Proceedings of the
44th IEEE Conference on Decision and Control, and
the European Control Conference 2005
Seville, Spain, December 12-15, 2005

Symbolic sub-systems and symbolic control of linear systems

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

Abstract— **This paper presents an approach to the control of continuous systems through the use of symbolic models describing the system behavior *only at a finite number* of points in the state space. These symbolic models can be seen as abstract representations of the continuous dynamics enabling the use of algorithmic controller design methods. We identify a class of linear control systems for which the loss of information incurred by working with symbolic sub-systems can be compensated by feedback. We also show how to transform symbolic controllers designed for a symbolic sub-system into controllers for the original system. The resulting controllers combine symbolic controller dynamics with continuous feedback control laws and can thus be seen as hybrid systems. Furthermore, if the symbolic controller already accounts for software/hardware requirements, the hybrid controller is guaranteed to enforce the desired specifications by construction thereby reducing the need for formal verification.**

I. Introduction

In this paper we propose an approach to the control of linear systems based on *symbolic models*. In particular, we are interested in finite state models capturing the essential properties of linear control systems. The finite state nature of these models is important for two main reasons. Firstly, finite state models are especially well suited for automated analysis and design which is becoming increasingly important given the size of nowadays complex control systems. The use of such models thus opens new algorithmic perspectives for analysis and design. Secondly, finite state models offer a common language to describe an abstract view of continuous dynamics as well as the software implementation of control algorithms. It is therefore possible to formally reason about the behavior of the interconnection between continuous dynamics, control software and hardware, which has been one of the main thrusts behind the research area of *hybrid systems*. With the objective of strengthening this connection between continuous models of dynamics and finite state models of software we will focus, in this paper, on a particular symbolic model for control systems: *symbolic sub-systems*.

The success of a ”symbolic systems theory” based on symbolic models of continuous systems relies on a satisfactory answer to the following fundamental questions:

1) Which classes of control systems admit symbolic models?
2) Can these symbolic models be efficiently computed?
3) How can we transfer properties to and from symbolic models?

The objective of this paper is precisely to identify a class of control systems admitting symbolic models in which the loss of information incurred in the passage from an infinite to finite number of states can be compensated by feedback. In particular, we will show that:

1) The dynamics of a stabilizable linear system can be recovered from its restriction, henceforth called ”sub-system”, to a finite number of states up to a certain resolution.
2) Symbolic controllers designed for a sub-system can be transformed into controllers for the original system enforcing specifications up to a certain resolution.

These are rewarding results since we can easily compute restrictions of linear control systems to a finite number of states resulting in finite models that can be integrated with finite models of software and hardware. Furthermore, control designs based on these models can then be converted into controllers for the original system. This design approach, by integrating continuous dynamics with software and hardware at design time, results in controllers described by hybrid systems formally describing embedded control software that is correct by construction.

On the technical side this paper is strongly influenced by the paper [KPS01] and draws inspiration from other symbolic control models such as quantized control systems [BMP02] and maneuver automata [FDF03]. Also related is the recent work on approximate bisimulations for continuous systems reported in [GP05]. This work contains two different ingredients that distinguishes it from previous work by the author on finite bisimulations based control [TP03], [TP04]:

- We consider sub-systems instead of quotient systems for symbolic models and in particular our constructions will not be based of partitions of the state space but will rather require coverings of the state space.
- The symbolic models discussed in this paper are not bisimulations. Even though symbolic sub-systems may not capture all the behavior of the original system, we can synthesize controllers based on very simple and therefore very efficiently computable symbolic sub-systems.

This research was partially supported by the National Science Foundation CAREER award 0446716.

0-7803-9568-9/05/$20.00 ©2005 IEEE

II. Notation

We shall denote by $\mathbb{N}$, $\mathbb{R}$, and $\mathbb{R}_0^+$ the Natural, Real, and non-negative Real numbers, respectively. For any function $f : A \to B$ and $C \subseteq A$, $f|_C : C \to B$ will denote the restriction of f to C while $f(C)$ will denote the subset of B defined by $\cup_{c \in C}\{f(c)\}$. We will identify a relation $R \subseteq A \times B$ with the function $R : A \to 2^B$ defined by $b \in R(a)$ iff $(a, b) \in R$. We will say that a relation R is surjective when for every $b \in B$ there exists a $a \in A$ such that $(a, b) \in R$. If $\pi_2 : A \times B \to B$ denotes the canonical projection on the second factor then surjectivity of R is equivalent to $\pi_2(R) = B$. Given a relation $R \subseteq A \times B$, R^{-1} will denote the inverse relation defined by $R^{-1} = \{(b, a) \in B \times A \mid (a, b) \in R\}$.

