Model checking LTL over controllable linear systems is decidable

Paulo Tabuada and George J. Pappas

Department of Electrical and Systems Engineering
University of Pennsylvania
Philadelphia, PA 19104
{tabuadap,pappasg}@seas.upenn.edu

Abstract. The use of algorithmic verification and synthesis tools for hybrid systems is currently limited to systems exhibiting simple continuous dynamics such as timed automata or rectangular hybrid systems. In this paper we enlarge the class of systems amenable to algorithmic analysis and synthesis by showing decidability of model checking Linear Temporal Logic (LTL) formulas over discrete time, controllable, linear systems. This result follows from the construction of a language equivalent, finite abstraction of a control system based on a set of finite observations which correspond to the atomic propositions appearing in a given LTL formula. Furthermore, the size of this abstraction is shown to be polynomial in the dimension of the control system and the number of observations. These results open the doors for verification and synthesis of continuous and hybrid control systems from LTL specifications.

1 Introduction

Hybrid systems are a powerful modeling paradigm for large-scale, complex systems where interaction between discrete and continuous components occurs. Due to highly nontrivial interaction between discrete and continuous components, one would like to have automatic tools for the analysis and synthesis of such systems. Unfortunately, existing tools only address classes of systems with very simple continuous dynamics, such as timed automata [2], multi-rate automata [1] or rectangular hybrid systems [25, 13].

The main contribution of this paper is to show that algorithmic approaches are also possible for larger classes of continuous dynamics. In particular, we show that given a specification described by a Linear Temporal Logic (LTL) formula φ, it is possible to construct a language equivalent finite abstraction of a discrete time, controllable, linear system based on the formula φ. This construction immediately implies that model checking a specification formula φ over a linear system is decidable as it can be performed on the finite abstraction. Furthermore, these results also open the doors for automatic controller synthesis of linear systems from LTL specifications. Combining automatic synthesis for continuous systems with existing tools for discrete systems [16, 18, 19, 11] will eventually lead to automatic synthesis for hybrid systems.

Automatic analysis and synthesis of hybrid systems began with the seminal work of Alur and Dill on timed automata [2]. Subsequent extensions lead to multi-rate automata [1] and rectangular hybrid automata [25, 13] which lies on the decidability boundary [14]. Different classes of dynamics for which finite abstractions exist were introduced in [17] by combining tools from logic and linear dynamical systems. See also [3] for a survey of these methods. Nonlinear dynamics were considered in [6], and bisimulation based on foliations transverse to the nonlinear flow were introduced. A different kind of dynamics, simple planar differential inclusions, was considered in [4] where it was shown that qualitative analysis of system trajectories is decidable by making use of unique topological properties of the plane.

Our approach differs from all the above in that we consider *control* systems instead of dynamical systems. It is the use of control that allows to modify[1] the trajectories of the system into a form which admits a finite representation. Hence, our results are closer to synthesis than verification problems. Another important difference is that we consider continuous control systems in discrete time as opposed to continuous time. Synthesis for hybrid systems using logic has already been considered in [22], however the logic is not used to model the specifications but rather to motivate the development of the synthesis procedures as well as to prove several facts regarding the proposed algorithms. Other synthesis techniques include supervisory control based on approximate finite abstractions [9], invariants for the continuous dynamics [28], convexity properties of affine systems [12] and mixed integer linear programming [5].

The construction of the finite abstraction of a given control system is performed in two steps, each exploiting in a fundamental way the ability to shape the system trajectories by appropriate choices of control. First, we show that (by the use of control) we can transform any discrete time, controllable, linear system into a canonical form, which induces a quotient system on a denumerable state space, namely $\mathbb{Z}^n$. If the observations are also compatible with the quotient, we have a bisimilar quotient. This first step depends crucially on the controllability of the original system. The second step further abstracts the quotient system into a finite, language equivalent system based on a finite set of observations. The finiteness of this abstraction is again a consequence of the controllability properties of the original control system.

The outline of this paper is the following. We revisit transition systems in Section 2 and discuss the relation between linear control systems and transition systems in Section 3. Section 4 shows how a denumerable (but not finite, however) bisimulation of a control system can be obtained. This bisimulation can be further reduced to a language equivalent finite abstraction. This is described in

[1] We note that the results in [4] can also be given a control interpretation. The authors prove a normal form for the edge crossing trajectories (edge signatures) by showing that for any system trajectory, there is another with the same qualitative properties but with a special structure. This new trajectory can then be though as the result of applying a suitable control law. However, our results are not restricted to planar systems.

Section 5, where the main contribution of the paper is also presented. For space reasons no proofs are presented and the interested reader is referred to [29].

2 Transition systems

Given a set S (finite or not), we denote by S^ω the set of all infinite strings formed by elements of S. An element of S^ω is of the form $\alpha = \alpha_1\alpha_2\ldots$ and we identify it with the map $\alpha : \mathbb{N} \to S$ by setting $\alpha(1) = \alpha_1$, $\alpha(2) = \alpha_2$, etc. The main object used in this work are transition systems:

Definition 1. *A transition system with observations is a tuple $T = (Q, L, \longrightarrow, O, h)$, where:*

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

We say that T is finite when Q, L, O are finite, and infinite otherwise. We will usually denote by $q \xrightarrow{l} q'$ a triple (q, l, q') belonging to $\longrightarrow$. As we will only use transition systems with observations, we shall refer to them simply as transition systems. Transition systems define subsets of O^ω, also called languages:

Definition 2. *Given a transition system $T = (Q, L, \longrightarrow, O, h)$, we say that $\gamma \in O^\omega$ is an observed string of T if there exists a pair of infinite strings $(\alpha, \beta) \in Q^\omega \times L^\omega$ such that $\alpha(i) \xrightarrow{\beta(i)} \alpha(i+1)$ and $\gamma(i) = h(\alpha(i))$ for every $i \in \mathbb{N}$. The collection of all observed strings is denoted by $\mathbf{L}(T)$ and defines the language of the transition system.*

