December 2003

From Discrete Specifications to Hybrid Control

Paulo Tabuada
University of Notre Dame

George J. Pappas
University of Pennsylvania, pappasg@seas.upenn.edu

From Discrete Specifications to Hybrid Control

Abstract

A great challenge for modern systems theory is the design of controllers for continuous systems but with logical specifications. In this paper, we are interested in developing algorithmic methods which given a discrete time controllable linear system and a discrete specification (in the form of a finite transition system or a temporal logic formula), automatically design controllers resulting in desired, closed-loop behavior. This can be achieved using a natural approach involving three steps. In the first step, given a controllable linear system and discrete specification, we extract a finite transition system model which is equivalent (bisimilar) to the continuous system. The second step solves the controller synthesis problem for finite transition systems using well known and well developed algorithms. The third step, which is the focus of this paper, refines the discrete controller of the finite transition system, to a (necessarily) hybrid controller for the original continuous system. The hybrid controller composed with the continuous plant results in a closed-loop hybrid system that, by construction, satisfies the desired, discrete specification.

Proceedings of the 42nd IEEE
Conference on Decision and Control
Maui, Hawaii USA, December 2003

From Discrete Specifications to Hybrid Control[1]

Paulo Tabuada
Dept. of Electrical Engineering
University of Notre Dame
Notre Dame, IN 46556
e-mail: ptabuada@nd.edu

George J. Pappas
Dept. of Electrical and Systems Engineering
University of Pennsylvania
Philadelphia, PA 19104
e-mail: pappasg@seas.upenn.edu

Abstract

A great challenge for modern systems theory is the design of controllers for continuous systems but with logical specifications. In this paper, we are interested in developing algorithmic methods which given a discrete-time controllable linear system and a discrete specification (in the form of a finite transition system or a temporal logic formula), automatically design controllers resulting in desired, closed-loop behavior. This can be achieved using a natural approach involving three steps. In the first step, given a controllable linear system and discrete specification, we extract a finite transition system model which is equivalent (bisimilar) to the continuous system. The second step solves the controller synthesis problem for finite transition systems using well known and well developed algorithms. The third step, which is the focus of this paper, refines the discrete controller of the finite transition system, to a (necessarily) hybrid controller for the original continuous system. The hybrid controller composed with the continuous plant results in a closed-loop hybrid system that, by construction, satisfies the desired, discrete specification.

1 Introduction

The invasion of computation and networking inside physical devices has resulted in great challenges for modern and future systems and control theory. Improved understanding and reliable design tools for software controlled systems remain elusive. The greatest technical challenge for our community is understanding the relationship, and mapping properties between the continuous world of control systems, and the discrete world of (programming) languages, automata, and logic.

The above problems very frequently arise when one would like to design a controller for a continuous system but with discrete or logical specifications. Consider, for example, the controllable discrete-time system

$$\Sigma: \qquad x(t+1) = Ax(t) + Bu(t)$$

where the desired specification is neither traditional controllability nor stabilizability objectives, but rather a (linear) temporal logic formula ϕ, such as

$$\phi : \Box\ (o_1 \implies \Diamond_5\, o_2 \vee (\Diamond\, o_3))$$

where o_1, o_2, o_3 are symbols representing regions of the state space of Σ (for example o_1 could denote the set $(x_1 < -5 \wedge x_2 < -3)$), $\Box$ means always, $\Diamond$ means eventually, and $\Diamond_5$ means within five time steps. The desired specification for our example is : it is always the case that if the system visits region o_1 then either the system goes to o_2 within five steps, or, otherwise, must eventually go to region o_3.

Note that the specification captures both desired continuous behavior but also desired discrete logic. Therefore, controller design for this problem includes designing the software logic in addition to designing the continuous control. Furthermore, note that any controller for the above problem must have at least one bit of memory in order to know whether o_1 has been visited or not. Our goal is to develop algorithmic methods that design controllers for linear systems with respect to temporal logic specifications.