We now recall some formal language notions. Given a set S we denote by S^* the set of all finite strings obtained by concatenating elements in S. An element of S is therefore given by $s_1 s_2 \ldots s_n$ with $s_i \in S$ for $i = 1, \ldots, n$. Given a string α belonging to S^* we denote by $\alpha(i)$ the ith element of s. The length of a string $\alpha \in S^*$ is denoted by $|\alpha|$ and a subset of S^* is called a language. Given a map $f : A \to B$ we shall use the same letter to denote the extension of f to $f : A^* \to B^*$ defined by:

$$f\big(\alpha(1)\alpha(2)\ldots\alpha(n)\big) = f\big(\alpha(1)\big)f\big(\alpha(2)\big)\ldots f\big(\alpha(n)\big)$$

III. Transition systems

Transition systems are the symbolic models considered in this paper. Because of their simplicity, transition systems model very general classes of dynamics including control systems, software systems and even hardware systems.

Definition 3.1: A transition system $T = (Q, Q^0, \longrightarrow, O, H)$ consists of:

- A set of states Q;
- A set of initial states $Q^0 \subseteq Q$;
- A transition relation $\longrightarrow \subseteq Q \times Q$;
- An observation set O;
- An observation function $H : Q \to 2^O$.

We will follow standard practice and denote an element $(q, q') \in \longrightarrow$ by $q \longrightarrow q'$. We shall say that a transition system T is finite when Q is finite. Transition systems capture dynamics through the transition relation. For any states $q, q' \in Q$, $q \longrightarrow q'$ simply means that it is possible to evolve or jump from state q to state q'. Note that we cannot model $\longrightarrow$ as a function since, in general, there may be several states $q', q'' \in Q$ such that $q \longrightarrow q'$ and $q \longrightarrow q''$. Transition systems can be graphically represented through a directed graph having Q as vertex set and $\longrightarrow$ as the set of edges.

Transition systems define languages which we could regard as the analogue of control systems' trajectories:

Definition 3.2: Let T be a transition system. A run of T is a string $\alpha \in Q^*$ satisfying:

1) $\alpha(1) \in Q^0$;
2) $\alpha(i) \longrightarrow \alpha(i+1)$ for $i = 1, 2, \ldots, |\alpha| - 1$.

A string $\beta \in O^*$ is said to be an observed run of T if there exists a run $\alpha \in Q^*$ of T such that $\beta \in H(\alpha)$. The language of T, denoted by $L(T)$, is defined as the set of all observed runs of T.

Control systems can also be seen as transition systems. Before discussing how we can embed the class of control systems into the class of transition systems we introduce the class of control systems considered in this paper.

Definition 3.3: A linear control system $\Sigma = (A, B, \mathcal{U})$ is defined a pair of matrices $(A, B) \in \mathbb{R}^{n \times n} \times \mathbb{R}^{n \times m}$ and a family of admissible input trajectories $\mathcal{U}$. A curve $\mathtt{x} : I \to \mathbb{R}^n$, defined on a open set $I \subseteq \mathbb{R}$ containing the origin, is a trajectory of control system Σ if there exists an admissible input trajectory $\mathcal{U} \ni \mathtt{u} : I \to \mathbb{R}^m$ satisfying:

$$\frac{d}{dt}\mathtt{x}(t) = A\mathtt{x}(t) + B\mathtt{u}(t) \qquad \text{(III.1)}$$

for almost all $t \in I$.

We will frequently refer to trajectories $\mathtt{x} : [0, \tau] \to \mathbb{R}^n$ of Σ defined on closed intervals with the understanding of the existence of a trajectory $\mathtt{x}' : I \to \mathbb{R}^n$ satisfying Definition 3.3 with $[0, \tau] \subset I$ and $\mathtt{x}'|_{[0,\tau]} = \mathtt{x}$. The results presented in this paper are independent of the chosen class of admissible input trajectories $\mathcal{U}$ provided that for each $\mathtt{u} \in \mathcal{U}$ the solution of (III.1) exists and is unique. Examples of admissible input trajectories include the class of piece-wise constant, piece-wise continuous and piece-wise smooth curves.

We now introduce the promised embedding of linear control systems in the class of transition systems.

Definition 3.4: Let Σ be a linear control system. The transition system induced by Σ, denoted by $T_\Sigma = (Q, Q^0, \longrightarrow, O, H)$, is defined by:

1) $Q = \mathbb{R}^n$;
2) $Q^0 = Q$;
3) $x \longrightarrow x'$ if there exists a trajectory $\mathtt{x} : [0, \tau] \to \mathbb{R}^n$ of Σ satisfying $\mathtt{x}(0) = x$ and $\mathtt{x}(\tau) = x'$;
4) $O = Q$;
5) $H(x) = \{x\}$.