Given transition systems T_1 and T_2 with the same observation space, we say that T_1 is language equivalent to T_2 when $\mathbf{L}(T_1) = \mathbf{L}(T_2)$. For later use we introduce also the Pre operator. Given a state $q \in Q$, we denote by $\mathrm{Pre}(q)$ the set of states in Q that can reach q in one step, that is:

$$\mathrm{Pre}(q) = \{q' \in Q \ : \ q' \xrightarrow{l} q \text{ for some } l \in L\}$$

We extend Pre to sets $Q' \subseteq Q$ in the usual way:

$$\mathrm{Pre}(Q') = \bigcup_{q' \in Q'} \mathrm{Pre}(q')$$

Finally, we recursively define $\mathrm{Pre}^i(Q')$ by:

$$\begin{aligned} \mathrm{Pre}^1(Q') &= \mathrm{Pre}(Q') \\ \mathrm{Pre}^i(Q') &= \mathrm{Pre}(\mathrm{Pre}^{i-1}(Q')) \end{aligned} \tag{1}$$

2.1 Transition systems as LTL models

Linear temporal logic (LTL) provides a succinct and formal way of representing temporal properties of dynamical and control systems. In this section we briefly describe the syntax and semantics of LTL.

Specification formulas are built from atomic propositions belonging to a finite set $\mathcal{P}$ and are recursively defined by:

- **true**, **false**, p and $\neg p$ are LTL formulas for all $p \in \mathcal{P}$;
- if φ_1 and φ_2 are LTL formulas, then $\varphi_1 \wedge \varphi_2$ and $\varphi_1 \vee \varphi_2$ are LTL formulas;
- if φ_1 and φ_2 are LTL formulas, then $\bigcirc\varphi_1$ and $\varphi_1 \mathcal{U} \varphi_2$ are LTL formulas.

The operator $\bigcirc$ is read as "next", with the meaning that the formula it precedes will be true in the next time step. The second operator $\mathcal{U}$ is read as "until" and the formula $\varphi_1 \mathcal{U} \varphi_2$ specifies that φ_1 must hold until φ_2 holds.

We shall interpret LTL formulas over observed sequences of transition systems. We consider that the set of observations O is defined by $\mathcal{P} \cup \{\tau\}$ for some element $\tau \notin \mathcal{P}$. This allows to use LTL formulas to specify the sequences of observations. The special symbol τ is used to represent observations not corresponding to any atomic proposition. LTL formulas are now interpreted over sequences of observations $\gamma : \mathbb{N} \to O$ as follows:

For any $p \in \mathcal{P}$, LTL formulas φ_1, φ_2, and $i \in \mathbb{N}$:

- $\gamma(i) \models p$ iff $p = \gamma(i)$,
- $\gamma(i) \models \varphi \wedge \varphi_2$ iff $\gamma(i) \models \varphi_1$ and $\gamma(i) \models \varphi_2$,
- $\gamma(i) \models \varphi \vee \varphi_2$ iff $\gamma(i) \models \varphi_1$ or $\gamma(i) \models \varphi_2$,
- $\gamma(i) \models \bigcirc\varphi_1$ iff $\gamma(i+1) \models \varphi_1$,
- $\gamma(i) \models \varphi_1 \mathcal{U} \varphi_2$ iff $\exists j \geq i$ such that for all k, $0 \leq k < j$ $\gamma(k) \models \varphi_1$ and $\gamma(j) \models \varphi_2$.

Finally we say that a sequence γ satisfies formula φ iff $\gamma(0) \models \varphi$.

In Section 3 we will associate a transition system with a given control system. Such association will enable the use of LTL as a specification mechanism for control systems through the use of the associated transition system.

2.2 Relationships between transition systems

We now review some relationships between transition systems. The interested reader may which to consult [8, 20] for a detailed discussion these and other related concepts. We start by introducing simulation and bisimulation relations [21, 24].

Definition 3 (Simulation and Bisimulation). *Let $T_1 = (Q_1, L_1, \longrightarrow_1, O, h_1)$ and $T_2 = (Q_2, L_2, \longrightarrow_2, O, h_2)$ be transition systems and $R \subseteq Q_1 \times Q_2$ a relation. Relation R is a simulation relation from T_1 to T_2 if $(q_1, q_2) \in R$ implies:*

- *if $q_1 \xrightarrow{l_1}_1 q_1'$, there exists $q_2' \in Q_2$, $l_2 \in L_2$ such that $q_2 \xrightarrow{l_2}_2 q_2'$ and $(q_1', q_2') \in R$,*
- *$h(q_1) = h(q_2)$.*

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

We note that in the previous definition we require the observation spaces of T_1 and T_2 to be the same. Furthermore, we only require T_2 to match transitions in T_1 with transitions having equal observations but not necessarily equal labels as we are only interested in the observed behavior.

We now review several important consequences of bisimulation relations:

Theorem 1. *Let $T_1 = (Q_1, L_1, \longrightarrow_1, O, h_1)$ and $T_2 = (Q_2, L_2, \longrightarrow_2, O, h_2)$ be transition systems and $R \subseteq Q_1 \times Q_2$ a bisimulation relation between T_1 and T_2. Then, they are language equivalent, that is, the following equality holds:*

$$\mathbf{L}(T_1) = \mathbf{L}(T_2)$$

Language equivalence is important as it ensures that properties expressible in LTL are preserved:

Theorem 2 ([27, 10]). *Let T_1 and T_2 be two language equivalent transition systems. Then, any LTL formula interpreted over observed sequences is satisfied by T_1 iff it is satisfied by T_2.*

Combining Corollary 1 with Theorem 2 we conclude that bisimilarity preserves properties expressible in LTL, however bisimilarity also preserves properties expressible in other temporal logics such as CTL, CTL* and μ-calculus [8].

