Symbolic Control of Linear Systems Based on Symbolic Subsystems

Paulo Tabuada, *Member, IEEE*

***Abstract*—This paper describes 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 subsystems can be compensated by feedback. We also show how to transform symbolic controllers designed for a symbolic subsystem 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.**

***Index Terms*—Bisimulation, simulation, symbolic control, symbolic subsystems.**

I. Introduction

A. Motivation

RESEARCH in systems and control theory is steadily shifting attention from single monolithic systems to large-scale complex systems often built from simpler subsystems. This change reflects the problems faced by practicing engineers when analyzing and designing nowadays complex control systems such as power networks, ad hoc networks of sensor/actuators, transportation systems, biological systems, enterprise systems, etc. From a systems theoretic point of view, there is a common question arising in the analysis and control of these applications:

> How do we manage the intrinsic complexity of such systems?

One approach to tame the complexity of these applications, advocated by many researchers over the years, is the use of abstraction to model, analyze and control systems at different levels of detail. This requires the construction of different models capturing different aspects of the system being analyzed/designed, and the construction of relations between these models explaining how the results of analysis/design for one model can be used in another model. Analysis/design could then be performed on simpler models thereby reducing the complexity of these tasks.

In this paper, we take important steps along this direction by focusing on the control of continuous time 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. First, 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. Second, 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 and software 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 subsystems*.

B. Contributions

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 a finite number of states can be compensated by feedback. In particular, we will show that the following hold.

1) The dynamics of a stabilizable linear system can be recovered from its restriction to a finite number of states, henceforth called "subsystem", up to a certain resolution.
2) Symbolic controllers designed for a subsystem 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

Manuscript received December 29, 2004; revised August 15, 2005. Recommended by Guest Editor E. Frazzoli. This work was supported in part by the National Science Foundation under CAREER award 0446716.

The author is with the Department of Electrical Engineering, University of Notre Dame, Notre Dame, IN 46556 USA (e-mail: ptabuada@nd.edu).

Digital Object Identifier 10.1109/TAC.2006.876946

original system. This approach, by integrating continuous dynamics with software and hardware at design time, results in controllers modeled by hybrid systems which formally describe embedded control software that is correct by construction.

On the technical side, this paper is strongly influenced by [27] and draws inspiration from other symbolic control models such as quantized control systems [6], [37] and maneuver automata [16]. The arguments used in this paper can be seen as refined versions of the ideas described in [27]. However, contrary to [27] we do not require the existence of trajectory tracking controllers neither do we impose conditions for switching between different controllers. Quantized control systems [6], [37] can also be seen as symbolic models of continuous systems leading to simpler control designs. These control designs are based on the observation that for certain classes of systems the dynamics assumes an especially simple form when restricted to a lattice structure describing the reachable set. In our approach, we focus on the finite structure of subsystems rather than on the lattice theoretic properties of reachable sets. Also, one of the main objectives of this paper is to show that symbolic subsystems can be used to design controllers that act on the whole continuous state space rather than on the subset of points over which the subsystem is defined. The symbolic models we use throughout this paper describe the behavior of the original system for certain choices of input trajectories as is also the case for maneuver automata. However, we impose no conditions on the choice of input trajectories and this distinguishes our work from [16], where the choice of trajectories is based on the existence of certain symmetries.

Finally, this work contains two different ingredients that distinguishes it from previous work by the author on finite bisimulations based control [43], [48]–[51].

- We consider subsystems 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 subsystems may not capture all the behavior of the original system, we can still synthesize controllers based on very simple and, therefore, very efficiently computable symbolic subsystems.

C. Related Work

Symbolic models of continuous/discrete dynamics, usually in the form of finite bisimulations, have been traditionally associated with problems of verification. Starting with Alur and Dill's work on timed automata [2], a fair amount of work was done to push the boundaries of the class of systems admitting finite bisimulations. This work culminated with [1], [18], [19], and [38] introducing the decidable classes of multi-rate and rectangular hybrid automata. On the purely continuous side, we mention the work of Lafferriere *et al.* [28] which used o-minimality to ensure existence of symbolic models. See also [7] for a simpler and more insightful proof of the same results. The aforementioned efforts suggest that finite bisimulations only exist under quite restrictive conditions and this observation led several researchers to investigate approximate notions [9] as well as semialgorithms [20], [46]. Especially related to the work described in this paper is the recent notion of approximate bisimulation introduced in [17]. Further comments on the relation between this notion and the symbolic models introduced in this paper are postponed until Section VIII.

The approach advocated in this paper is quite different from the above mentioned ones since one is interested in developing symbolic models for *control* rather than for *verification*. This different perspective justifies why this paper focuses on *control systems* rather than on *dynamical systems*. Initial approaches to the development of symbolic models of control systems seem to have been based on the use of integrals of motion [8], [41]. Integrals or constants of motion are a quite natural way of defining state space partitions compatible with the continuous dynamics. The results in this paper, however, apply to a class of systems for which constants of motion do not necessarily exist and can therefore be seen as complementary to existing work, especially to [8]. A different but related line of research consists on the study of bisimulations of purely continuous [35], [52], [53] or hybrid systems [21], [39]. Even though these works do not consider the construction of symbolic models, they represent another approach to complexity reduction through the use of bisimulation based abstractions. Other symbolic approaches to complexity reduction include the use of finite state machines to combine maneuvers in motion planning problems [16], motion description languages [23] and a study on the interplay between feedback and specification complexity of control tasks [15].

Finally, we would like to emphasize that the idea of using symbolic models for the control of continuous systems is not new and motivated much research in the area of hybrid systems [4], [12], [14], [24], [29], [32], [34], [40], [42], [47], [54]. Even though the use of symbolic models was advocated by these and many other researchers, the applicability of the proposed methods has always remained an open problem due to the lack of results ensuring existence of symbolic models for control systems. The results of this paper partially address this issue by identifying a class of systems for which symbolic models exists and can be easily computed.

D. Organization

This paper starts by introducing some notation in Section II and by reviewing a mathematical model describing continuous dynamics, software and hardware in Section III. We then introduce the notions of simulation and bisimulation in Section IV allowing us to transfer analysis and design problems from the original system to simpler symbolic models as described in Section V. In Section VI, we show how feedback can be used to control the original system at states not described by the symbolic subsystem. This use of feedback is further refined in Section VII leading to hybrid systems models for controllers acting on the original continuous system. This paper ends in Section IX with a critical discussion of the presented results.