IV. Simulation and bisimulation relations

The objective of this paper is to transfer control design problems from a continuous model Σ to a symbolic model. This transfer is only possible if the symbolic model captures properties of Σ that are relevant for design. While the standard notion of equivalence between transition systems is bisimulation [Par81], [Mil89] we shall work with a one-sided version termed simulation.

Definition 4.1: Let $T_i = (Q_i, Q_i^0, \longrightarrow_i, O, H_i)$ with $i = 1, 2$ be transition systems and let $R \subseteq Q_1 \times Q_2$ be a relation. Relation R is said to be a simulation relation from T_1 to T_2 if the following holds:

1) **(R respects observations)** $(q_1, q_2) \in R$ implies $H_1(q_1) \subseteq H_2(q_2)$;
2) **(R respects initial states)** $(q_1, q_2) \in R$ and $q_1 \in Q_1^0$ implies $q_2 \in Q_2^0$;
3) **(R respects dynamics)** $(q_1, q_2) \in R$ and $q_1 \longrightarrow_1 q_1'$ in T_1 implies the existence of $q_2' \in Q_2$ satisfying $q_2 \longrightarrow_2 q_2'$ in T_2 and $(q_1', q_2') \in R$.

The existence of a simulation relation from T_1 to T_2 is denoted by $T_1 \prec T_2$. Relation R is said to be a bisimulation relation between T_1 and T_2 if R is a simulation from T_1 to T_2 and R^{-1} is a simulation from T_2 to T_1. The existence of a bisimulation relation between T_1 and T_2 is denoted by $T_1 \cong T_2$ and T_1 and T_2 are said to be bisimilar.
The symbolic models of linear control systems Σ we will consider in this paper are related to T_Σ through a simulation relation which is in fact the graph of an inclusion.

Definition 4.2: Let $T_i = (Q_i, Q_i^0, \longrightarrow_i, O, H_i)$ with $i = 1, 2$ be transition systems. Transition system T_1 is said to be a sub-system of T_2 if $Q_1 \subseteq Q_2$ and the relation defined by the graph of the natural inclusion $\imath : Q_1 \to Q_2$ sending $q \in Q_1$ to $\imath(q) = q \in Q_2$ is a simulation relation from T_1 to T_2. Transition system is said to be a symbolic sub-system if it is a finite sub-system.

Although there are many different ways to construct a symbolic sub-system of T_Σ we now illustrate one such possibility based on quantization of inputs as studied in [BMP02]. For simplicity of presentation let us consider the double integrator as our control system Σ which is described by the following equations:

$$\begin{aligned} \dot{x}_1 &= x_2 \\ \dot{x}_2 &= u \end{aligned} \tag{IV.1}$$

We now chose a subset of admissible trajectories defined by $\mathcal{U} = \{\mathtt{u}_-, \mathtt{u}_0, \mathtt{u}_+\}$. Each $\mathtt{u} \in \mathcal{U}$ is defined on the interval $I = [0, 1]$ by:

$$\mathtt{u}_-(t) = -1, \quad \mathtt{u}_0(t) = 0, \quad \mathtt{u}_+(t) = +1 \tag{IV.2}$$

Since all the elements of $\mathcal{U}$ have unit duration we can compute the discrete time model of Σ for a sampling period of unit duration. The resulting discrete-time linear system is given by:

$$\begin{aligned} x_1(k+1) &= x_1(k) + x_2(k) + \frac{1}{2}u(k) \\ x_2(k+1) &= x_2(k) + u(k) \end{aligned} \tag{IV.3}$$

If we start at the origin, for example, an apply inputs (IV.2) to (IV.1) we can compute with the help of (IV.3) the symbolic sub-systems represented in Figure 1. These symbolic sub-systems represent a very coarse description of the dynamics of Σ which, nevertheless, can be used to synthesize controllers for Σ. For example the sequence of inputs:

$$\mathtt{u}_-(\mathtt{u}_+\mathtt{u}_+\mathtt{u}_0\mathtt{u}_-\mathtt{u}_-\mathtt{u}_0)^\omega$$

where ω denotes infinite repetition, controls T_Σ from the origin to a closed orbit. But how can we control the behavior of T_Σ if the initial condition does not belong to the set of states of T? And what kind of control can we expect when using a coarse model such as T? Answers to these questions will be provided in the remaining paper independently of the process used to obtain sub-systems. Other possibilities to compute symbolic sub-systems include numerical methods or the use of feedback controllers leading to known motion patterns.

