u,u'))$. This implies $x' \in R^{[n\Delta+B]}(u,P)$ for n=N and hence condition C1 is true for n=N. Moreover, since x' ∈*P'*, condition C2 is also true. Therefore, whenever there is a discrete transition in T_H, the procedure above creates the appropriate transition to create an AQTS T_H/P. Simliar logic applies to the initial state transitions.

Regarding the stopping condition, this will be always satisfied if we assume that the trajectories eventually violate some guard.

```
Procedure for computing →P

BEGIN PROCEDURE

→P := ϕ            % initialize →P as an empty set
FOR each u ∈ U0, P ∈ Po, Pθ ∈ Pθ
   FOR each u' ∈ U
   IF( R^{P_θ+B}(u,P) ∩ G(u,u') ≠ ϕ )&      ((u,u')∈E))
_     OR  (((R^{P_θ+B}(u,P) ∩ X_u) = ϕ)   &  (u'=u) )
         FOR  each P' ∈ P_u'
          IF  R^{P_θ+B}(u,P) ∩ P' ≠ ϕ
                Add ((P,u,Pθ),(P',u')) to →P
          END
         END
       END
   END
END

FOR each u ∈ U
   FOR each P ∈ P_u ∪ P_o,u
      n=1
      REPEAT
        FOR each u' ∈ U such that (u,u') ∈ E
           IF (R^[nΔ+B](u,P) ∩ G(u,u')) ≠ ϕ
                FOR each P' ∈ P_u'
                   IF (R^[nΔ+B](u,P) ∩ P') ≠ ϕ
                        Add ((P,u),(P',u')) to →P
                   END
                END
           END
        END
      n= n+1
      UNTIL (R^[nΔ+B](u,P) ∩ X_u ) = ϕ
      % stops when all states satisfy some guard;
   END
END

END PROCEDURE
```

Figure 2. Procedure for computing →$_{\mathsf{P}}$.

5 EXAMPLE

Consider an two-mode control system modeled by a hybrid automaton A as follows:

X = ([0,3],[0,5]), U = {on,off}

X_0={([1, 1.5],0}, U_0= on

$E = \{e_1=(on,off);\ e_2=(off,on)\}$

$$f_{on} = \begin{bmatrix} -1 & 1 \\ 1 & -3 \end{bmatrix} x + \begin{bmatrix} 0 \\ 10 \end{bmatrix};\ f_{off} = \begin{bmatrix} -1 & 1 \\ 1 & -3 \end{bmatrix} x + \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

$G(on) = \{(x_1,x_2) | x_2 \geq 2.5\},\ G(off) = \{(x_1,x_2) \mid x_2 \leq 2\}$

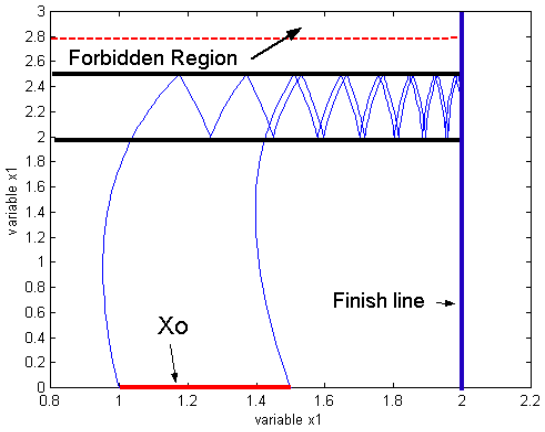

Figure 3. Simulation for hybrid automaton A.

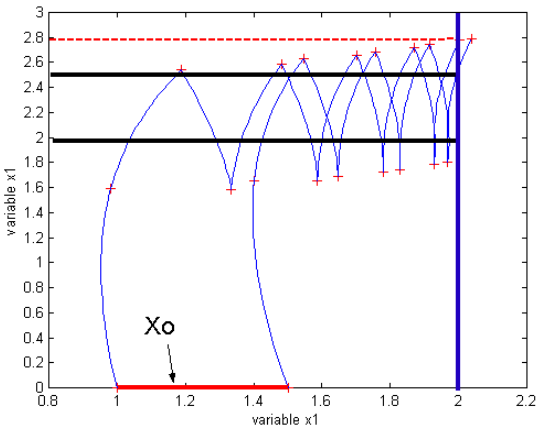

Figure 4. Simulation for SDHA H (δ=0.2s, θ=0s).

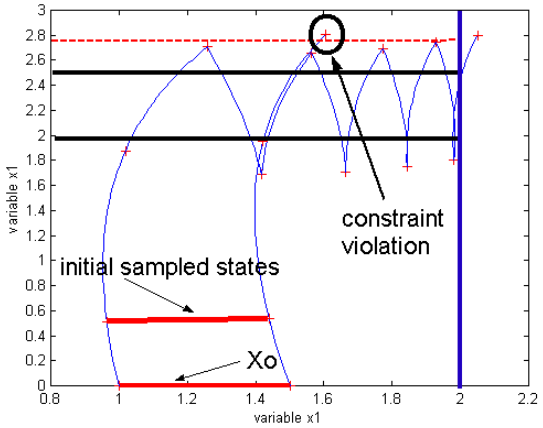

Figure 5 – System behavior for δ=0.2 s , θ=0.05s

For this system we would like to guarantee that the continuous state variable x_2 never goes beyond 2.8 and that eventually, variable $x_1 = 2$. Since the system is linear and 2-dimensional, we can characterize the flow of all possible behaviors just performing the simulation of the extreme point of the initial set. The trajectories in figure 3 shows that the finish line region is reached before going into the forbidden region.