II. Notation

We shall denote by $\mathbb{N}$, $\mathbb{R}$, and $\mathbb{R}_0^+$ the Natural, Real, and nonnegative Real numbers, respectively. For any function $f : A \rightarrow B$ and $C \subseteq A$, $f|_C : C \rightarrow B$ 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 $R^{-1} = \{(b, a) \in B \times A | (a, b) \in R\}$. A continuous function $\gamma : [0, a[\to \mathbb{R}_0^+$, $a > 0$, is said to belong to class $\mathcal{K}$ if it is strictly increasing and $\gamma(0) = 0$. It is said to belong to class $\mathcal{K}_\infty$ if $a = \infty$ and $\gamma(r) \to 0$ as $r \to \infty$. A continuous function $\beta : [0, a[\times \mathbb{R}_0^+ \to \mathbb{R}_0^+$ is said to belong to class $\mathcal{KL}$ if, for each fixed s, the map $\beta(r, s)$ belongs to class $\mathcal{K}$ with respect to r and, for each fixed r, the map $\beta(r, s)$ is decreasing with respect to s and $\beta(r, s) \to 0$ as $s \to \infty$.

Given a point $x \in \mathbb{R}^n$, $|x|$ will denote the usual Euclidean norm while $\|\mathtt{x}\|_\infty$ will denote $\sup_{t \in [0,\tau]} |\mathtt{x}(t)|$ for any given function $\mathtt{x} : [0, \tau] \to \mathbb{R}^n$, $\tau > 0$.

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 will use the same letter to denote the extension of f to $f : A^* \to B^*$ defined by

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

III. Transition Systems

A. 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. In fact, much of the work on formal analysis and verification of software systems has some version of transition systems as its underlying model [11].

Definition 3.1: A transition system T is quintuple $(Q, Q^0, \longrightarrow, O, H)$ consisting 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 will 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 by a directed graph having Q as vertex set and $\longrightarrow$ as the set of edges. Transition system defined by

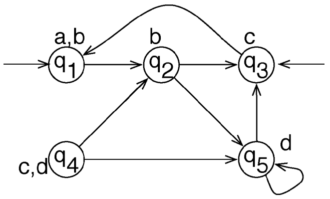

Fig. 1. Graphical representation of transition system defined by (1)–(5).

$$Q = \{q_1, q_2, q_3, q_4, q_5\} \quad Q^0 = \{q_1, q_3\} \tag{1}$$
$$O = \{a, b, c, d\} \tag{2}$$
$$H(q_1) = \{a, b\} \quad H(q_2) = \{b\} \quad H(q_3) = \{c\} \tag{3}$$
$$H(q_4) = \{c, d\} \quad H(q_5) = \{d\} \tag{4}$$

and

$$\longrightarrow = \{(q_1, q_2), (q_2, q_3), (q_3, q_1), (q_4, q_2), (q_4, q_5), (q_5, q_5), (q_5, q_3), (q_2, q_5)\} \tag{5}$$

has the graphical representation displayed in Fig. 1 where initial states are distinguished by having a sourceless incoming arrow.

Remark 3.2: The introduced notion of transition system differs from other notions encountered in the literature in that observations are not associated with transitions but rather with states. These two models can easily be seen equivalent given the well known equivalence between Moore and Mealy machines [22]. The presented model is, however, more natural since observations of control systems depend on the states and this structure is inherited by the several transition systems used in this paper to capture the dynamics of control systems.

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

Definition 3.3: 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 3.4: A linear control system Σ is a triple $(A, B, \mathcal{U})$ consisting of

- a matrix $A \in \mathbb{R}^{n \times n}$;
- a matrix $B \in \mathbb{R}^{n \times m}$;
- 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:

$$\frac{d}{dt}\mathtt{x}(t) = A\mathtt{x}(t) + B\mathtt{u}(t) \tag{6}$$

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 3.4 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 (6) 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 3.5: 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

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

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 [31], [36] 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) $(q_1, q_2) \in R$ implies $H_1(q_1) \subseteq H_2(q_2)$.
2) $(q_1, q_2) \in R$ and $q_1 \in Q_1^0$ implies $q_2 \in Q_2^0$.
3) $(q_1, q_2) \in R$ 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 relation from T_1 to T_2 and R^{-1} is a simulation relation 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 subsystem 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 T_1 is said to be a symbolic subsystem if it is a finite subsystem.

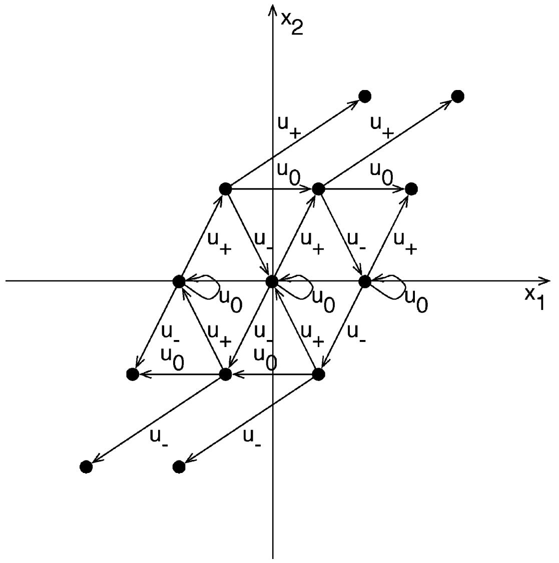

Fig. 2. Symbolic subsystem T_1 of T_Σ with Σ described by (7). Transitions are labeled with the corresponding input trajectory.

Although there are many different ways to construct a symbolic subsystem of T_Σ we now illustrate one such possibility based on quantization of inputs as extensively studied in [6] and [37]. For simplicity of presentation, let us consider the double integrator as our control system Σ which is described by the following equations:

$$\begin{aligned}\dot{x}_1 &= x_2\\ \dot{x}_2 &= u.\end{aligned} \tag{7}$$

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

$$\mathtt{u}_-(t) = -1 \quad \mathtt{u}_0(t) = 0 \quad \mathtt{u}_+(t) = +1. \tag{8}$$

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

$$\begin{aligned}x_1(k+1) &= x_1(k) + x_2(k) + \frac{1}{2}u(k)\\ x_2(k+1) &= x_2(k) + u(k).\end{aligned} \tag{9}$$

If we start at the origin, for example, an apply inputs (8) to (7) we can compute with the help of (9) the symbolic subsystems T_1 and T_2 represented in Figs. 2 and 3, respectively. These symbolic subsystems T_1 and T_2 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. However, how can we control the behavior of T_Σ if the initial condition does not belong to the set of states of

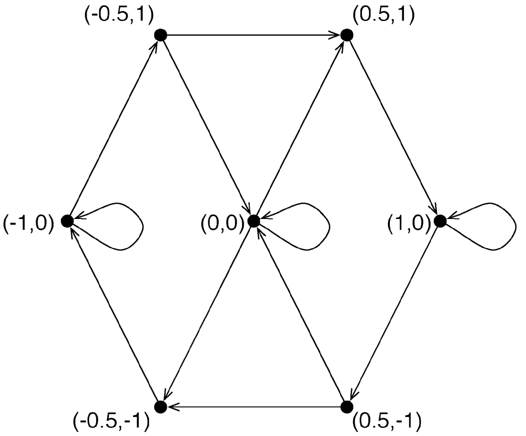

Fig. 3. Symbolic subsystem T_2 of T_Σ with Σ described by (7). Each state is represented as a point in $\mathbb{R}^2$ labeled by the corresponding coordinates.