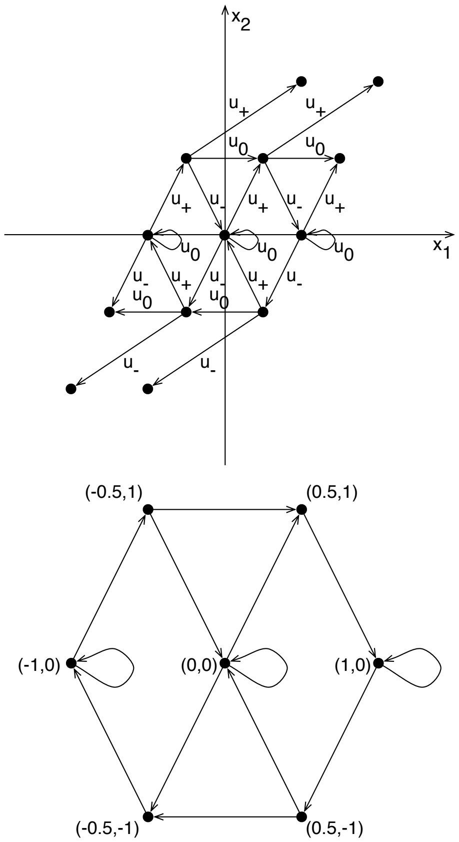

Fig. 1. Two different symbolic sub-systems of system (IV.1). On the top we label transitions with the corresponding input trajectory while on the bottom we show the coordinates of each state.

V. Symbolic control based on symbolic sub-systems

We start by reviewing the notion of parallel composition that models synchronization of transition systems on the common output space.

Definition 5.1: The parallel composition of transition systems $T_i = (Q_i, Q_i^0, \longrightarrow_i, O, H_i)$, $i = 1, 2$, is denoted by $T_1 \parallel T_2$ and defined as the transition system $T_1 \parallel T_2 = (Q_{12}, Q_{12}^0, \longrightarrow_{12}, O, H_{12})$ consisting of:

- $Q_{12} = \{(q_1, q_2) \in Q_1 \times Q_2 \mid H_1(q_1) \cap H_2(q_2) \neq \varnothing\}$;
- $Q_{12}^0 = \{(q_1, q_2) \in Q_1^0 \times Q_2^0 \mid H_1(q_1) \cap H_2(q_2) \neq \varnothing\}$;
- $(q_1, q_2) \longrightarrow_{12} (q_1', q_2')$ for $(q_1, q_2), (q_1', q_2') \in Q_{12}$ if $q_1 \longrightarrow_1 q_1'$ in T_1 and $q_2 \longrightarrow_2 q_2'$ in T_2;
- $H_{12}(q_1, q_2) = H_1(q_1) \cap H_2(q_2)$.

The language of the parallel composition $T_1 \parallel T_2$ can be expressed in terms of the languages of T_1 and T_2 by $L(T_1 \parallel T_2) = L(T_1) \cap L(T_2)$. Since composing T_1 with T_2 has the effect of restricting the language of T_2 we can thus think of T_1 as a controller for T_2 which prevents the strings in $L(T_2) \backslash L(T_1)$ from happening.

The following observation is crucial for the results developed in this section:

Once a linear control system Σ is embedded in the class of transition systems as T_Σ, controller synthesis for Σ can be identified with controller synthesis for T_Σ.

At a transition system level there are essentially two different types of control problems to be considered: linear time control and branching time control.

Problem 5.2 (Linear Time Control): Given a transition system $T = (Q, Q^0, \longrightarrow, O, H)$ and a language specification $S \subseteq O^*$, synthesize a controller T_c such that $L(T_c \parallel T) \subseteq S$ or $L(T_c \parallel T) = S$.

Problem 5.3 (Branching Time Control): Given a transition system $T = (Q, Q^0, \longrightarrow, O, H)$ and transition system specification $T_S = (Q_S, Q_S^0, \longrightarrow_S, O, H_S)$, synthesize a controller T_c such that $T_c \parallel T \prec T_S$ or $T_c \parallel T \cong T_S$.

Both control problems can be made more realistic by adding additional requirements and constraints such as non-blocking controllers, partial observability, maximal permissivity, etc. Nevertheless, the above described problems are sufficient to illustrate the merit of a symbolic approach to the control of continuous systems. The following result explains how we can transfer the design of controllers solving Problems 5.2 and 5.3 from T_Σ to a symbolic sub-system T.

Theorem 5.4: Let T_Σ be the transition system induced by a linear control system Σ and assume that transition system T satisfies $T \prec T_\Sigma$. Then, for any specification transition system T_S with language $S = L(T_S)$ the following holds:

1) If there exists a controller T_c such that $L(T_c \parallel T) \subseteq S$, then controller $T_c' = T_c \parallel T$ satisfies $L(T_c' \parallel T_\Sigma) \subseteq S$.
2) If there exists a controller T_c such that $L(T_c \parallel T) = S$, then controller $T_c' = T_c \parallel T$ satisfies $L(T_c' \parallel T_\Sigma) = S$.