3 Linear control systems as transition systems

Control systems can be seen as specifying infinite transition systems. In this section we will see how to extract such transition systems from linear control systems:

Definition 4. *A discrete time, linear control system $\Sigma = (A, B)$ is a controlled, discrete dynamical system defined by:*

$$x(t+1) = Ax(t) + Bu(t) \tag{2}$$

with $x(t) \in \mathbb{R}^n$, $A \in \mathbb{Q}^{n \times n}$, $B \in \mathbb{Q}^{n \times m}$, $u(t) \in \mathbb{R}^m$ and $t \in \mathbb{N}$.

The vector x describes the state of the system which can be influenced by the inputs u through the controlled dynamics (2). Although in control theory [26] it is customary to define A as an element of $\mathbb{R}^{n\times n}$ and B as an element of $\mathbb{R}^{n\times m}$, we consider only rational entries since any computer implementation of the results presented in this paper requires computable or decidable fields. The number n is the dimension of the control system.

In this paper we consider discrete time, controllable, linear systems. A control system is controllable if every point in the state space is reachable from any other point in the state space. Such property can be effectively decided through Kalman's rank condition which asserts that a control system $\Sigma = (A, B)$ is controllable iff the matrix $[B\ AB\ \ldots\ A^{n-2}B\ A^{n-1}B]$ is full row rank [26].

Given a linear control system $\Sigma = (A, B)$ we can construct a transition system T_Σ defined by:

$$T_\Sigma = (\mathbb{R}^n, \mathbb{R}^m, \longrightarrow, O, h)$$

with $\longrightarrow \subseteq \mathbb{R}^n \times \mathbb{R}^m \times \mathbb{R}^n$ given by $x \xrightarrow{u} x'$ iff $x' = Ax + Bu$.

The observation space O and observation map h are not defined by Σ. The nature of O and h will be determined in the next sections. We note that T_Σ is only one of the possible embeddings, described in [23], of control systems in the class of transition systems.

4 Controllable linear systems and their denumerable bisimulations

The main contribution of this paper is to provide a decidability result obtained through the computation of a finite abstraction of T_Σ for a given discrete time, controllable, linear system Σ and finite set of observations O. This finite abstraction will be constructed through several intermediate steps, the first one being the extraction of a bisimulation with denumerable state space from T_Σ. This will be achieved by transforming the control system into a normal form which immediately suggests how to obtain such a denumerable bisimulation.

We consider a discrete time, controllable, linear system $\Sigma = (A, B)$ and transform it to a special form called Brunovsky normal form [26]:

Definition 5 (Brunovsky normal form). *Consider a linear control system of dimension n with m inputs defined by the pair of matrices (A, B) and let $k = (k_1, k_2, \ldots, k_r)$ be a sequence of integers satisfying:*

$$k_1 \geq k_2 \geq \ldots \geq k_r \quad \text{and} \quad k_1 + k_2 + \ldots + k_r = n \tag{3}$$

We say that the pair (A, B) is in Brunovsky normal form if matrices A and B are of the following form:

$$A = \begin{bmatrix} A_{k_1} & 0 & \ldots & 0 \\ 0 & A_{k_2} & \ldots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \ldots & A_{k_r} \end{bmatrix} \quad B = \begin{bmatrix} b_{k_1} & 0 & \ldots & 0 & 0 \ldots 0 \\ 0 & b_{k_2} & \ldots & 0 & 0 \ldots 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \ddots \vdots \\ 0 & 0 & \ldots & b_{k_r} & 0 \ldots 0 \end{bmatrix} \tag{4}$$

where matrix A is partitioned in r^2 blocks while matrix B is partitioned in mr blocks. Each block A_{k_i} and b_{k_i} are of the form:

$$A_{k_i} = \begin{bmatrix} 0 & 1 & 0 & \dots & 0 \\ 0 & 0 & 1 & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \dots & 1 \\ \alpha_1^i & \alpha_2^i & \alpha_3^i & \dots & \alpha_{k_i}^i \end{bmatrix} \quad b_{k_i} = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ 1 \end{bmatrix} \tag{5}$$

where $\lambda^{k_i} - \alpha_{k_i}^i \lambda^{k_i - 1} - \dots - \alpha_2^i \lambda - \alpha_1^i$ is the characteristic polynomial[2] of A_{k_i}.

Any controllable linear system can be effectively transformed to Brunovsky normal form by a change of state coordinates as asserted by the following result:

Theorem 3 ([7, 15]). *For every controllable linear system, there exists a unique sequence of integers $k = (k_1, k_2, \dots, k_r)$ satisfying (3) and an invertible linear transformation $P \in \mathbb{Q}^{n \times n}$ such that the pair (PAP^{-1}, PB) is in Brunovsky normal form.*