T_1 or T_2? And what kind of control can we expect when using a coarse model such as T_1 or T_2? Answers to these questions will be provided in the remaining paper independently of the process used to obtain subsystems. Other possibilities to compute symbolic subsystems include numerical methods or the use of feedback controllers leading to known motion patterns.

V. Symbolic Control Based on Symbolic SubSystems

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

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

- $Q_{12} = \{(q_1, q_2) \in Q_1 \times Q_2 | H_1(q_1) \cap H_2(q_2) \neq \varnothing\}$;
- $Q_{12}^0 = \{(q_1, q_2) \in Q_1^0 \times Q_2^0 | H_1(q_1) \cap H_2(q_2) \neq \varnothing\}$;
- $(q_1, q_2) \longrightarrow_{12} (q_1', q_2')$ 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 think of T_1 as a controller for T_2 which prevents the strings in $L(T_2) \setminus L(T_1)$ from happening. In the next proposition we summarize several properties of simulations, bisimulations and parallel composition that will be required in the remaining paper.

Proposition 5.2: For transition systems T_1, T_2, and T_3 the following holds:

1) $T_1 \prec T_2 \Rightarrow L(T_1) \subseteq L(T_2)$;
2) $T_1 \prec T_2 \Rightarrow T_3 \parallel T_1 \prec T_3 \parallel T_2$;
3) $T_1 \prec T_2 \wedge T_2 \prec T_3 \Rightarrow T_1 \prec T_3$;
4) $T_1 \cong T_2 \Rightarrow L(T_1) = L(T_2)$;
5) $T_1 \cong T_2 \Rightarrow T_3 \parallel T_1 \cong T_3 \parallel T_2$;
6) $T_1 \cong T_2 \wedge T_2 \cong T_3 \Rightarrow T_1 \cong T_3$;
7) $T_1 \parallel T_2 \prec T_1$;
8) $T_1 \parallel T_2 \prec T_2$.

The results developed in this section are based on the following observation.

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 5.3: (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 5.4: (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 nonblocking controllers, partial observability, maximal permissivity, etc. Nevertheless, the previously 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 5.3 and 5.4 from T_Σ to a symbolic subsystem T.

Theorem 5.5: 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 the following hold.

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 $T_c' \parallel T_\Sigma \cong T_S$.

Proof: We follow the enumeration of the theorem and start by proving 1). By assumption, we have $T \prec T_\Sigma$ and it follows by Proposition 5.2:

$$L(T) \subseteq L(T_\Sigma). \tag{10}$$

Language $L(T_c' \parallel T_\Sigma)$ is now given by

$$\begin{aligned} & L(T_c \parallel T \parallel T_\Sigma) && \text{by definition of } T_c' && (11)\\ &= L(T_c) \cap L(T) \cap L(T_\Sigma) && \text{by definition of } \parallel && (12)\\ &= L(T_c) \cap L(T) && \text{by (10)} && (13)\\ &= L(T_c \parallel T) && \text{by definition of } \parallel . && (14)\end{aligned}$$

Since $L(T_c \parallel T) \subseteq S$ by assumption, we conclude $L(T_c' \parallel T_\Sigma) = L(T_c \parallel T) \subseteq S$ as desired. The proof of (2) is similar. We now prove (3).

$$\begin{aligned} & T_c' \parallel T_\Sigma \cong T_c \parallel T \parallel T_\Sigma && \text{by definition of } T_c' && (15)\\ & T_c' \parallel T_\Sigma \cong T_c \parallel T && \text{since } T \parallel T_\Sigma \cong T && (16)\\ & T_c' \parallel T_\Sigma \prec T_S && \text{since } T_c \parallel T \prec T_S. && (17)\end{aligned}$$

The proof of (4) is similar to the proof of (3). ■

Theorem 5.5 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 [10], [25], [30] and controller synthesis [5], [13], [26], [33], [44], [45] 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 5.5 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.

Remark 5.6: There is a natural tradeoff between the complexity (the size) of the symbolic subsystem T and the solutions to Problems 5.3 and 5.4 that can be found by working with T. Simpler subsystems T reduce the complexity of controller synthesis but they also lead to more restrictive controllers in the sense of preventing behaviors that are allowed by the specification. Ideally, one would like to work with subsystems that would be bisimulations so that no essential information is lost in replacing T_Σ with T. This problem of completeness will not be addressed in this paper and is further discussed in Section IX.

Remark 5.7: It may appear that assumption $T \prec T_\Sigma$ has not been used in the proof of (3) and (4). This is not the case since $T \cong T \parallel T_\Sigma$ combined with $T \parallel T_\Sigma \prec T_\Sigma$ leads to $T \prec T_\Sigma$. $T \cong T \parallel T_\Sigma$ is therefore a stronger assumption than $T \prec T_\Sigma$.

VI. Constructing 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', x) is a state of $T'_c \parallel T_\Sigma$ only if $H'_c(q') \cap H_\Sigma(x) \neq \varnothing$ 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 $H_\Sigma(x) = \{x\}$, $H_c(q) \cap H(p) \cap H_\Sigma(x) \neq \varnothing$ only if $p = x \in Q$ 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 $\imath : Q \rightarrow Q_\Sigma$ to a surjective simulation relation $R \subseteq Q \times Q_\Sigma$. If such extension exists, then for any point $x \in Q_\Sigma$ we can obtain a point $q \in Q$ which is R-related to x and apply an input at x 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. Corresponding to the linear feedback we have a quadratic Lyapunov function $V = x^T P x$, for a symmetric positive–definite matrix P, satisfying

$$\dot{V}(x) = \frac{\partial V}{\partial x}(A + BK)x \leq 0 \tag{18}$$

in the case of stability or

$$\dot{V}(x) = \frac{\partial V}{\partial x}(A + BK)x \leq -\alpha V(x) \tag{19}$$

with $\alpha > 0$ for the case of asymptotically stability. It will also be useful to denote by V_q^μ the set:

$$\{x \in \mathbb{R}^n | V(q - x) \leq \mu\}.$$

The linear feedback $u = Kx$ and the corresponding Lyapunov function V can now be used to construct surjective simulation relations.