Furthermore, if $T \parallel T_\Sigma \cong T$ also holds, then:

3) If there exists a controller T_c such that $T_c \parallel T \prec T_S$, then controller $T_c' = T_c \parallel T$ satisfies $T_c' \parallel T_\Sigma \prec T_S$.
4) If there exists a controller T_c such that $T_c \parallel T \cong T_S$, then controller $T_c' = T_c \parallel T$ satisfies $T_c' \parallel T_\Sigma \cong T_S$.

Theorem 5.4 shows that existence of a controller T_c for T immediately leads to a controller for T_Σ. Furthermore, when T is finite existing supervisory control [KG95], [CL99] and controller synthesis [Tab04], [AVW03], [MT02], [Tho95] techniques can be immediately used for the construction of T_c. In addition to provide a new computational approach to controller synthesis problems for continuous control systems, Theorem 5.4 also shows that it is now possible to design controllers based on specifications that, traditionally, have not been considered for continuous systems such as regular languages, transitions systems, temporal logics, etc. Furthermore, by combining symbolic model T with a transition system model of existing software and hardware it is possible to synthesize controllers enforcing control specifications (describing the desired behavior of the continuous dynamics) and software specifications (describing the desired behavior of the control code). The resulting controller can then be refined to a hybrid system model of control software that is correct by construction. The construction of such hybrid controllers is discussed in Section VII-B.

VI. Surjective simulation relations

We have seen in the previous section that it is possible to synthesize controllers for T_Σ by working with the simpler symbolic model T. However, such designs result in controllers that can only be applied at states of T_Σ that are also states of T. To see this, note that (q', r) is a state of $T_c' \parallel T_\Sigma$ only if $H_c'(q') \cap H_\Sigma(r) \neq \varnothing$ and $q' = (q, p)$ is a state of $T_c \parallel T$ only if $H_c(q) \cap H(p) \neq \varnothing$. Since $H(p) = \{p\}$ and $H_\Sigma(r) = \{r\}$, $H_c(q) \cap H(p) \cap H_\Sigma(r) \neq \varnothing$ only if $p = r \in Q$ which shows that T_c' is a controller that only works for states of T_Σ that are also states of T. In order to extend symbolic controllers to controllers that can be used at any state of T_Σ we need to extend the simulation relation defined by the graph of the inclusion $\imath : Q \to Q_\Sigma$ to a surjective simulation relation $R \subseteq Q \times Q_\Sigma$. If such extension exists, then for any point $r \in Q_\Sigma$ we can obtain a point $q \in Q$ which is R-related to r and apply an input at r based on the input defined by T_c' at q. In this section we will show that such extension is possible under a stabilizability assumption on Σ and by restricting attention to a bounded region of the state space. Recall that a linear control system is stabilizable if there exists a linear feedback $u = Kx$ making $\dot{x} = (A + BK)x$ stable and is asymptotically stabilizable if $u = Kx$ makes $\dot{x} = (A + BK)x$ asymptotically stable.

Theorem 6.1: Let $T_\Sigma = (Q_\Sigma, Q_\Sigma^0, \longrightarrow_\Sigma, O_\Sigma, H_\Sigma)$ be the transition system associated with a linear control system Σ. If Σ is stabilizable, then for any control Lyapunov function V of Σ, for any symbolic sub-system $T = (Q, Q^0, \longrightarrow, O, H)$ of T_Σ and for any bounded subset Q_Σ' of Q_Σ containing Q, there exists a real number $\mu \in \mathbb{R}$ such that:

1) $R \subseteq Q \times Q_\Sigma$, defined by $(q, x) \in R$ when $V(x - q) \leq \mu$, is a simulation relation from T to T_Σ' satisfying $\pi_2(R) = Q_\Sigma$;
2) $T \parallel T_\Sigma' \cong T$,

where $T_\Sigma' = (Q_\Sigma', Q_\Sigma', \longrightarrow, O, H_\Sigma')$ with $q \in H_\Sigma'(x)$ when $V(x - q) \leq \mu$.

Intuitively, Theorem 6.1 shows that we can use a stabilizing controller to robustify controller T_c'. This is done by using the input trajectory $\mathtt{u}$ associated with a transition $q \longrightarrow q'$ in T to compute a new input trajectory:

$$\mathtt{u} - K(\mathtt{q} - \mathtt{x}) \qquad \text{(VI.1)}$$

to be applied at points $x \in Q_\Sigma$ satisfying $V(x - q) \leq \mu$. Input trajectory (VI.1) controls points x that are close to q (points satisfying $V(x - q) \leq \mu$) to points x' that are close to q' (points satisfying $V(x' - q') \leq \mu$).