In the computer science community it is well known how to algorithmically translate temporal logic formulas to finite transition systems [16]. We therefore consider the equivalent problem of designing controllers for control system Σ for specifications modeled as finite transition systems. Our approach involves three steps. In the first step (which is the focus of [15]), given controllable system Σ and an observation map, sending continuous states into a finite set of symbols $O = \{o_1, \ldots o_p\}$, we construct a finite transition system that is bisimilar to the continuous system. Therefore both the controllable system and the discrete transition system can generate exactly the same sequences of symbols. In the second step, we can use existing methods

[1]This research is partially supported by NSF Information Technology Research Grant CCR01-21431 and NSF CAREER CCR-01-32716.

0-7803-7924-1/03/$17.00 ©2003 IEEE

and algorithms ([8, 10, 5]) for temporal logic synthesis of finite transition systems. The third step of the approach, which is the focus of this paper, is concerned with mapping the controller designed for the discrete transition system, to a controller for the original continuous system. If the specification is not memoryless, then the controller is necessarily a hybrid system specifying continuous (control) as well as discrete (software, switching logic) information. Furthermore, we show that the hybrid controller composed with the original system indeed satisfy the the desired discrete specification, which is our overall goal.

Related literature: Controller synthesis using logic is described in [12] however, logic is not used as a specification mechanism but rather to motivate the development of the synthesis procedures as well as to prove several facts regarding the proposed algorithms. Other synthesis techniques for continuous or hybrid systems with discrete specifications include supervisory control based on approximate finite abstractions [3], invariants for the continuous dynamics [14], convexity properties of affine systems [6], game theoretic approaches [7], and mixed integer linear programming [1]. Language based descriptions of motion have also been considered resulting in motion description languages [2, 9, 4].

2 Transition Systems

Transition systems, which we now define, will be the main modeling tool in this paper.

Definition 2.1 *A transition system with observations is a tuple* $T = (Q, Q^0, \longrightarrow, O, H)$, *where:*

- Q *is a (possibly infinite) set of states,*
- $Q^0 \subseteq Q$ *is a set of initial states,*
- $\longrightarrow \subseteq Q \times Q$ *is a transition relation,*
- O *is a (possibly infinite) set of observations,*
- $H : Q \to O$ *is a map assigning to each* $q \in Q$ *an observation* $H(q) \in O$.

We say that T is finite when both Q and O are finite, and infinite otherwise. We will usually denote a pair $(q, q') \in \longrightarrow$ by $q \longrightarrow q'$. The Post operator returns all the states that are one step reachable from a given state, formally we have:

$$\text{Post(q)} = \{\text{q}' \in \text{Q} \ : \ \text{q} \longrightarrow \text{q}'\}$$

Linear systems can be seen as generating infinite transition systems. Given the discrete-time linear system

$$\Sigma : \qquad x(t+1) = Ax(t) + Bu(t)$$

we can define transition system

$$T_\Sigma = (\mathbb{R}^n, \mathbb{R}^n, \longrightarrow_\Sigma, O, H_\Sigma) \tag{2.1}$$

where $Q = Q^0 = \mathbb{R}^n$, the state space, and the transition relation is defined as $x \longrightarrow_\Sigma x'$ iff there exists input $u \in \mathbb{R}^m$ such that $x' = Ax + Bu$. The transition system therefore captures the state dynamics of Σ, without maintaining the input which produced them. Therefore, T_Σ is a slightly more (control) abstract model than Σ. In order to complete the definition of transition system we must also specify the observation map H_Σ and O. The correct choice of O and H_Σ is one of the factors enabling the refinement of discrete to hybrid controllers.