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

1) stabilizing feedback $u = Kx$ and corresponding Lyapunov function V;
2) symbolic subsystem $T = (Q, Q^0, \longrightarrow, O_\Sigma, H)$ of T_Σ;
3) bounded set $X \subset Q_\Sigma$ containing Q;

there exists a real number $\mu \in \mathbb{R}$ such that

1) $X \subseteq L := \bigcup_{q \in Q} V_q^\mu$;
2) $R \subseteq Q \times L$, defined by $(q, l) \in R$ when $V(l - q) \leq \mu$, is a simulation relation from T to T_L satisfying $\pi_2(R) = L$;
3) $T \parallel T_L \cong T$;

where $T_L = (L, \cup_{q \in Q^0} V_q^\mu, \longrightarrow_\Sigma \cap (L \times L), O_\Sigma, H_L)$ with $q \in H_L(l)$ when $V(l - q) \leq \mu$.

Proof: Since X is bounded and V is radially unbounded there exists a real number μ such that $L = \bigcup_{q \in Q} V_q^\mu$ covers K, that is

$$X \subseteq L \tag{20}$$

Furthermore, $\pi_2(R) = L$ holds by construction of R and so do requirements 1) and 2) in Definition 4.1. We now prove that requirement 3) also holds. Assume that $q \longrightarrow q'$ in T and recall that this implies the existence of an input trajectory $\mathtt{u} : [0, \tau] \rightarrow \mathbb{R}^m$ whose corresponding state trajectory $\mathtt{q} : [0, \tau] \rightarrow \mathbb{R}^n$ satisfies $\mathtt{q}(0) = q$ and $\mathtt{q}(\tau) = q'$. Consider now any point $l \in L$ such that $(q, l) \in R$. By definition of R it follows that $V(l - q) = V(q - l) \leq \mu$. If $\mathtt{l} : [0, \tau] \rightarrow \mathbb{R}^n$ is the trajectory satisfying

$$\frac{d}{dt}\mathtt{l}(t) = A\mathtt{l}(t) + B\left(\mathtt{u}(t) - K\left(\mathtt{q}(t) - \mathtt{l}(t)\right)\right)$$

with $\mathtt{l}(0) = l$ then we claim that $\mathtt{l}(\tau) = l'$ satisfies $V(q' - l') \leq \mu$. This claim is proved by showing that $V(z)$, with $z = q - l$, is a Lyapunov function for the linear dynamical system defined by

$$\dot{z}=Aq+Bu-Al-B\left(u-K(q-l)\right)=Az+BKz. \tag{21}$$

However, this follows at once from (18).

We now show, with the help of R, that $T \parallel T_L \cong T$ holds. Consider the relation $S \subseteq (Q\times L)\times Q$ defined by $((q,l),p) \in S$ if $q = p$ and $V(q-l) \leq \mu$. We claim that S defines a bisimulation relation between T and $T \parallel T_L$. Let us show first that S is a simulation relation from $T \parallel T_L$ to T. Since requirements 1) and 2) in Definition 4.1 follow immediately from the construction of S we focus on 3). Let $(q,l) \longrightarrow (q',l')$ in $T \parallel T_L$ and note that by definition of S, $((q,l),p) \in S$ implies $q = p$. Since, by definition of parallel composition $(q,l) \longrightarrow (q',l')$ in $T \parallel T_L$ implies $q \longrightarrow q'$ in T, it follows the existence of $p' := q' \in Q$ satisfying $((q',l'),p') \in S$ and $p \longrightarrow p'$ in T. We now show that S^{-1} is a simulation relation from T to $T \parallel T_L$. Once again we focus on requirement 3) since requirements 1) and 2) follow trivially from the definition of S. Let $p \longrightarrow p'$ in T and consider any (q,l) such that $((q,l),p) \in S$. By definition of S, this implies $q = p$ and $V(p-l) \leq \mu$. Since V is Lyapunov function for dynamical system (21) we have $l \longrightarrow l'$ in T_Σ with $V(p'-l') \leq \mu$ and, thus, $l \longrightarrow l'$ in T_L. It then follows from the definition of parallel composition that $(q,l) \longrightarrow (q'=p',l')$ in $T \parallel T_L$ and, furthermore $((q',l'),p') \in S$ which concludes the proof. ■

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

$$\mathtt{u} - K(\mathtt{q} - \mathtt{l}) \tag{22}$$

to be applied at points $l \in L$ satisfying $V(l-q) \leq \mu$. Input trajectory (22) controls points l that are close to q (points satisfying $V(l-q) \leq \mu$) to points l' that are close to q' (points satisfying $V(l'-q') \leq \mu$).

We now revisit the double integrator example with the purpose of illustrating Theorem 6.1. Let T be the transition system displayed on Fig. 3 and let X 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 Lyapunov function associated with the feedback $u = -x_1 - x_2$ we can take $\mu = 1$ which results in each set $V_q^\mu = \{x \in \mathbb{R}^2 | V(x-q) \leq \mu = 1\}$ being a closed ball of radius 1 centered at q. In Fig. 4, we can see how this choice results in a covering for X.

By analyzing Fig. 4, 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 $l_1, l_2 \in L$ if $V(l_1 - q) \leq \mu$ and $V(l_2 - q) \leq \mu$ 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 subsystem T which would lead to a lower value for μ resulting in less uncertainty in the position of the state. Alternatively, we can use feedback to reduce the uncertainty associated with the location of the state as discussed in the next section.

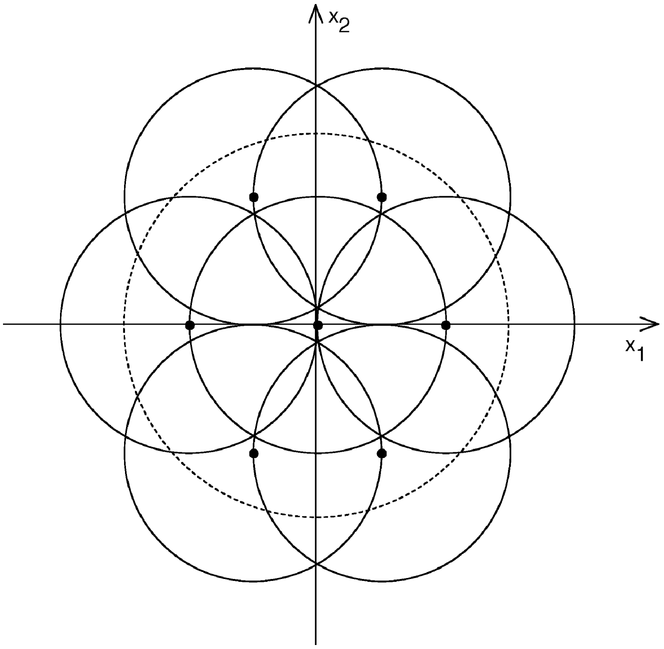

Fig. 4. Covering of K by L. The set K is delimited by the dashed circle while the sets $V_q^\mu = \{x \in \mathbb{R}^2 | V(x-q) \leq \mu = 1\}$ are delimited by the solid circles.