We now revisit the double integrator example with the purpose of illustrating Theorem 6.1. For symbolic sub-system we consider the transition system displayed on the bottom of Figure 1. We now chose Q_Σ' to be the closed ball of radius $3/2$ centered at the origin which is guaranteed to contain all the points of Q. If we now use $V = x_1^2 + x_2^2$ as a control Lyapunov function for the double integrator we can take $\mu = 1$ which results in each set $V_q^\mu = \{x \in \mathbb{R}^2 \mid V(x - q) \leq \mu = 1\}$ being a closed ball of radius 1

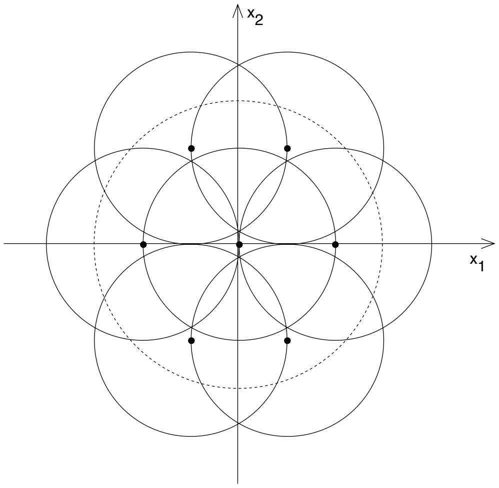

Fig. 2. Covering of Q'_Σ by disks of radius 1. The set Q'_Σ is enclosed by the dashed circle while the sets $V_q^\mu = \{x \in \mathbb{R}^2 \mid V(x-q) \leq \mu = 1\}$ are enclosed by the solid circles.

centered at q. In Figure 2 we can see how this choice results in a covering for Q'_Σ.

By analyzing Figure 2 we also see that Theorem 6.1 is not entirely satisfactory since we are only able to exert very coarse control in the sense that we cannot distinguish between points $x, y \in Q'_\Sigma$ if $V(x-q) \leq \mu$ and $V(y-q) \leq \mu$ for some $q \in Q$. In fact, the parameter μ provides a measure of such coarseness. This difficulty can be addressed in two different ways. We can construct a more detailed symbolic sub-system T which would lead to a lower value for μ resulting in less uncertainty in the position of the state. Or we can use feedback to reduce the uncertainty associated with the location of the state as discussed in detail in the next section.

VII. From symbolic controllers to hybrid systems controllers

A. Reducing uncertainty

In this section we strengthen the stabilizability assumption on Σ to asymptotic stabilizability. Based on this assumption we will be able to use feedback to reduce the coarseness of the exerted control.

Asymptotic stabilizability of Σ implies the existence of a linear feedback $u = Kx$ and of a Lyapunov function $V : \mathbb{R}^n \to \mathbb{R}_0^+$ satisfying:

$$\frac{\partial V}{\partial x}(Ax + BKx) = \dot{V} = -\alpha V \qquad \text{(VII.1)}$$

for some positive constant α that we will call rate of decay. We will also say that V admits a rate of decay α if there exists a linear feedback $u = Kx$ such that (VII.1) holds. Integrating (VII.1) we obtain:

$$V(t) = V(0)e^{-\alpha t}$$

which shows that the uncertainty in the location of the state is reduced by the factor $0 < e^{-\alpha\tau} < 1$ every time that a control command is executed. This suggests that we should use a symbolic model T describing the number of implemented control commands in addition to its effect on the states. For simplicity of presentation we will assume throughout this section that any $q \longrightarrow q'$ in T has been obtained through an input trajectory of length τ and we will denote by σ the number $\sigma = e^{-\alpha\tau}$.

Definition 7.1: Let T_Σ be the transition system induced by a linear control system Σ. For any sub-system $T = (Q, Q^0, \longrightarrow, O, H)$ of T_Σ, $T_{\mathbb{N}_0}$ denotes the transition system defined by $T_{\mathbb{N}_0} = (Q \times \mathbb{N}_0, Q^0 \times \mathbb{N}_0, \longrightarrow_{\mathbb{N}_0}, O \times \mathbb{N}_0, H_{\mathbb{N}_0})$ where $(q, n) \longrightarrow_{\mathbb{N}_0} (q', n')$ if $q \longrightarrow q'$ in T and $n' = n+1$, and $H_{\mathbb{N}_0}(q, n) = \{(q, n)\}$.

Intuitively, a state (q, n) of $T_{\mathbb{N}_0}$ counts the number n of transitions required to reach that state from some state in $Q^0 \times \{0\}$. Since each transition results in a better estimate for the state location we can now synthesize controllers guaranteeing not only logic but also quantitative specifications. We thus have the following "graded" version of Theorem 6.1.