Transition systems, with possibly different number of states, can be related by so-called simulation and bisimulation relations. Given a relation $R \subseteq Q_1 \times Q_2$ we denote by $R(Q_1)$ the image of Q_1, that is

$$R(Q_1) = \{q_2 \in Q_2 \mid \exists q_1 \in Q_1 \text{ with } (q_1, q_2) \in R\}$$

and by R^{-1} we denote the inverse relation defined by:

$$R^{-1} = \{(q_2, q_1) \in Q_2 \times Q_1 \ : \ (q_1, q_2) \in R\}$$

Definition 2.2 *Let* $T_1 = (Q_1, Q_1^0, \longrightarrow_1, O, H_1)$ *and* $T_2 = (Q_2, Q_2^0, \longrightarrow_2, O, H_2)$ *be transition systems and let* $R \subseteq Q_1 \times Q_2$ *be a relation. Relation* R *is called a simulation relation from* T_1 *to* T_2 *if* $R(Q_1^0) \subseteq Q_2^0$, *and* $(q_1, q_2) \in R$ *implies:*

- *if* $q_1 \longrightarrow_1 q_1'$, *then there exists* $q_2' \in Q_2$ *such that* $q_2 \longrightarrow_2 q_2'$ *and* $(q_1', q_2') \in R$,
- $H(q_1) = H(q_2)$.

Relation R *is a bisimulation relation between* T_1 *and* T_2 *if* R *is a simulation relation from* T_1 *to* T_2 *and* R^{-1} *is a simulation relation from* T_2 *to* T_1.

Note that, in Definition 2.2, we require the observation spaces of T_1 and T_2 to be the same. If T_1 is a transition system with state set Q_1, then transition system T_2 with state set $Q_2 \subseteq Q_1$ is called a subtransition system (or subsystem) of T_1 if T_1 simulates T_2 with respect to the inclusion map $i : Q_2 \longrightarrow Q_1$, that is, the relation $R = \{(q_2, q_1) \in Q_2 \times Q_1 \mid q_1 = i(q_2)\}$ is a simulation relation.

We now define a composition operator for the class of transition systems that we consider in this paper. In particular, we consider a composition operator that synchronizes the transition systems based on their respective observations.

Definition 2.3 *Let* $T_1 = (Q_1, Q_1^0, \longrightarrow_1, O, H_1)$ *and* $T_2 = (Q_2, Q_2^0, \longrightarrow_2, O, H_2)$ *be two transition systems with the same observation set* O. *The parallel composition of* T_1 *and* T_2 *(with output synchronization) is denoted by*

$$T_1 \parallel_O T_2 = (Q, Q^0, \longrightarrow, O, H)$$

where

- $Q = \{(q_1, q_2) \in Q_1 \times Q_2 \ : \ H_1(q_1) = H_2(q_2)\}$;
- $Q^0 = \{(q_1, q_2) \in Q_1^0 \times Q_2^0 \ : \ H_1(q_1) = H_2(q_2)\}$;
- $(q_1, q_2) \longrightarrow (q_1', q_2')$ *for* $(q_1, q_2), (q_1', q_2') \in Q$ *iff* $q_1 \longrightarrow_1 q_1'$ *and* $q_2 \longrightarrow_2 q_2'$;
- $H(q_1, q_2) = H_1(q_1) = H_2(q_2)$.

Our controller synthesis problem is the following : Given continuous plant Σ, its corresponding infinite transition system T_Σ, and discrete specification T_S, design controller T_C such that $T_C \parallel_O T_\Sigma$ is simulated by the specification T_S. Therefore the closed loop behavior is captured by the desirable behavior. This is performed in three steps. In the first step, described in Section 3, given a continuous linear system Σ we show how to extract a finite transition system T_Δ that is bisimilar to T_Σ. The second step, described in Section 4, we show that controllers for T_Δ exist if and only if controllers for T_Σ exist. Finally, in Section 5, we show how to construct the closed loop system for T_Σ, given designed discrete controllers for T_Δ.