VII. From Symbolic Controllers to Hybrid 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.

Recall that asymptotic stabilizability of Σ implies the existence of a linear feedback $u = Kx$ and of a Lyapunov function V satisfying (19). Integrating (19), we obtain

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

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 of duration $\tau > 0$ 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 subsystem $T = (Q, Q^0, \longrightarrow, O_\Sigma, H)$ of T_Σ, $T_{\mathbb{N}_0}$ denotes the transition system defined by $(Q\times\mathbb{N}_0, Q^0\times\mathbb{N}_0, \longrightarrow_{\mathbb{N}_0}, O_\Sigma\times\mathbb{N}_0, H_{\mathbb{N}_0})$ where $(q,n) \longrightarrow_{\mathbb{N}_0}$

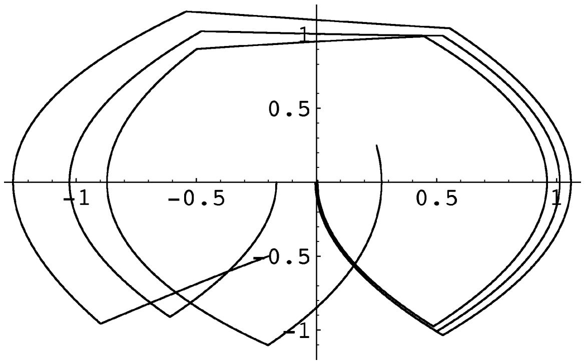

Fig. 5. Initial conditions (1/4,1/4), $(-1/5, -1/2)$ and $(-1/6,0)$ corresponding to observation $\{((0,0),0)\}$ and controlled to observation $\{((0,0),7)\}$. Control is enforced by the sequence of input trajectories $\mathtt{u}_-\mathtt{u}_+\mathtt{u}_+\mathtt{u}_0\mathtt{u}_-\mathtt{u}_-\mathtt{u}_+$.

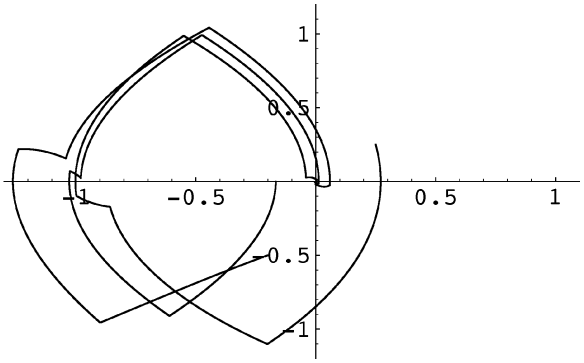

Fig. 6. Initial conditions (1/4,1/4), $(-1/5, -1/2)$ and $(-1/6,0)$ corresponding to observation $\{((0,0),0)\}$ and controlled to observation $\{((0,0),7)\}$. Control is enforced by the sequence of input trajectories $\mathtt{u}_-\mathtt{u}_+\mathtt{u}_0\mathtt{u}_+\mathtt{u}_-\mathtt{u}_0\mathtt{u}_0$.

(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 q from some state in Q^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

1) asymptotically stabilizing feedback $u = Kx$ and corresponding Lyapunov function V;
2) symbolic subsystem $T = (Q, Q^0, \longrightarrow, O_\Sigma, H)$ of T_Σ;
3) bounded set $X \subset Q_\Sigma$ containing Q;

there exists a real number $\mu \in \mathbb{R}$ such that

1) $X \subseteq L := \bigcup_{q \in Q} V_q^\mu$;
2) $R \subseteq (Q \times \mathbb{N}_0) \times L$, defined by $((q, n), l) \in R$ when $V(l - q) \leq \mu\sigma^n$, is a simulation relation from $T_{\mathbb{N}_0}$ to T_L satisfying $\pi_2(R) = L$;
3) $T_{\mathbb{N}_0} \parallel T_L \cong T_{\mathbb{N}_0}$,

where $T_L = (L, \cup_{q \in Q^0} V_q^\mu, \longrightarrow_\Sigma \cap (L \times L), O_\Sigma \times \mathbb{N}_0, H_L)$ with $(q, n) \in H_L(l)$ when $V(l - q) \leq \mu\sigma^n$.

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

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

satisfying $\dot{V} = -V$ for the linear feedback $u = -x_1 - x_2$. As symbolic subsystem T we consider again the transition system represented on Fig. 3. The sequence of inputs $\mathtt{u}_-\mathtt{u}_+\mathtt{u}_+\mathtt{u}_0\mathtt{u}_-\mathtt{u}_-\mathtt{u}_+$ guarantees that any state l contained in the set $V_{(0,0)}^1$ and corresponding to observation $\{((0,0),0)\} \in H_L(l)$ will be controlled to some point l' in the set $V_{(0,0)}^{\sigma^7} \approx V_{(0,0)}^{0.0009}$ and corresponding to observation $\{((0,0),7)\} \in H_L(l')$. The result of this sequence of inputs can be seen on Fig. 5 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 Fig. 6 for the same initial conditions. These examples illustrate how the loss of resolution incurred in the passage from T_Σ to symbolic subsystem T can be compensated by feedback.

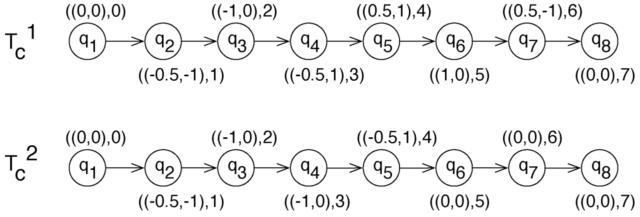

Fig. 7. Symbolic controllers for T.

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. Adopting such techniques would lead to the symbolic controllers displayed in Fig. 7 where T_c^1 enforces the sequence of inputs $\mathtt{u}_-\mathtt{u}_+\mathtt{u}_+\mathtt{u}_0\mathtt{u}_-\mathtt{u}_-\mathtt{u}_+$ and T_c^2 the sequence $\mathtt{u}_-\mathtt{u}_+\mathtt{u}_0\mathtt{u}_+\mathtt{u}_-\mathtt{u}_0\mathtt{u}_0$.