Theorem 7.2: Let $T_\Sigma = (Q_\Sigma, Q_\Sigma^0, \longrightarrow_\Sigma, O_\Sigma, H_\Sigma)$ be the transition system associated with a linear control system Σ. If Σ is asymptotically stabilizable, then for any control Lyapunov function V of Σ admitting rate of decay α, for any symbolic sub-system $T = (Q, Q^0, \longrightarrow, O, H)$ of T_Σ and for any bounded set $Q'_\Sigma \subseteq Q_\Sigma$ containing Q, there exists a real number $\mu \in \mathbb{R}$ such that:

1) $R \subseteq (Q \times \mathbb{N}_0) \times Q_\Sigma$, defined by $((q, n), x) \in R$ when $V(x - q) \leq \mu\sigma^n$, is a simulation relation from $T_{\mathbb{N}_0}$ to T'_Σ satisfying $\pi_2(R) = Q_\Sigma$;
2) $T_{\mathbb{N}_0} \parallel T'_\Sigma \cong T_{\mathbb{N}_0}$.

where $T'_\Sigma = (Q'_\Sigma, Q'_\Sigma, \longrightarrow_\Sigma, O \times \mathbb{N}_0, H'_\Sigma)$ with $(y, n) \in H'_\Sigma(x)$ when $V(x - y) \leq \mu\sigma^n$.

Consider again the double integrator and the following control Lyapunov function:

$$V = \frac{1}{2}\left(x_1^2 + x_1x_2 + x_2^2\right)$$

admitting a rate of decay $\alpha = 1$ for the linear feedback $u = -x_1 - x_2$. As symbolic sub-system T we consider again the transition system represented on the bottom of Figure 1. The sequence of inputs $\mathtt{u}_-\mathtt{u}_+\mathtt{u}_+\mathtt{u}_0\mathtt{u}_-\mathtt{u}_-\mathtt{u}_+$ guarantees that any state x contained in the disk of radius 1 centered around the origin and corresponding to observation $\{((0,0),0)\} \in H'_\Sigma(x)$ will be controlled to some point x' in the disk of radius $\sigma^7 \approx 0.0009$ centered at the origin and corresponding to observation $\{((0,0),7)\} \in H'_\Sigma(x')$. The result of this sequence of inputs can be seen on the top of Figure 3 for three different initial conditions. By inspecting T we see that there are other sequences of inputs controlling observation $((0,0),0)$ to observation $((0,0),7)$. The result of one such sequence, $\mathtt{u}_-\mathtt{u}_+\mathtt{u}_0\mathtt{u}_+\mathtt{u}_-\mathtt{u}_0\mathtt{u}_0$, is displayed on the bottom of Figure 3 for the same initial conditions. Even though the problem of transferring states corresponding to observation $\{((0,0),0)\}$ to states corresponding to observation $\{((0,0),7)\}$ can be solved by inspection of T, more complex specifications require a more systematic approach based on supervisory control techniques.

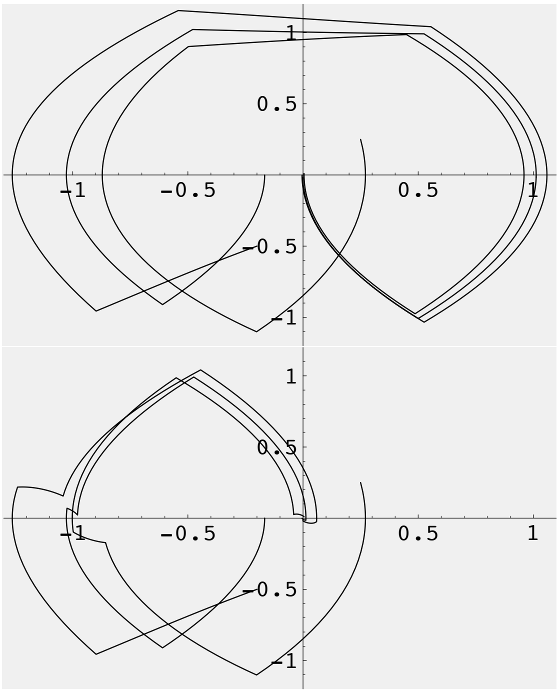

Fig. 3. Initial conditions $(1/4, 1/4)$, $(-1/5, -1/2)$ and $(-1/6, 0)$ corresponding to observation $\{((0,0),0)\}$ and controlled to observation $\{((0,0),7)\}$. On the left, control is enforced by the sequence of input trajectories $\mathtt{u}_-\mathtt{u}_+\mathtt{u}_+\mathtt{u}_0\mathtt{u}_-\mathtt{u}_-\mathtt{u}_+$ while on the right by the sequence $\mathtt{u}_-\mathtt{u}_+\mathtt{u}_0\mathtt{u}_+\mathtt{u}_-\mathtt{u}_0\mathtt{u}_0$.