In addition to state transformation P, we consider also a feedback transformation determining new inputs u' from inputs u and state x as follows:

$$\begin{bmatrix} u'_1 \\ u'_2 \\ \vdots \\ u'_m \end{bmatrix} = \begin{bmatrix} u_1 + \alpha_1^1 x_1 + \alpha_2^1 x_2 + \dots + \alpha_{k_1}^1 x_{k_1} \\ u_2 + \alpha_1^2 x_{k_1+1} + \alpha_2^2 x_{k_1+2} + \dots + \alpha_{k_2}^2 x_{k_1+k_2} \\ \vdots \\ u_m + \alpha_1^r x_{n-k_r+1} + \alpha_2^r x_{n-k_r+2} + \dots + \alpha_{k_r}^r x_n \end{bmatrix} \tag{6}$$

where we assumed $m = r$, which corresponds to the requirement that columns of matrix B are linearly independent and constitutes no loss of generality. Such assumption will be used throughout the paper. Combining state transformation P with feedback transformation (6) we obtain the invertible transformation $U : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}^n \times \mathbb{R}^m$ defined by:

$$\begin{bmatrix} x' \\ u' \end{bmatrix} = U \begin{bmatrix} x \\ u \end{bmatrix} = \begin{bmatrix} P & \mathbf{0}_{n \times m} \\ V & \mathbf{I}_{m \times m} \end{bmatrix} \begin{bmatrix} x \\ u \end{bmatrix} \tag{7}$$

where $\mathbf{0}_{n \times m}$ is a $n \times m$ matrix of zeros, $\mathbf{I}_{m \times m}$ is the $m \times m$ identity matrix and V is a matrix where each row v_i is of the form:

$$v_i = [\mathbf{0}_{1 \times (k_1 + k_2 + \dots + k_{i-1})} \ \alpha_1^i \ \alpha_2^i \ \dots \ \alpha_{k_i}^i \ \mathbf{0}_{1 \times (k_{i+1} + \dots + k_r)}] \tag{8}$$

Note that, as it was the case for P, transformation U is also invertible and has rational entries. The control system $\Sigma' = (A', B')$ obtained from Σ by

[2] The characteristic polynomial of a square matrix A is given by $\det(A - \lambda I)$, where I is the identity matrix. Note that since A has rational entries, the coefficients α_j^i are also rational.

transformation U has state space $\mathbb{R}^n$ with coordinates $x' = Px$, input space $\mathbb{R}^m$ with coordinates u' defined by (6) and is of the form:

$$x'(t+1) = A'x'(t) + B'u'(t) \tag{9}$$

where the pair (A', B') is defined by block matrices with the format (4), but where each block is of the form:

$$A'_{k_i} = \begin{bmatrix} 0\,1\,0\,\ldots\,0 \\ 0\,0\,1\,\ldots\,0 \\ \vdots\,\vdots\,\vdots\,\ddots\,\vdots \\ 0\,0\,0\,\ldots\,1 \\ 0\,0\,0\,\ldots\,0 \end{bmatrix} \qquad b'_{k_i} = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ 1 \end{bmatrix} \tag{10}$$

We now see that the effect of feedback (6) was to cancel out the last line of the matrices A'_{k_i}. This has the effect of rendering these matrices nilpotent and can only be achieved by means of control. This fact marks the departure of the presented results from existing techniques for dynamical systems, where no inputs are available to modify the system dynamics. We shall refer to the form (9), (10) as the *shift register* form. This name is justified by the representation of control system (9) presented in Figure 1. In the control literature, the use of feedback transformations of the form (6) is usually referred to as *deadbeat control*, see for example, [26], Section 1.3, for a quick introduction to digital control.

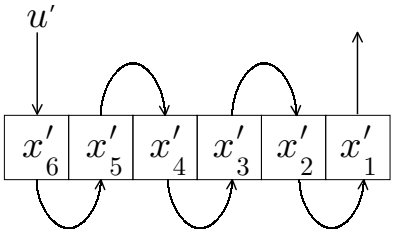

Fig. 1. Representation of a discrete time system in shift register form as a shift register for $n = 6$ and $m = 1$.

Control systems Σ and Σ' are related by the invertible transformation U which has the following immediate consequence for the transition systems they define:

Proposition 1. *Let $T_\Sigma = (\mathbb{R}^n, \mathbb{R}^m, \longrightarrow_\Sigma, O, h_\Sigma)$ be the transition system defined by a discrete time, controllable, linear system Σ of dimension n with m inputs, let Σ' be the shift register form of Σ and $T_{\Sigma'} = (\mathbb{R}^n, \mathbb{R}^m, \longrightarrow_{\Sigma'}, O, h_{\Sigma'})$ the corresponding transition system. Given any observation map $h_{\Sigma'} : \mathbb{R}^n \to O$*

for $T_{\Sigma'}$, the choice of observation map $h_\Sigma = h_{\Sigma'} \circ P$ for T_Σ renders T_Σ bisimilar to $T_{\Sigma'}$ with respect to the relation $R \subseteq \mathbb{R}^n \times \mathbb{R}^n$ defined by:

$$R = \{(x, x') \in \mathbb{R}^n \times \mathbb{R}^n \ : \ Px = x'\}$$

Proposition 1 is not surprising since Σ and Σ' are isomorphic via the invertible linear transformation U. However, this result paves the way to the introduction of a new transition system, bisimilar to $T_{\Sigma'}$, but with state space $\mathbb{Z}^n$. The new state space $\mathbb{Z}^n$ is obtained from $\mathbb{R}^n$ through the following quantization:

$$\mathcal{B} = \bigcup_{z \in \mathbb{Z}^n} B^\delta(z)$$

where the blocks $B^\delta(z)$ are defined for any $\delta \in \mathbb{Q}^+$ by:

$$B^\delta(z) = \left\{x \in \mathbb{R}^n \ : \ \delta z_i - \delta/2 < x'_i \leq \delta z_i + \delta/2 \qquad i = 1, 2, \ldots, n\right\}$$

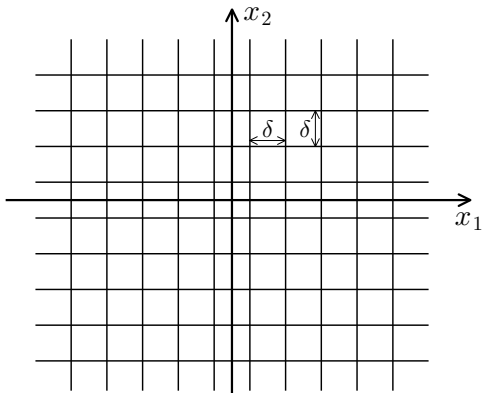

Fig. 2. Quantization $\mathcal{B}$ of $\mathbb{R}^2$, where each square represents a block $B^\delta(z)$.