3 From the continuous to the discrete

In this section, we summarize the results obtained in [15], which are utilized in this paper. Consider a discrete time controllable linear system:

$$\Sigma : \quad x(t+1) = Ax(t) + Bu(t)$$

Controllability guarantees the existence of a feedback transformation:

$$\begin{bmatrix} y \\ v \end{bmatrix} = U \begin{bmatrix} x \\ u \end{bmatrix} = \begin{bmatrix} F & \mathbf{0}_{n \times m} \\ G & H \end{bmatrix} \begin{bmatrix} x \\ u \end{bmatrix} \tag{3.1}$$

transforming system Σ into Brunovsky normal form [13]. This transformation incorporates important system information that will be used in this section as well as in Section 5. Associated with Σ is the infinite transition system T_Σ described in (2.1):

$$T_\Gamma = (\mathbb{R}^n, \mathbb{R}^n, \longrightarrow_\Sigma, O, H_\Sigma)$$

To obtain a finite bisimulation of T_Σ we consider a finite set of observations O. Observations will correspond to subsets of $\mathbb{R}^n$ defined by boolean combinations of predicates of the form:

$$fx + c \sim 0 \tag{3.2}$$

where f is a row of matrix F, $c \in \mathbb{Q}$ and $\sim \in \{<, \leq, =, \geq, >\}$. Given p such predicates we define the observation space to be $\{0,1\}^p$ and the observation map as:

$$H_\Sigma(x) = \begin{bmatrix} H_{\Sigma 1}(x) \\ H_{\Sigma 2}(x) \\ \vdots \\ H_{\Sigma p}(x) \end{bmatrix} \qquad H_{\Sigma i} = \begin{cases} 1 & \text{if} \quad f_i x + c_i \sim_i 0 \\ 0 & \text{if} \quad \text{otherwise} \end{cases} \tag{3.3}$$

The vector $H_\Sigma(x)$ will then have a 1 at position i when state x satisfies the predicate $f_i x + c_i \sim_i 0$ and a 0 otherwise. The main result of [15] can now be stated as follows:

Theorem 3.1 ([15]) *Let* Σ *be a discrete time controllable linear system and* T_Σ *its associated infinite transition system with observation space* $O = \{0,1\}^p$ *and observation map as defined in (3.3). Then, there exists an effectively computable finite transition system* T_Δ, *bisimilar to* T_Σ.

The bisimulation relation between T_Σ and T_Δ is in fact defined by a map $\pi : \mathbb{R}^n \to Q_\Delta$, that is $R = \{(x, q) \in \mathbb{R}^n \times Q_\Delta \ : \ \pi(x) = q\}$. More details regarding the construction of relation R and transition system T_Δ can be found in [15].

In this paper, we are interested in designing controllers for T_Σ, where the desired specification is modeled by a finite transition system with observation space $O = \{0,1\}^p$. Such transition systems can be translations of temporal logic formulas, such as LTL (see [16]) formulas, or they can be high level specifications for the desired closed loop behavior expressed directly in transition system form. We denote such a specification transition system by T_S and we will assume that the observation space of T_S is the observation space of the plant.

4 Discrete Controllers

A controller forcing our discrete model T_Δ to satisfy the specification given by T_S can now be defined.

Definition 4.1 (Discrete Controller) *Let* T_Δ *be the transition system described in Theorem 3.1, and let* T_S *be a transition system with the same observation space, modeling the desired specification.*

A controller for T_Δ, denoted by T_C, is a subtransition system of $T_S \|_O T_\Delta$, that is, $T_S \|_O T_\Delta$ is a simulation of T_C with respect to the inclusion map.

We now show in what sense T_C can be seen as a controller.

Proposition 4.2 *Transition system T_S is a simulation of transition system $T_C \|_O T_\Delta$.*

The existence of a simulation from $T_C \|_O T_\Delta$ to T_S implies that the observed behavior of $T_C \|_O T_\Delta$ is included in the observed behavior of the specification. We also note that we can choose our controller to be $T_C = T_S \|_O T_\Delta$, however $T_S \|_O T_\Delta$ may fail to satisfy certain important properties usually required by a controller, such as nonblocking for example. Such a drawback can be incorporated in the control design by selecting a subtransition system of $T_S \|_O T_\Delta$ with the desired (say nonblocking) properties. Proposition 4.2 is a consequence of the following two lemmas:

Lemma 4.3 *Transition system T_C is bisimilar to transition system $T_C \|_O T_\Delta$.*