In practical applications, it is enough to consider a truncated version of $T_{\mathbb{N}_0}$ where $\mathbb{N}_0$ is replaced by $\{0, 1, 2, \ldots, r\}$ since other sources of disturbances prevent the uncertainty in the location of the state to be reduced below σ^r for some $r \in \mathbb{N}$. This truncated version of $T_{\mathbb{N}_0}$ is symbolic and thus permits again the use of supervisory control techniques for control design. A different alternative is to encode requirements on the state uncertainty not on the states but rather at the level of the specification. This allows one to use a symbolic subsystem of the kind described in Section IV to synthesize supervisors enforcing also quantitative specifications regarding the state uncertainty.

B. Controllers as Hybrid Systems

Symbolic controllers T_c^1 and T_c^2 are 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 an abstract transition $q \longrightarrow q'$ the feedback control law implementing it on Σ. For example, since transition

$q_1 \longrightarrow q_2$ in T_c^2 corresponds to input $\mathtt{u}_-$, we can enrich or label the transition $q_1 \longrightarrow q_2$ with the feedback law

$$(l, t) \mapsto \mathtt{u}_-(t) - K\left(\mathtt{q}(t) - l\right)$$

where $\mathtt{q}$ is the trajectory corresponding to input $\mathtt{u}_-$ and satisfying $\mathtt{q}(0) = (0, 0)$. By repeating this process for every transition 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 $H_c \parallel \Sigma$.

It is important to emphasize that the process of transforming symbolic controllers into hybrid controllers results in controllers that are, in general, of a true hybrid nature in the sense that they cannot be described by a continuous-state feedback map. This is already the case for controllers T_c^1 and T_c^2 albeit their simplicity. If one considers states q_1 and q_6 of T_c^2 we see that they have the same continuous state observations (0,0), even though the actions to be performed are different. Discrete states are therefore essential to determine the input signals and we cannot reduce the controller to a mapping from states in $L \subseteq \mathbb{R}^n$ to inputs in $\mathbb{R}^m$.

VIII. Nonlinear Control Systems

Even though we have focused on linear systems throughout the paper, Theorems 6.1 and 7.2 carry over to nonlinear control systems. This generalization relies on the observation that the essential ingredient in constructing surjective simulation relations is a certain stability property on the trajectories of control systems. This property, termed incremental stability, follows from stability in the linear case but it has to be separately assumed in the nonlinear case. Following [3], we say that a nonlinear control system $\dot{x} = f(x, u)$ is incrementally input-to-state stable if there exists a $\mathcal{KL}$ function β and a $\mathcal{K}_\infty$ function γ such that for any $t \geq 0$, for any $x_1, x_2 \in \mathbb{R}^n$ and for any input trajectories $\mathtt{u}_1$, $\mathtt{u}_2$ we have

$$\left|\mathtt{x}_{(x_1,\mathtt{u}_1)}(t) - \mathtt{x}_{(x_2,\mathtt{u}_2)}(t)\right| \leq \beta\left(|x_1 - x_2|, t\right) + \gamma\left(\|\mathtt{u}_1 - \mathtt{u}_2\|_\infty\right) \tag{23}$$

where $\mathtt{x}_{(x_1,\mathtt{u}_1)}$ and $\mathtt{x}_{(x_2,\mathtt{u}_2)}$ denote the state trajectories corresponding to initial conditions x_1 and x_2, and input trajectories $\mathtt{u}_1$ and $\mathtt{u}_2$, respectively. From (23), we see that by choosing $\mathtt{u}_1 = \mathtt{u}_2$ we guarantee that the error between trajectories starting at different initial conditions, measured by $|\mathtt{x}_{(x_1,\mathtt{u}_1)}(t) - \mathtt{x}_{(x_2,\mathtt{u}_2)}(t)|$, will decrease over time according to $\beta(|x_1 - x_2|, t)$. We can thus use the input trajectory $\mathtt{v}$ inducing a transition $q \longrightarrow q'$ in T to control any initial condition $x \in Q_\Sigma$ close to $q \in Q$ to a state $x' \in Q_\Sigma$ that is close to $q' \in Q$. Incremental input-to-state stability also admits a Lyapunov characterization, at least when the set of inputs is compact, which can then be used to define the surjective simulation relation from T to T_Σ. We refer the interested reader to [3] for more details on incremental stability and its Lyapunov characterizations. It is perhaps in this nonlinear setting that the proposed controller design methods acquire its full significance since it is not possible to explicitly construct exact discrete time models of nonlinear control systems. Nevertheless, we can still construct a symbolic subsystem by resorting to numerical simulation. Taking into account the numerical accuracy of the symbolic model it is possible to synthesize controllers efficiently while providing concrete guarantees of performance for nonlinear systems.

When a control system Σ is incrementally input-to-state stable, trajectories starting at close initial conditions will remain close provided that the same input is used. It is therefore possible to regard Σ as being approximately bisimilar to Σ in the sense of [17]. This observation highlights incremental input-to-state stability or approximate bisimulation as a common underlying tool that can be used for the control of continuous systems based on symbolic subsystems, as described in this paper, or for verification of continuous dynamical systems as in [17].

IX. Discussion

In this paper, we have shown that symbolic subsystems 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 subsystems result in hybrid systems models for controllers. Furthermore, since we can combine symbolic subsystems 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.

We have also discussed how the proposed methodology carries over to nonlinear control systems based on the notion of incremental input-to-state stability. This nonlinear generalization is quite important since, contrary to the linear case, it is not possible to obtain exact discrete-time models of nonlinear control systems. Nevertheless, symbolic subsystems can still be obtained by resorting to numerical simulation or by using feedback controllers enforcing known motion patterns.

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 subsystems. Even though the presented results are applicable to any symbolic subsystem, criteria to obtain complete (describing all the behavior of the original system up to a certain resolution) and yet small subsystems would be extremely important in practice.

REFERENCES

[1] R. Alur, C. Courcoubetis, N. Halbwachs, T. A. Henzinger, P. H. Ho, X. Nicollin, A. Olivero, J. Sifakis, and S. Yovine, "Hybrid automata: An algorithmic approach to specification and verification of hybrid systems," *Theoret. Comput. Sci.*, vol. 138, pp. 3–34, 1995.

[2] R. Alur and D. L. Dill, "A theory of timed automata," *Theoret. Comput. Sci.*, vol. 126, pp. 183–235, 1994.