The quantization induced by $\mathcal{B}$ defines the new state space by identifying each block $B^\delta(z) \subset \mathbb{R}^n$ with the point $z \in \mathbb{Z}^n$. This identification is given by the quantization map $h_\mathcal{B} : \mathbb{R}^n \to \mathbb{Z}^n$, where the ith component $h_{\mathcal{B}i}$ of $h_\mathcal{B}$ is defined by:

$$h_{\mathcal{B}i}(x') = \left\lfloor \frac{x'_i}{\delta} + \frac{1}{2}\mathrm{sgn}(\frac{x'_i}{\delta}) \right\rfloor \tag{11}$$

In the previous expression, $\lfloor a \rfloor$ denotes the floor map, returning the integer part of a while sgn denotes the sign map defined by:

$$\mathrm{sgn}(a) = \begin{cases} -1 \text{ if } a < 0 \\ \ \ 1 \text{ if } a \geq 0 \end{cases}$$

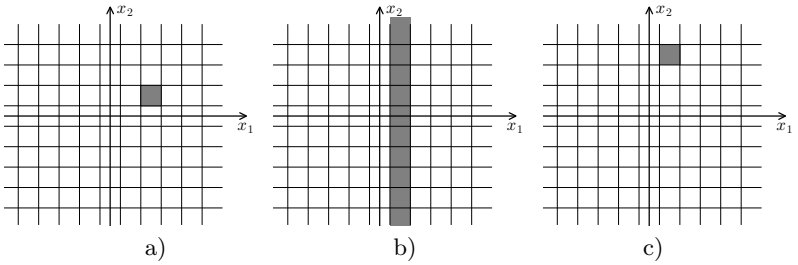

Fig. 3. Controlled evolution on the space of blocks. The left figure represents the initial state $B^\delta(2,1)$ at $t=0$. The middle figure illustrates the reachable set in one step from the initial condition, while the right figure represents the set reached for input $w=2$.

For the sake of exposition, we now assume that $n=2$ and $m=1$, as this will allow to represent graphically the next steps of the construction. For $n=2$, the quantization defined by $\mathcal{B}$ divides the plane in a square grid of resolution δ as displayed in Figure 2. We now introduce a control system on $\mathbb{Z}^2$, which we shall denote by Γ, such that T_Γ will be bisimilar to $T_{\Sigma'}$ under suitably defined[3] observations. The construction of Γ exploits the fact that, the shift register form of Σ' induces a well defined controlled dynamics between blocks of the quantization $\mathcal{B}$. This means that, under appropriate inputs, blocks will move into other blocks of the grid.

To illustrate the main idea of the construction process, consider all the points $(x_1', x_2') \in B^\delta(z_1, z_2)$ for some $(z_1, z_2) \in \mathbb{Z}^2$. These points (x_1', x_2') are represented in Figure 3a) for $(z_1, z_2) = (2, 1)$ and satisfy at $t = 0$:

$$\delta z_2 - \delta/2 < x_2'(0) \leq \delta z_2 + \delta/2$$

Since control system Σ' is in shift register form, $x_1'(1) = x_2'(0)$, that is:

$$\delta z_2 - \delta/2 < x_1'(1) \leq \delta z_2 - \delta/2 \tag{12}$$

This shows that, after one time step, all the points in block $B^\delta(z_1, z_2)$ will be contained in the vertical strip displayed in Figure 3b) and defined by (12). To have a well defined controlled dynamics between blocks, we only need to ensure that $x_2'(1)$ will, in fact, be contained in a interval of length δ. However, this can easily be achieved with the control law:

$$u'(t) = x_2'(t) + \delta y(t), \quad y \in \mathbb{Z} \tag{13}$$

The following computations show that $x_2'(1)$ will also lie on a interval of length δ:

$$\begin{aligned} \delta z_2 - \delta/2 \qquad &< x_2'(0) \leq \qquad \delta z_2 + \delta/2 \\ \delta z_2 + \delta y(0) - \delta/2 &< x_2'(0) + \delta y(0) \leq \delta z_2 + \delta y(0) + \delta/2 \\ \delta\big(z_2 + y(0)\big) - \delta/2 \qquad &< x_2'(1) \leq \qquad \delta\big(z_2 + y(0)\big) + \delta/2 \end{aligned}$$

[3] Recall that Proposition 1 holds for any observation map $h_{\Sigma'}$.

This simple control law ensures that the system evolves from $B^\delta(z_1(0), z_2(0))$ to $B^\delta(z_1(1), z_2(1))$ with $z_1(1) = z_2(0)$ and $z_2(1) = z_2(0) + y(0)$ therefore defining a control system on $\mathbb{Z}^2$ with input $y \in \mathbb{Z}$. We note that the controlled dynamics between blocks is not in shift register form since $z_2(t+1) = z_2(t) + y(t)$. As was the case for Σ, we introduce a feedback transformation to convert the controlled dynamics between blocks to shift register form. This transformation determines the new input w from the previous input y and state z_2 by:

$$w = y + z_2 \tag{14}$$

The previous argument used for the construction of the controlled dynamics between blocks can be extended for any n and m resulting in control system Γ defined by:

$$z(t+1) = A'z(t) + B'w(t)$$

for $z \in \mathbb{Z}^n$ and $w \in \mathbb{Z}^m$. Transition system $T_{\Sigma'}$ can now be related to transition system T_Γ defined by control system Γ:

Theorem 4. *Let $T_{\Sigma'} = (\mathbb{R}^n, \mathbb{R}^m, \longrightarrow_{\Sigma'}, O, h_{\Sigma'})$ be the transition system defined by control system Σ' and let $T_\Gamma = (\mathbb{Z}^n, \mathbb{Z}^m, \longrightarrow_\Gamma, O, h_\Gamma)$ be the transition system defined by control system Γ. Given any observation map $h_\Gamma : \mathbb{Z}^n \to O$ for T_Γ, the choice of observation map $h_{\Sigma'} = h_\Gamma \circ h_{\mathcal{B}}$ for $T_{\Sigma'}$ renders $T_{\Sigma'}$ bisimilar to T_Γ with respect to the relation $R^\delta \subseteq \mathbb{R}^n \times \mathbb{Z}^n$ defined by:*

$$R^\delta = \left\{(x', z) \in \mathbb{R}^n \times \mathbb{Z}^n \;\; : \;\; h_{\mathcal{B}}(x') = z\right\} \tag{15}$$

In practice it is not necessary to compute T_Γ to obtain a finite abstraction of T_Σ. Instead, we compute the finite abstraction directly from Σ as described in the next section. However, the introduction of T_Γ greatly simplifies the presentation of the forthcoming results.

5 Language equivalent finite abstractions

In this section we further restrict the set of observations O to be finite. We consider a finite set $S \subset \mathbb{Z}^n$ on the state space of T_Γ and define the observation space O to be:

$$O = S \cup \{\tau\} \tag{16}$$

for some $\tau \notin S$. The set S is identified with the finite set of atomic propositions $\mathcal{P}$ appearing in a given LTL specification formula. Finiteness of observations, and the shift register form of Γ will now allow to obtain a language equivalent, finite abstraction of T_Γ. This finite abstraction requires the following subsets of the state space, defined for any $a \in S$:

$$\begin{aligned} C_0^a &= \{a\} \\ C_1^a &= \mathrm{Pre}(C_0^a) \backslash S \\ C_2^a &= \mathrm{Pre}(C_1^a) \backslash S \\ &\vdots \\ C_{k_1}^a &= \mathrm{Pre}(C_{k_1-1}^a) \backslash S \end{aligned}$$

These sets have some properties that are fundamental to define the finite abstraction of T_Γ:

Proposition 2. *For any $0 \leq i \leq k_1$ and $a \in S$, the sets C_i^a are nonempty.*

Proposition 3. *For any $a \in S$, $b \in \mathbb{Z}^n$ there exists a $c \in C_{k_1}^a$ such that $b \in \mathrm{Pre}(c)$.*

Proposition 3 is the heart of the finite abstraction construction. It shows that, if it is possible to go from a state z to a state a without visiting S, then it is possible to do so in no more than $k_1 + 1$ steps. This bound on the number of steps implies a finite bound on the number of states necessary to encode the controlled dynamics between states in S and hence finiteness of the abstraction.

We now have all the necessary ingredients to construct a finite transition system $T_\Delta = (Q, L, \longrightarrow_\Delta, O, h_\Delta)$ defined by:

$$Q = S \times \{0, 1, 2, \ldots, k_1\} \tag{17}$$

$$L = \{1\} \tag{18}$$

$$O = S \cup \{\tau\} \tag{19}$$

$$h_\Delta((a,i)) = \begin{cases} a \text{ if } i = 0 \\ \tau \text{ if } i \neq 0 \end{cases} \tag{20}$$

It remains to define the transition relation $\longrightarrow_\Delta \subseteq Q \times L \times Q$. We proceed as follows:

$$((a,k_1),1,(a,k_1)) \in \longrightarrow_\Delta \quad \text{for any } a \in S, \tag{21}$$

$$((a,i),1,(a,i-1)) \in \longrightarrow_\Delta \quad \text{for any } a \in S \text{ and any } 1 \leq i \leq k_1, \tag{22}$$

$$((a,0),1,(b,0)) \in \longrightarrow_\Delta \quad \text{for any } a, b \in S \text{ such that } a \in \mathrm{Pre}(b), \tag{23}$$

$$((a,0),1,(b,i)) \in \longrightarrow_\Delta \quad \text{for any } a, b \in S \text{ such that } a \in \mathrm{Pre}(c) \text{ for some } c \in C_i^b,\ 1 \leq i \leq k_1 - 1, \tag{24}$$

$$((a,0),1,(b,k_1)) \in \longrightarrow_\Delta \quad \text{for any } a, b \in S. \tag{25}$$

Intuitively, we retain from T_Γ all the states $a \in S$ in the form $(a,0) \in Q$, as well as all the transitions between these states. However, to capture the controlled dynamics between states not in S, only k_1 states of additional memory are required for each $a \in S$ and these states are encoded by the states $(a,i) \in Q$ for $1 \leq i \leq k_1$.

The finite transition system T_Δ obtained through the previous construction is language equivalent to the infinite transition system T_Γ as asserted in the next result:

Theorem 5. *Let $T_\Delta = (Q, L, \longrightarrow_\Delta, O, h_\Delta)$ be the finite transition system obtained from $T_\Gamma = (\mathbb{Z}^n, \mathbb{Z}^m, \longrightarrow_\Gamma, O, h_\Gamma)$ by the previous construction and consider the observation space $O = S \cup \{\tau\}$ and the observation maps:*

$$h_\Gamma(z) = \begin{cases} z \text{ if } z \in S \\ \tau \text{ if } z \notin S \end{cases} \qquad h_\Delta((z,i)) = \begin{cases} z \text{ if } i = 0 \\ \tau \text{ if } i \neq 0 \end{cases}$$

Transition systems T_Γ and T_Δ are language equivalent.