Proof: Consider the relation $R \subseteq (Q_S \times Q_\Delta) \times ((Q_S \times Q_\Delta) \times Q)$ defined by $((q_S, q_\Delta,), ((q'_S, q'_\Delta), q''_\Delta) \in R$ iff $q_\Delta = q'_\Delta = q''_\Delta$ and $q_S = q'_S$. We first show that $T_C \|_O T_\Delta$ simulates T_C. Assume that $(q_S, q_\Delta) \longrightarrow_C (q'_S, q'_\Delta)$ and note that this implies $q_\Delta \longrightarrow_\Delta q'_\Delta$. Consider now any state R-related to (q_S, q_Δ). By definition of R, such state is of the form $((q_S, q_\Delta), q_\Delta)$ and by definition of parallel composition, we have that $(q_S, q_\Delta) \in Q_C \Rightarrow H_C(q_S, q_\Delta) = H_\Delta(q_\Delta) = H_{\|_O}((q_S, q_\Delta), q_\Delta)$. Similarly $H_C(q'_S, q'_\Delta) = H_\Delta(q'_\Delta) = H_{\|_O}((q'_S, q'_\Delta), q'_\Delta)$ holds. These equalities between observation maps combined with $(q_S, q_\Delta) \longrightarrow_C (q'_S, q'_\Delta)$ and $q_\Delta \longrightarrow_\Delta q'_\Delta$ now imply that $((q_S, q_\Delta), q_\Delta) \longrightarrow_{\|_O} ((q'_S, q'_\Delta), q'_\Delta)$ which shows that $T_C \|_O T_\Delta$ simulates T_C.

Conversely, lets assume that $((q_S, q_\Delta), q_\Delta) \longrightarrow_{\|_O} ((q'_S, q'_\Delta), q'_\Delta)$. Such transition implies that $(q_S, q_\Delta) \longrightarrow_C (q'_S, q'_\Delta)$ and since any state R-related to $((q_S, q_\Delta), q_\Delta)$ is of the form (q_S, q_Δ) we only need to show that $H_{\|_O}((q_S, q_\Delta), q_\Delta) = H_C(q_S, q_\Delta)$ and $H_{\|_O}((q'_S, q'_\Delta), q'_\Delta) = H_C(q'_S, q'_\Delta)$ to conclude that T_C simulates $T_C \|_O T_\Delta$. However this immediately follows from the definition of parallel composition with output synchronization. ∎

Lemma 4.4 *Transition system T_S simulates transition system T_C.*

Proof: The proof follows the same argument as the proof of Lemma 4.3 once one considers the relation $R \subseteq Q_C \times Q_S$ defined by $(q_C, q'_S) = ((q_S, q_\Delta), q'_S) \in R$ iff $q_S = q'_S$. ∎

We now show that a controller T_C for T_Δ exists if and only if a controller T'_C for T_Σ exists. This is a consequence of the existence of a bisimulation relation between T_Σ and T_Δ.

Theorem 4.5 *A controller T_C forcing system T_Δ to satisfy specification T_S exists iff there exists a controller T'_C forcing system T_Σ to satisfy specification T_S. Furthermore, we can take $T_C = T'_C$.*

This theorem is a simple consequence of the following well know property of bisimulations and Proposition 4.2.

Proposition 4.6 (Adapted from [11]) *Let T_1 and T_2 be transition systems with the same observation space. If T_1 is bisimilar to T_2 then, for any transition system T with the same observation space, $T \|_O T_1$ is bisimilar to $T \|_O T_2$.*

The proof of Theorem 4.5 is now a simple application of the previous proposition. Given transition systems T_1 and T_2, we denote by $T_1 \cong T_2$ the existence of bisimulation relation between T_1 and T_2. We now have $T_\Sigma \cong T_\Delta$ from which follows $T_C \|_O T_\Sigma \cong T_C \|_O T_\Delta$ by Proposition 4.6. Now since T_S simulates $T_C \|_O T_\Delta$ it also simulates $T_C \|_O T_\Sigma$ which shows that T_C is a controller for T_Σ.

Existence of controllers is therefore ensured, however T_C is an abstract (discrete) description of our controller. In the next section we refine our controller from the discrete system T_Δ to the continuous system T_Σ.