Now, suppose we define an SDHA H = (A,C) where the clock structure will be C = ([0,0.05],[0.2,0.2],[0,0]). Thus, the sampling period is $\delta_m=\delta_M=0.2$, there is initial displacement ($\theta_m=0,\theta_M=0.05$) and there is no jitter ($\beta_m=\beta_M=0$). For a simulation with $\theta=0$, the system still satisfies the requirement, as shown in figure 4, even though the behavior is getting close to the forbidden region.Let us now simulate the system for $\theta = 0.05$. Figure 5 shows that in this case the system fails to fulfill the requirements. Some trajectories cross the alarm line before crossing the finish line. It illustrates the situation where a hybrid system that satisfies the specification for a continuous behavior could fail in a verification if taking into account the sampled-data behavior.

6 DISCUSSION

This paper introduces sampled-data hybrid automata (SDHA) as a formal model for hybrid systems in which the guard conditions for the discrete-state transitions are evaluated at clock-generated sampling times. The clock structure includes parameters to represent variations in the clock initial phase and the clock period. Variations in the difference between the clock events and the actual sampling times are also represented by a jitter parameter.

Formal verification of SDHA is possible through the construction of approximate quotient transition systems (AQTS) for the sampled-trace transition system (STTS) that represents the behavior of the SDHA at the instants the discrete-state transitions occur for all possible valid clock sequences.

Currenly, we are focusing on the numerical implementation of the proposed approach for formal verification of sampled-data hybrid systems by extending the features of the hybrid system verification tool *CheckMate* [8,9]. Hybrid dynamic systems are defined in *CheckMate* using the MATLAB Simulink graphical user interface. For example, the *CheckMate* model for the example in the previous section is shown in Figure 5.

Routines and data structures for computing sets of reachable states as required for the AQTS construction procedure are already available in *CheckMate*. The principle tasks are to incorporate the clock structure parameters in the *CheckMate* specification and automate the procedure for constructing the AQTS for the SDHA. Experiments can then be conducted to compare the SDHA approach with clock-based

approaches to modeling and verifying properties of sampled-date control systems.

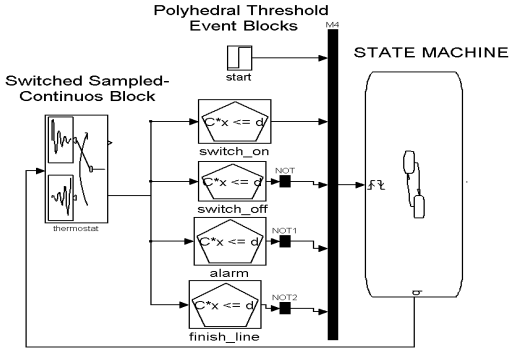

Figure 6. Two-modes SDHS model.

Routines and data structures for computing sets of reachable states as required for the AQTS construction procedure are already available in *CheckMate.* The principle tasks are to incorporate the clock structure parameters in the *CheckMate* specification and automate the procedure for constructing the AQTS for the SDHA. Experiments can then be conducted to compare the SDHA approach with clock-based approaches to modeling and verifying properties of sampled-date control systems.

Directions for further research on the modeling and verification of sampled-data hybrid systems include:

- improving the reachability approximation in the AQTS procedure by eliminating states that cannot be reached because of guard conditions that were satisfied at earlier sampling times;
- Including special discrete states that once the trajectories enter in the location defined by them, there is no further discrete transition. These states could be used for stopping conditions during the verification.
- methods for evaluating when effects of sampling are critical for verification and when the sampling rate is fast enough that the traditional models can be used;
- modeling clock-driven discrete-time computations (e.g., digital filters);
- integrating clock-triggered events with events generated by continuous state trajectories (the types of events captured by traditional hybrid automata).

All of these issues are critical for developing tools that address problems arising in sampled-data control applications.

REFERENCES

[1] T. Henzinger, The theory of hybrid automata, in Proc. 11th Annual IEEE Symposium on Logic in Computer Science (LICS 1996), pp. 278-292, 1996.

[2] Bemporad and M. Morari, Verification of hybrid systems via mathematical programming, in Hybrid Systems: Computation and Control, Lecture Notes of Computer Science, vol. 1569, pp. 30- 45, Springer-Verlag, 1999.

[3] Y. Huiqum, P. Pandya, and S. Yongqiang, A calculus for hybrid sampled data systems, Formal Techniques in Real-Time and Fault-Tolerant Systems. Third International Symposium Proceedings, ProCoS, pp. 716-37, Springer-Verlag, 1994.

[4] J. Raisch and S.D. O'Young, Discrete approximation and supervisory control of continuous systems, IEEE Trans. on Automatic Control, vol. 43, no. 4, April 1998.

[5] J. Lunze, Qualitative modelling of linear dynamical systems with quantized state measurements, Automatica, vol. 30, no. 3, pp. 417-432, March 1994.

[6] Chutinan and B.H. Krogh, Approximate quotient transition systems for hybrid systems, in Proc. 2000 American Control Conference, Chicago, June 2000.

[7] T. A. Henzinger, Hybrid automata with finite bisimulations, in ICALP 95: Automata, Languages, and Programming, pp. 324-335, Springer-Verlag, 1995.

[8] Chutinan, B.H. Krogh, B. Silva, K. Richeson, , Modeling and verifying hybrid dynamic systems using CheckMate, ADPM2000, Dortmund – Germany, Sept 2000.

[9] I. Silva and B. H. Krogh, Formal verification of hybrid systems using CheckMate: a case study, in Proc. 2000 American Control Conference, Chicago, June 2000,