Transition system T_Δ has $(k_1+1)|S|$ states, which shows that in the worst case the size of T_Δ grows linearly with the dimension of the control system. The size of T_Δ is also linear in the number of observables, which is natural, as the observed dynamics is also encoded in T_Δ. Furthermore, the size of T_Δ is independent of the grid resolution δ considered in the previous section. This is another important factor contributing for the scalability of the proposed results.

As our main result asserts decidability of model checking, we start by showing that the computation of T_Δ from Σ can be performed in a finite number of steps:

Lemma 1. *Let $\Sigma = (A, B)$ be a discrete time, controllable, linear system and consider a finite set of observations $O = S \cup \{\tau\}$, where $\tau \notin S$ and elements in S denote subsets of $\mathbb{R}^n$ of the form $P^{-1}h_{\mathcal{B}}^{-1}(h_{\mathcal{B}}(Px))$, $x \in \mathbb{R}^n$ and $h_{\mathcal{B}}^{-1}$ denoting the set valued inverse of $h_{\mathcal{B}}$. Then, constructing T_Δ from Σ can be performed in a finite number of steps.*

The main contribution of the paper is now obtained by combining Proposition 1 with Theorems 4 and 5, and Lemma 1. Intuitively the result follows from the ability to effectively construct T_Δ and from the fact that T_Σ is language equivalent to T_Δ under the appropriate observations.

Theorem 6. *Let Σ be a discrete time, controllable, linear system on $\mathbb{R}^n$ and φ any LTL formula where each atomic proposition denotes a subset of $\mathbb{R}^n$ of the form $P^{-1}h_{\mathcal{B}}^{-1}(h_{\mathcal{B}}(Px))$, $x \in \mathbb{R}^n$ and $h_{\mathcal{B}}^{-1}$ denoting the set valued inverse of $h_{\mathcal{B}}$. Then, determining if T_Σ satisfies φ is decidable.*

The previous result can also be summarized in the following diagram:

$$T_\Sigma \xrightarrow{bisimilar} T_{\Sigma'} \xrightarrow{bisimilar} T_\Gamma \xrightarrow{lang.\ eq.} T_\Delta$$

where the relationships between the several transition systems are represented. As bisimilarity implies language equivalence and language equivalence preserves LTL formulas, the previous figure shows that deciding if T_Σ is a model for a LTL formula is equivalent to decide if T_Δ is a model for the same formula. However, such translation between models is accompanied by an appropriate translation of the sets denoted by the atomic propositions. To see this, consider the following diagram representing the relation between the observation spaces of the several transitions systems:

$$\mathbb{R}^n \xrightarrow{P} \mathbb{R}^n \xrightarrow{h_{\mathcal{B}}} \mathbb{Z}^n \xrightarrow{h_\Gamma} S \cup \{\tau\}$$

We now see that an atomic proposition denoting a set of the form $P^{-1}h_{\mathcal{B}}^{-1}(h_{\mathcal{B}}(Px))$ for T_Σ is mapped by P to a set of the form $PP^{-1}h_{\mathcal{B}}^{-1}(h_{\mathcal{B}}(Px)) = h_{\mathcal{B}}^{-1}(h_{\mathcal{B}}(x'))$ for $T_{\Sigma'}$ and since this set is a quantization block, the quantization map $h_{\mathcal{B}}$ now transforms it to an element $h_{\mathcal{B}}(x') = z \in \mathbb{Z}^n$. The finite set S is chosen so as to capture every $z \in \mathbb{Z}^n$ which is the image of a set in $\mathbb{R}^n$ denoted by an

atomic proposition. Therefore observed sequences of transition system T_Δ represent sequences of atomic propositions interleaved with sequences of the element τ, denoting that no atomic proposition is satisfied.

The above discussion shows that there is a tradeoff between the decidability result of Theorem 6 and the requirement of having atomic propositions denoting sets of the special form $P^{-1}h_{\mathcal{B}}^{-1}(h_{\mathcal{B}}(Px))$. These sets are hypercubes skewed by the linear transformation P^{-1}, where the amount of linear distortion caused by P^{-1} is determined by the controllability properties of Σ. Nevertheless, these sets can be arbitrarily scaled by properly adjusting the resolution parameter δ and used to model any other set as a union of such skewed hypercubes.

6 Discussion

Theorem 6 shows that given a discrete time, controllable, linear system Σ, and a LTL formula φ with adequate atomic propositions, we can effectively decide if T_Σ is a model for φ. However, this result is far more interesting and important for control design than for verification. From the assumption of controllability, we know that every state is reachable from any other state, which somewhat reduces the interest in verification questions. Furthermore, we have used several times, in a fundamental way, the fact that control inputs u take values in an unbounded set, namely $\mathbb{R}^m$. This makes somewhat difficult to interpret the action of u as the environment or as a disturbance acting on the system. Nevertheless, these results seem extremely promising at the level of control design. Given a specification formula φ, one can determine the intersection of the language satisfying φ with the language of transition system T_Δ in the form of a Buchi automaton. Such automaton can then be refined to define a controller for T_Σ. This allows to design controllers for all the specifications expressible in LTL thereby enriching the class of specifications (and respective controller designs) usually used in control theory. We note that, although the system is assumed to be controllable, not every specification is achievable by the system. Even the simple formula $\varphi_1 \mathcal{U} \varphi_2$ may represent an unachievable specification as the system may need to leave the set denoted by atomic proposition φ_1 in order to reach the set denoted by φ_2.

7 Acknowledgments

The authors greatly acknowledge Salvatore La Torre, P. Madhusudan and Rajeev Alur for their guidance among many automata theoretic ramifications of temporal logic. This research was partially supported by the NSF Information Technology Research grant CCR01-21431 and NSF CAREER CCR-01-32716.