B. *Controllers as hybrid systems*

Symbolic controllers T_c designed for symbolic model T abstract representations that do not specify which control signals should be sent to the continuous plant in order to enforce the desired behavior. These more detailed controllers can be immediately obtained by associating to a symbolic state q the feedback control laws enforcing transitions outgoing from q. By repeating this process for every state we obtain a hybrid system model of the desired controller. If we denote by H_c the hybrid controller obtained from symbolic controller T_c through this process and if $H_c \parallel \Sigma$ represents the closed loop system, then H_c can be seen as an implementation of the abstract controller T_c in the sense that $T_c \parallel T_\Sigma \cong T_{H_c\parallel\Sigma}$ where $T_{H_c\parallel\Sigma}$ is the transition system capturing the behavior of the closed loop system $H_c \parallel \Sigma$.

VIII. Discussion

In this paper we have shown that symbolic sub-systems can be used as abstract models of stabilizable linear systems for control design. The loss of information incurred in the passage from a model with an infinite number of states to a model with a finite number of states can be compensated by feedback. We have shown how to construct feedback control laws providing such compensation which combined with symbolic supervisors designed for symbolic sub-systems results in hybrid systems models for controllers. Furthermore, since we can combine symbolic sub-systems with finite models of software and hardware, the synthesis of symbolic supervisors can address in a integrated fashion specifications stemming from the continuous dynamics, from software and even from hardware. The proposed design methodology is then guaranteed to produce hybrid controllers which, if regarded as models for embedded control software, require no further verification or validation as they satisfy the desired specifications by construction.

It remains to be investigated how existing results on the existence of finite bisimulations for discrete-time linear control systems can be related to the results presented in this paper. Of particular importance are methodologies for the choice of symbolic sub-systems. Even though the presented results are applicable to any symbolic sub-system, a criteria to obtain complete (describing all the behavior of the original system up to a certain resolution) and yet small sub-systems would be extremely important in practice.

References

[AVW03] A. Arnold, A. Vincent, and I. Walukiewicz. Games for synthesis of controllers with partial observation. *Theoretical Computer Science*, 28(1):7–34, 2003.

[BMP02] A. Bicchi, A. Marigo, and B. Piccoli. On the rechability of quantized control systems. *IEEE Transaction on Automatic Control*, April 2002.

[CL99] C. Cassandras and S. Lafortune. *Introduction to discrete event systems*. Kluwer Academic Publishers, Boston, MA, 1999.

[FDF03] E. Frazzoli, M.A. Dahleh, and E. Feron. A maneuver-based hybrid control architecture for autonomous vehicle motion planning. In G. Balas and T. Samad, editors, *Software Enabled Control: Information Technology for Dynamical Systems*. IEEE Press, 2003.

[GP05] Antoine Girard and George J. Pappas. Approximation metrics for discrete and continuous systems. *Technical Report MS-CIS-05-10*, May 2005. Dept. of CIS, University of Pennsylvania.

[KG95] R. Kumar and V.K. Garg. *Modeling and Control of Logical Discrete Event Systems*. Kluwer Academic Publishers, 1995.

[KPS01] T. J. Koo, G. J. Pappas, and S. Sastry. Mode switching synthesis for reachability specifications. In M. D. Di Benedetto and A. Sangiovanni-Vincentelli, editors, *Hybrid Systems: Computation and Control*, volume 2034 of *Lecture Notes in Computer Science*, pages 333–346. Springer-Verlag, 2001.

[Mil89] R. Milner. *Communication and Concurrency*. Prentice Hall, 1989.

[MT02] P. Madhusudan and P.S. Thiagarajan. Branching time controllers for discrete event systems. *Theoretical Computer Science*, 274:117–149, March 2002.

[Par81] D.M.R. Park. Concurrency and automata on infinite sequences. volume 104 of *Lecture Notes in Computer Science*, pages 167–183, 1981.

[Tab04] Paulo Tabuada. Open maps, alternating simulations and controller synthesis. In Philippa Gardner and Nobuko Yoshida, editors, *Proceedings of the 15th International Conference on Concurrence Theory*, volume 3170 of *Lecture Notes in Computer Science*, pages 466–480, London, UK, 2004. Springer.

[Tho95] W. Thomas. On the synthesis of strategies in infinite games. In *Proceedings of the 12th Annual Symposium on Theoretical Aspects of Computer Science*, volume 900 of *Lecture Notes in Computer Science*, pages 1–13. Springer, 1995.

[TP03] Paulo Tabuada and George J. Pappas. Finite bisimulations of controllable linear systems. In *Proceedings of the 42nd IEEE Conference on Decision and Control*, Hawaii, 2003.

[TP04] Paulo Tabuada and George J. Pappas. Linear Time Logic control of linear systems. February 2004. Under review, available at `http://www.nd.edu/~ptabuada`.