5 From the discrete to the continuous

Given any controller T_C, we now construct a (discrete-time) hybrid control system H based on Σ and T_C such that the transition system T_H associated with H is bisimilar to $T_C \|_O T_\Sigma$. We start by characterizing the set of inputs for the linear system Σ associated with a given transition in T_Δ. We denote by $[q]$ the set of all points $x \in \mathbb{R}^n$ such that $\pi(x) = q$ (the map π defines the bisimulation relation between T_Σ and T_Δ as discussed in Section 3). This set is defined by boolean combinations of predicates of the form $\phi_i = f_i x + c_i \sim_i 0$, $i \in I$. The predicates ϕ_i and the map Ξ defined by

$$\Xi(\phi_i) = \textbf{True}$$

when $i \notin \{k_1, k_1+k_2, \ldots, k_1+k_2+\ldots+k_r\}$ and

$$\Xi(\phi_i) = g_j x + h_j u + c_i \sim_i 0$$

when $i = k_1+k_2+\ldots+k_j$ and where g_j and h_j are the rows of matrices G and H defined in (3.1), respectively, will be instrumental in stating the next result:

Proposition 5.1 *Let T_Δ be the finite bisimilar quotient of transition system T_Σ associated with a discrete time controllable linear system $\Sigma = (A, B)$. If $q_\Delta \longrightarrow_\Delta q'_\Delta$ in T_Δ and $[q'_\Delta]$ is defined by:*

$$[q'_\Delta] = \left\{ x \in \mathbb{R}^n \ : \ \bigvee_{r \in R} \bigwedge_{s \in S_r} \phi_{rs}(x) \right\} \tag{5.1}$$

then, the inclusion $Ax + Bu \in [q'_\Delta]$ is satisfied for any $x \in [q_\Delta]$ iff $(x, u) \in \mathcal{A}(q_\Delta, q'_\Delta)$ with $\mathcal{A}$ defined by:

$$\mathcal{A}(q_\Delta, q'_\Delta) = \left\{ (x,u) \in [q_\Delta] \times \mathbb{R}^m : \bigvee_{r \in R} \bigwedge_{s \in S_r} \Xi(\phi_{rs})(x,u) \right\} \tag{5.2}$$

Proof: Assume, without loss of generality, that Σ has been transformed into Brunovsky normal form. From $q_\Delta \longrightarrow_\Delta q'_\Delta$ and bisimilarity between T_Δ and T_Σ follows that any $y \in [q_\Delta]$ satisfies:

$$y \longrightarrow_\Sigma y' \in [q'_\Delta] \tag{5.3}$$

Furthermore, from the Brunovsky form of Σ, (5.3) holds iff the inputs v satisfy:

$$v_j = y'_{k_1+k_2+\ldots+k_j} \tag{5.4}$$

for $j = 1, 2, \ldots, m$. Since $y' \in [q'_\Delta]$, y' satisfies the predicates in (5.1) and from (5.4) we conclude that v satisfies all the predicates ϕ_{rs} defining $[q'_\Delta]$ such that $\phi_{rs} = y'_{k_1+k_2+\ldots+k_j} + c \sim 0$. Noting that the transformed inputs v are obtained from the original states x and inputs u by $v = Gx + Hu$ we immediately see that:

$$\begin{aligned} y'_{k_1+k_2+\ldots+k_j} + c &= v_j + c \\ &= w_j v + c \\ &= w_j(Gx + Hu) + c \\ &= w_j G x + w_j H u + c \\ &= g_j x + h_j u + c \end{aligned}$$

where w_j is the row vector with a 1 on position j and zeros elsewhere. We thus see that for any $x \in [q_\Delta]$ we have that $Ax + Bu \in [q'_\Delta]$ iff $(x, u) \in \mathcal{A}(q_\Delta, q'_\Delta)$. ∎

Having identified the set of inputs associated with any transition in T_Δ, we can control Σ by restricting its inputs. Such restriction is captured in the following hybrid closed loop model:

Definition 5.2 *Given a controllable discrete-time linear system $\Sigma = (A, B)$ and a controller $T_C = (Q_C, Q_C^0, \longrightarrow_C, O, H_C)$, the implementation of $T_C \parallel_O T_\Sigma$ is given by the hybrid closed loop system H defined by:*

$$\begin{aligned} x(t+1) &= Ax(t) + Bu(t) \\ (x(t), u(t)) &\in \widehat{\mathcal{A}}(q_C(t)) \\ q_C(t+1) &\in H_C^{-1} \circ H_\Sigma(x(t+1)) \end{aligned}$$

where

$$\widehat{\mathcal{A}}(q_C(t)) = \bigcup_{q'_C \in \mathrm{Post}(q_C(t))} \mathcal{A}(\pi_\Delta(q_C(t)), \pi_\Delta(q'_C))$$

and $\pi_\Delta : Q_S \times Q_\Delta \to Q_\Delta$ is the natural projection from $Q_S \times Q_\Delta$ to Q_Δ.

Associated to hybrid system H is the transition system $T_H = (Q_H, Q_H^0, \longrightarrow_H, H_H, O)$ defined by:

- $Q_H = Q_C \times \mathbb{R}^n$;
- $Q_H^0 = \{(q, x) \in Q_C^0 \times \mathbb{R}^n \ : \ H_C(q) = H_\Sigma(x)\}$;
- $(q_C, x) \longrightarrow_H (q'_C, x')$ iff $x' = Ax + Bu$, $(x, u) \in \widehat{\mathcal{A}}(q_C)$ and $H_C(q') = H_\Sigma(x')$.
- $H_H(q_C, x) = H_C(q_C) = H_\Sigma(x)$.

Transition system T_H allows to show that the closed loop hybrid system H is in fact an implementation of the closed loop behavior described by $T_C \parallel_O T_\Sigma$.

Proposition 5.3 *Transition system T_H is bisimilar to $T_C \parallel_O T_\Sigma$.*

Proof: Consider the relation $R \subseteq Q_H \times (Q_S \times Q_\Delta)$ defined by $(q_H, (q_S, q_\Delta)) = ((q_\Delta^c, x), (q_S, q_\Delta)) = (((q'_S, q'_\Delta), x), (q_S, q_\Delta)) \in R$ iff $(q'_S, q'_\Delta) = (q_S, q_\Delta)$. The proof now follows the same argument as the proof of Lemma 4.3. ∎

Proposition 5.3 shows that H constitutes the desired closed loop system since $T_C \parallel_O T_\Delta$ being bisimilar to T_H and $T_C \parallel_O T_\Delta$ satisfying the desired discrete specification T_S implies that T_H also satisfies the specification. Furthermore, as every step in the construction of H is effectively computable we have the following result:

Theorem 5.4 *Let Σ be a discrete time controllable linear system, T_Σ its associated transition system with observation space $O = \{0, 1\}^p$ and observation map as*

defined in (3.3) and T_S a specification transition system. Then, it is decidable to determine if there is a controller for Σ enforcing the specification T_S. Furthermore, when such controller exists, it admits the hybrid closed loop implementation described by H which is effectively computable.

Proof: Deciding the existence of a controller for Σ amounts to determine if the observed behavior of $T_S \parallel_O T_\Delta$ is non-empty which is decidable. Furthermore, since H is obtained from T_C by enriching the states of T_C with the finite predicates defining $\widehat{\mathcal{A}}$, H is also effectively computable. ∎

The previous result summarizes the paper main contributions. Existence of controllers for discrete specifications can be decided. Furthermore, when a controller exists it admits a hybrid closed loop implementation that can be obtained in a totally automated fashion. Another important characteristic of the presented method is the automatic synthesis of both the switching logic (implemented by software) and the continuous aspects of control. This fact is especially important since verification of hybrid systems is currently limited to systems with very simple continuous dynamics such as timed automata. The proposed approach, thus overcomes the need for formal verification since the resulting system satisfies the specification by design.