[3] D. Angeli, "A lyapunov approach to incremental stability properties," *IEEE Trans. Autom. Control*, vol. 47, no. 3, pp. 410–421, MAr. 2002.

[4] P. J. Antsaklis, J. A. Stiver, and M. D. Lemmon, "Hybrid system modeling and autonomous control systems," in *Hybrid Systems*, R. L. Grossman, A. Nerode, A. P. Ravn, and H. Rischel, Eds. New York: Springer-Verlag, 1993, vol. 736, Lecture papers in Computer Sience, pp. 366–392.

[5] A. Arnold, A. Vincent, and I. Walukiewicz, "Games for synthesis of controllers with partial observation," *Theoret. Comput. Sci.*, vol. 28, no. 1, pp. 7–34, 2003.

[6] A. Bicchi, A. Marigo, and B. Piccoli, "On the rechability of quantized control systems," *IEEE Trans. Autom. Control*, vol. 47, no. 4, pp. 546–563, Apr. 2002.

[7] T. Brihaye, C. Michaux, C. Riviére, and C. Troestler, "On o-minimal hybrid systems," in *Hybrid Systems: Computation and Control*, R. Alur and G. J. Pappas, Eds. New York: Springer-Verlag, 2004, vol. 2993, Lecture papers in Computer Science, pp. 219–233.

[8] M. Broucke, "A geometric approach to bisimulation and verification of hybrid systems," in *Hybrid Systems: Computation and Control*, F. W. Vaandrager and J. H. van Schuppen, Eds. New York: Springer-Verlag, 1999, vol. 1569, Lecture papers in Computer Science, pp. 61–75.

[9] A. Chotinan and B. H. Krogh, "Verification of infinite state dynamical systems using approximate quotient transition systems," *IEEE Trans. Autom. Control*, vol. 46, no. 9, pp. 1401–1410, Sep. 2001.

[10] C. Cassandras and S. Lafortune*, Introduction to Discrete Event Systems*. Boston, MA: Kluwer, 1999.

[11] E. M. M. Clarke, D. Peled, and O. Grumberg*, Model Checking*. Cambridge, MA: MIT Press, 1999.

[12] P. E. Caines and Y. J. Wei, "Hierarchical hybrid control systems: A lattice theoretic formulation," *IEEE Trans. Autom. Control*, vol. 43, no. 4, pp. 501–508, Apr. 1998.

[13] L. de Alfaro, T. A. Henzinger, and R. Majumdar, "Symbolic algorithms for infinite-state games," in *Proc. CONCUR 01: Concurrency Theory, 12th Int. Conf.*, 2001, vol. 2154, Lecture papers in Computer Science.

[14] J. M. Davoren, T. Moor, and A. Nerode, "Hybrid control loops, A/D maps and dynamic specifications," in *Hybrid Systems: Computation and Control*, C. Tomlin and M. R. Greenstreet, Eds. New York: Springer-Verlag, 2002, vol. 2289, Lecture papers in Computer Sience.

[15] M. Egerstedt and R. W. Brockett, "Feedback can reduce the specification complexity of motor programs," *IEEE Trans. Autom. Control*, vol. 48, no. 2, pp. 213–223, Feb. 2003.

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

[17] A. Girard and G. J. Pappas, "Approximate bisimulations for nonlinear dynamical systems," in *Proc. 44th IEEE Conf. Decision and Control*, Seville, Spain, 2005, pp. 684–689.

[18] T. A. Henzinger, P. W. Kopke, A. Puri, and P. Varaiya, "What's decidable about hybrid automata?," *J. Comput. Syst. Sci.*, vol. 57, pp. 94–124, 1998.

[19] T. A. Henzinger and R. Majumdar, "Symbolic model checking for rectangular hybrid systems," in *TACAS 2000: Tools and Algorithms for the Construction and Analysis of Systems*, S. Graf, Ed. New York: Springer-Verlag, 2000, Lecture papers in Computer Science.

[20] T. A. Henzinger, R. Majumdar, and J.-F. Raskin, "A classification of symbolic transition systems," *ACM Trans. Comput. Logic*, pp. 1–31, 2003.

[21] E. Haghverdi, P. Tabuada, and G. J. Pappas, "Bisimulation relations for dynamical, control and hybrid systems," *Theoret. Comput. Sci.*, vol. 34, no. 2–3, pp. 387–392, 2005.

[22] J. E. Hopcroft and J. D. Ullman*, Introduction to Automata Theory, Languages and Computation*. Reading, MA: Addison-Wesley, 1979.

[23] D. Hristu-Varsakelis, M. Egerstedt, and P. S. Krishnaprasad, "On the structural complexity of the motion description language mdle," in *Proc. 42nd IEEE Conf. Decision and Control*, Maui, HI, 2003, pp. 3360–3365.

[24] X. Koutsoukos and P. Antsaklis, "Safety and reachability of piecewise linear hybrid dynamical systems based on discrete abstractions," *J. Discrete Event Dyna. Syst.: Theory Appl.*, vol. 13, no. 3, pp. 203–243, 2003.

[25] R. Kumar and V. K. Garg*, Modeling and Control of Logical Discrete Event Systems*. Norwell, MA: Kluwer, 1995.

[26] O. Kupferman, P. Madhusudan, P. S. Thiagarajan, and M. Y. Vardi, "Open systems in reactive environments: control and synthesis," *Proc. 11th Int. Conf. Concurency Theory* vol. 1877, Lecture papers in Computer Science, pp. 92–107. New York, Springer-Verlag, 2000.

[27] T. J. Koo, G. J. Pappas, and S. Sastry, "Mode switching synthesis for reachability specifications," in *Hybrid Systems: Computation and Control*, M. D. Di Benedetto and A. Sangiovanni-Vincentelli, Eds. New York: Springer-Verlag, 2001, vol. 2034, Lecture papers in Computer Science, pp. 333–346.

[28] G. Lafferriere, G. J. Pappas, and S. Sastry, "Ominimal hybrid systems," *Math. Control, Signals Syst.*, vol. 13, no. 1, pp. 1–21, March 2000.

[29] J. Lunze, "Qualitative modelling of linear dynamical systems with quantized state measurements," *Automatica*, vol. 30, pp. 417–431, 1994.

[30] J. O. Moody and P. J. Antsaklis*, Supervisory Control of Discrete Event Systems Using Petri Nets*. Norwell, MA: Kluwer, 1998.

[31] R. Milner*, Communication and Concurrency*. Englewood Cliffs, NJ: Prentice-Hall, 1989.

[32] T. Moor, J. Raisch, and S. D. O'Young, "Discrete supervisory control of hybrid systems based on l-complete approximations," *J. Discrete Event Dyna. Syst.*, vol. 12, no. 1, pp. 83–107, 2002.