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. *Theoretical Computer Science*, 138:3–34, 1995.

2. R. Alur and D.L. Dill. A theory of timed automata. *Theoretical Computer Science*, 126:183–235, 1994.
3. Rajeev Alur, Thomas A. Henzinger, Gerardo Lafferriere, and George J. Pappas. Discrete abstractions of hybrid systems. *Proceedings of the IEEE*, 88:971–984, 2000.
4. E. Asarin, G. Schneider, and S. Yovine. On the decidability of the reachability problem for planar differential inclusions. In M. D. Di Benedetto and A. Sangiovanni-Vincentelli, editors, *Hybrid Systems: Computation and Control*, volume 2034 of *Lecture Notes in Computer Science*, pages 89–104. Springer-Verlag, 2001.
5. A. Bemporad and M.Morari. Control of systems integrating logic, dynamics and constraints. *Automatica*, 35(3):407–427, 1999.
6. Mireille Broucke. A geometric approach to bisimulation and verification of hybrid systems. In Fritz W. Vaandrager and Jan H. van Schuppen, editors, *Hybrid Systems: Computation and Control*, volume 1569 of *Lecture Notes in Computer Science*, pages 61–75. Springer-Verlag, 1999.
7. P. Brunovsky. A classification of linear controllable systems. *Kybernetika*, 6(3):173–188, 1970.
8. Edmund M. M. Clarke, Doron Peled, and Orna Grumberg. *Model Checking*. MIT Press, 1999.
9. J.E.R. Cury, B.H. Krogh, and T. Niinomi. Synthesis of supervisory controllers for hybrid systems based on approximating automata. *IEEE Transactions on Automatic Control : Special Issue on Hybrid Systems*, 43(4):564–568, April 1998.
10. E. A. Emerson. *Handbook of Theoretical Computer Science*, volume B, chapter Temporal and modal logic, pages 995–1072. Elsevier Science, 1990.
11. E. A. Emerson and E. M. Clarke. Using branching time temporal logic to synthesize synchronization skeletons. *Science of Computer Programming*, 2:241–266, 1982.
12. L.C.G.J.M. Habets and J. H. van Schuppen. Control of piecewise-linear hybrid systems on simplices and rectangles. In M. D. Di Benedetto and A. Sangiovanni-Vincentelli, editors, *Hybrid Systems: Computation and Control*, volume 2034 of *Lecture Notes in Computer Sience*, pages 261–274. Springer-Verlag, 2001.
13. T.A. Henzinger and R. Majumdar. Symbolic model checking for rectangular hybrid systems. In S. Graf, editor, *TACAS 2000: Tools and algorithms for the construction and analysis of systems*, Lecture Notes in Computer Science, New-York, 2000. Springer-Verlag.
14. Thomas A. Henzinger, Peter W. Kopke, Anuj Puri, and Pravin Varaiya. What's decidable about hybrid automata? *Journal of Computer and System Sciences*, 57:94–124, 1998.
15. R. E. Kalman. Kronecker invariants and feedback. In L. Weiss, editor, *Ordinary Differential Equations*, pages 459–471. Academic Press, New York, 1972.
16. Orna Kupferman, P. Madhusudan, P. S. Thiagarajan, and Moshe Y. Vardi. Open systems in reactive environments: Control and synthesis. In *Proceedings of the 11th International Conference on Concurency Theory*, volume 1877 of *Lecture Notes in Computer Science*, pages 92–107. Springer-Verlag, 2000.
17. Gerardo Lafferriere, George J. Pappas, and Shankar Sastry. O-minimal hybrid systems. *Mathematics of Control, Signals and Systems*, 13(1):1–21, March 2000.
18. P. Madhusudan and P.S. Thiagarajan. Branching time controllers for discrete event systems. *Theoretical Computer Science*, 274:117–149, March 2002.

19. Z. Manna and P. Wolper. Synthesis of communication processes from temporal logic specifications. *ACM Transactions on Programming Languages and Systems*, 6:68–93, 1984.
20. K. L. McMillan. *Symbolic Model Checking*. Kluwer Academic Publishers, 1993.
21. R. Milner. *Communication and Concurrency*. Prentice Hall, 1989.
22. T. Moor and J. M. Davoren. Robust controller synthesis for hybrid systems using modal logic. In M. D. Di Benedetto and A. Sangiovanni-Vincentelli, editors, *Hybrid Systems: Computation and Control*, volume 2034 of *Lecture Notes in Computer Science*. Springer-Verlag, 2001.
23. George J. Pappas. Bisimilar linear systems. *Automatica*, 2001. To appear.
24. D.M.R. Park. *Concurrency and automata on infinite sequences*, volume 104 of *Lecture Notes in Computer Science*. Springer-Verlag, 1980.
25. A. Puri and P. Varaiya. Decidability of hybrid systems with rectangular inclusions. In *Computer Aided Verification*, pages 95–104, 1994.
26. Eduardo D. Sontag. *Mathematical Control Theory*, volume 6 of *Texts in Applied Mathematics*. Springer-Verlag, New-York, 2nd edition, 1998.
27. Colin Stirling. *Handbook of logic in computer science*, volume 2, chapter Modal and Temporal Logics, pages 477–563. Oxford University Press, 1992.
28. J.A. Stiver, X.D. Koutsoukos, and P.J. Antsaklis. An invariant based approach to the design of hybrid control systems. *International Journal of Robust and Nonlinear Control*, 11(5):453–478, 2001.
29. Paulo Tabuada and George J. Pappas. Finite bisimulations of controllable linear systems. *Theoretical Computer Science*, January 2003. Submitted, available at `www.seas.upenn.edu/∼tabuadap`.