6 Discussion

In this paper we have shown how to design controllers enforcing discrete specifications for discrete time controllable linear systems. The synthesis procedure relied on the computation of a finite bisimulation of the original plant as described in [15]. A finite controller is first computed for the finite model and subsequently refined to an hybrid closed loop. The proposed synthesis methodology thus generates the switching logic stemming from the discrete specification as well as the continuous inputs that are admissible to steer the system while satisfying the specification.

The presented results suggest a framework for the automatic synthesis of controllers for temporal logic specifications by converting logic formulas into discrete specifications in the form of transition systems. Furthermore, the algorithmic nature of the approach also suggests the complete automation of controller synthesis which is currently being investigated by the authors.

References

[1] A. Bemporad and M.Morari. Control of systems integrating logic, dynamics and constraints. *Automatica*, 35(3):407–427, 1999.

[2] R. W. Brockett. Hybrid models for motion control systems. In H. Trentelman and J. Willems, editors, *Perspectives in control*, pages 29–54. Birkhauser, Boston, 1993.

[3] J.E.R. Cury, B.H. Krogh, and T. Niinomi. Synthesis of supervisory controllers for hybrid systems based on approximating automata. *IEEE Transactions on Automatic Control : Special Issue on Hybrid Systems*, 43(4):564–568, April 1998.

[4] M. Egerstedt and R.W. Brockett. Feedback can reduce the specification complexity of motor programs. In *Proceedings of the 40th IEEE Conference on Decision and Control*, Orlando, FL, December 2001.

[5] E. A. Emerson and E. M. Clarke. Using branching time temporal logic to synthesize synchronization skeletons. *Science of Computer Programming*, 2:241–266, 1982.

[6] L.C.G.J.M. Habets and J. H. van Schuppen. Control of piecewise-linear hybrid systems on simplices and rectangles. In M. D. Di Benedetto and A. Sangiovanni-Vincentelli, editors, *Hybrid Systems: Computation and Control*, volume 2034 of *Lecture Notes in Computer Sience*, pages 261–274. Springer-Verlag, 2001.

[7] John Lygeros, Claire Tomlin, and Shankar Sastry. Controllers for reachability specifications for hybrid systems. *Automatica*, 35(3), March 1999.

[8] P. Madhusudan and P.S. Thiagarajan. Branching time controllers for discrete event systems. *Theoretical Computer Science*, 274:117–149, March 2002.

[9] V. Manikonda, P.S. Krishnaprasad, and J. Hendler. Languages, behaviors, hybrid architectures and motion control. In J. Willems and J. Baillieul, editors, *Mathematical control theory*, pages 199–226. Springer-Verlag, Boston, 1998.

[10] Z. Manna and P. Wolper. Synthesis of communication processes from temporal logic specifications. *ACM Transactions on Programming Languages and Systems*, 6:68–93, 1984.

[11] R. Milner. *Communication and Concurrency*. Prentice Hall, 1989.

[12] T. Moor and J. M. Davoren. Robust controller synthesis for hybrid systems using modal logic. In M. D. Di Benedetto and A. Sangiovanni-Vincentelli, editors, *Hybrid Systems: Computation and Control*, volume 2034 of *Lecture Notes in Computer Science*. Springer-Verlag, 2001.

[13] Eduardo D. Sontag. *Mathematical Control Theory*, volume 6 of *Texts in Applied Mathematics*. Springer-Verlag, New-York, 2nd edition, 1998.

[14] J.A. Stiver, X.D. Koutsoukos, and P.J. Antsaklis. An invariant based approach to the design of hybrid control systems. *International Journal of Robust and Nonlinear Control*, 11(5):453–478, 2001.

[15] Paulo Tabuada and George J. Pappas. Finite bisimulations of controllable linear systems. In *Proceedings of the 42th IEEE Conference on Decision and Control*. To appear.

[16] Pierre Wolper. Constructing automata from temporal logic formulas: A tutorial. In E. Brinksma, H. Hermanns, and J. P. Katoen, editors, *Lectures on Formal Methods and Performance Analysis*, volume 2090 of *Lecture Notes in Computer Science*. Springer-Verlag, 2000.