[33] P. Madhusudan and P. S. Thiagarajan, "Branching time controllers for discrete event systems," *Theoret. Comput. Sci.*, vol. 274, pp. 117–149, Mar. 2002.

[34] A. Nerode and W. Kohn, "Models for hybrid systems: automata, topologies, controllability, observability," in *Hybrid Systems*, R. L. Grossman, A. Nerode, A. P. Ravn, and H. Rischel, Eds. New York: Springer-Verlag, 1993, vol. 736, Lecture papers in Computer Sience, pp. 317–356.

[35] G. J. Pappas, "Bisimilar linear systems," *Automatica*, vol. 39, no. 12, pp. 2035–2047, Dec. 2003.

[36] D. M. R. Park, Concurrency and Automata on Infinite Sequences vol. 104, Lecture papers in Computer Science, pp. 167–183, 1981.

[37] S. Pancanti, L. Leonardi, L. Pallottino, and A. Bicchi, "Optimal control of quantized linear systems," in *Hybrid Systems: Computation and Control*, C. Tomlin and M. R. Greenstreet, Eds. New York: Springer-Verlag, 2002, Lecture papers in Computer Sience, pp. 351–363.

[38] A. Puri and P. Varaiya, "Decidability of hybrid systems with rectangular inclusions," in *Computer Aided Verification*. Stanford, CA: Springer-Verlag, 1994, pp. 95–104.

[39] G. Pola, A. J. van der Schaft, and M. D. di Benedetto, "Bisimulation theory for switching linear systems," in *Proc. 43rd IEEE Conf. Decision and Control*, Paradise Island, Bahamas, 2004, pp. 1406–1411.

[40] J. Raisch and S. D. O'Young, "Discrete approximations and supervisory control of continuous systems," *IEEE Trans. Autom. Control*, vol. 43, no. 4, pp. 569–573, Apr. 1998.

[41] J. A. Stiver, X. D. Koutsoukos, and P. J. Antsaklis, "An invariant based approach to the design of hybrid control systems," *Int. J. Robust Nonlinear Control*, vol. 11, no. 5, pp. 453–478, 2001.

[42] O. Stursberg, S. Kowalewski, I. Hoffman, and J. Preussig, "Comparing timed and hybrid automata as approximations of continuous systems," in *Hybrid Systems IV*, P. J. Antsaklis, W. Kohn, A. Nerode, and S. Sastry, Eds. New York: Springer-Verlag, 1997, vol. 1273, Lecture papers in Computer Science, pp. 361–377.

[43] P. Tabuada, "Flatness and finite bisimulations in discrete time," in *Proc. 16th Int. Symp. Mathematical Theory of Networks and Systems*, Leuven, Belgium, Jul. 2004.

[44] ——, "Open maps, alternating simulations and controller synthesis," in P. Gardner and N. Yoshida, Eds., *Proc. 15th Int. Conf. Concurrence Theory* vol. 3170, Lecture papers in Computer Science, pp. 466–480 London, UK, Springer, 2004.

[45] W. Thomas, "On the synthesis of strategies in infinite games," *Proc. 12th Annu. Symposium on Theoretical Aspects of Computer Science* vol. 900, Lecture papers in Computer Science, pp. 1–13. New York, Springer-Verlag, 1995.

[46] A. Tiwari and G. Khanna, "Series of abstractions for hybrid automata," in *Hybrid Systems: Computation and Control*, C. Tomlin and M. R. Greenstreet, Eds. New York: Springer-Verlag, 2002, vol. 2289, Lecture papers in Computer Science, pp. 465–478.

[47] D. C. Tarraf, A. Megretski, and M. A. Dahleh, "Finite automata approximations with error bounds for systems with quantized actuation and measurement: a case study," in *Proc. 43rd IEEE Conf. Decision and Control*, Paradise Island, Bahamas, 2004, pp. 1436–1441.

[48] P. Tabuada and G. J. Pappas, "Linear time logic control of discrete-time linear systems," *IEEE Trans. Autom. Control*, 2006, to be published.

[49] ——, "Finite bisimulations of controllable linear systems," in *Proc. 42nd IEEE Conf. Decision and Control*, Honolulu, Hawaii, 2003.

[50] ——, "From discrete specifications to hybrid control," in *Proc. 42nd IEEE Conf. Decision and Control*, Honolulu, Hawaii, 2003.

[51] ——, "Model checking LTL over controllable linear systems is decidable," in *Hybrid Systems: Computation and Control*O. Maler and A. Pnueli, Eds. New York: Springer-Verlag, 2003, vol. 2623, Lecture papers in Computer Sience.

[52] ——, "Bisimilar control affine systems," *Syst. Control Lett.*, vol. 52, no. 1, pp. 49–58, 2004.

[53] A. J. van der Schaft, "Bisimulation of dynamical systems," in *Hybrid Systems: Computation and Control*, R. Alur and G. J. Pappas, Eds. New York: Springer-Verlag, 2004, vol. 2993, Lecture papers in Computer Science, pp. 555–569.

[54] R. Vidal, S. Schaffert, O. Shakernia, J. Lygeros, and S. Sastry, "Decidable and semi-decidable controller synthesis for classes of discrete time hybrid systems," in *Proc. 40th IEEE Con. Decision and Control*, Orlando, FL, Dec. 2001, pp. 1243–1248.

Paulo Tabuada (S'00–M'02) was born in Lisbon, Portugal, one year after the Carnation Revolution. He received the "Licenciatura" degree in aerospace engineering from Instituto Superior Técnico, Lisbon, Portugal, in 1998, and the Ph.D. degree in electrical and computer engineering from the Institute for Systems and Robotics, a private research institute associated with Instituto Superior Técnico, in 2002.

Between January 2002 and July 2003, he was a Postdoctoral Researcher at the University of Pennsylvania, Philadelphia. He is currently an Assistant Professor in the Department of Electrical Engineering at the University of Notre Dame, Notre Dame, IN. He co-edited the volume *Networked Embedded Sensing and Control* published in Springer's Lecture Notes in Control and Information Sciences series. His research interests include modeling, analysis and control of real-time, embedded, networked and distributed systems, geometric control theory, and mathematical systems theory.

Dr. Tabuada was the recipient of the Francisco de Holanda Prize in 1998 for the best research project with an artistic or aesthetic component, awarded by the Portuguese Science Foundation. He was a finalist for the Best Student Paper Award at both the 2001 American Control Conference and the 2001 IEEE Conference on Decision and Control, and he was the recipient of a National Science Foundation (NSF) CAREER Award in 2005.