Finite Bisimulations for Switched Linear Systems

Ebru Aydin Gol, Xuchu Ding, Mircea Lazar, and Calin Belta

***Abstract*—In this paper, we consider the problem of constructing a finite bisimulation quotient for a discrete-time switched linear system in a bounded subset of its state space. Given a set of observations over polytopic subsets of the state space and a switched linear system with stable subsystems, the proposed algorithm generates the bisimulation quotient in a finite number of steps with the aid of sublevel sets of a polyhedral Lyapunov function. Starting from a sublevel set that includes the origin in its interior, the proposed algorithm iteratively constructs the bisimulation quotient for the region bounded by any larger sublevel set. We show how this bisimulation quotient can be used for synthesis of switching laws and verification with respect to specifications given as syntactically co-safe Linear Temporal Logic formulae over the observed polytopic subsets.**

***Index Terms*—Abstractions, formal methods, switched systems.**

I. Introduction

In recent years, there has been a trend to bridge the gap between control theory and formal methods. Control theory allows for analysis and control of "complex" dynamical systems with infinite state spaces, such as systems of controlled differential equations, against "simple" specifications, such as stability and reachability. In formal methods, "simple" systems, such as finite transition systems, are checked against "complex" (rich and expressive) specification languages, such as temporal logics. Recent studies show that certain classes of dynamical systems can be abstracted to finite transition systems. Applications in robotics [1], multi-agent control systems [2], and bioinformatics [3] show that model checking and automata games can be used to analyze and control systems with nontrivial dynamics from specifications given as temporal logic formulae.

In this paper, we focus on switched linear systems made of stable subsystems, and show that a finite bisimulation abstraction of the system can be efficiently constructed within some bounded subset of the state space. Since the bisimulation quotient preserves all properties that are expressible in frameworks as rich as μ-calculus, and implicitly Computation Tree Logic (CTL) and Linear Temporal Logic (LTL) (see e.g., [4]–[6]), it can be readily used for system verification and controller synthesis against such specifications. We show how our method can be used for both controller synthesis and verification from specifications given as arbitrary formulae of a fragment of LTL, called syntactically co-safe LTL (scLTL) [7]. For controller synthesis, we find the largest set of initial states and switching sequences such that all system trajectories satisfy a given formula. For verification, we find the largest set of initial states such that all system trajectories satisfy the formula under arbitrary switching.

The concept of constructing a finite quotient of an infinite system has been widely studied, e.g., [8]–[12]. It is known that finite state bisimulation quotients exist only for specific classes of systems (e.g., timed automata [10] and controllable linear systems [8]), and the well known bisimulation algorithm [4] in general does not terminate [13]. For piecewise linear systems, guided refinement procedures were employed with the goal of constructing an over-approximating quotient that can be used for verification of universal properties [9], [13].

We propose to obtain a finite bisimulation quotient of the system by only considering the system behavior within a relevant state space that does not contain the origin, i.e., in between two positively invariant compact sets that contain the origin. Our approach relies upon the existence of a common infinity norm Lyapunov function, which is a necessary condition for stability under arbitrary switching [14]. We propose to partition the state space by using sublevel sets of the Lyapunov function. Such sublevel sets, which are polytopic, allow us to generate the bisimulation quotient incrementally as the abstraction algorithm iterates, with no "holes" in the covered state space. Since we can obtain polytopic sublevel sets of any size from the Lyapunov function, the balance between the size of the abstracted state space and the amount of computation can be easily adjusted and controlled. Starting from the observation that the existence of the Lyapunov function renders the origin asymptotically stable for the switched system, its trajectories can only spend a finite time in the region of interest. As a result, we restrict our attention to LTL specifications that can be satisfied in finite time, such as scLTL formulae.

The construction of finite abstractions of dynamical systems by utilizing stability properties and Lyapunov functions was studied in [15], [16]. Approximately bisimilar finite abstractions for continuous-time switched systems were constructed under incremental stability assumptions in [15], where sublevel sets of a common Lyapunov function (or multiple Lyapunov functions with additional assumptions) were used. The abstract model was defined by quantizing the state space of

Manuscript received February 13, 2013; revised December 8, 2013 and April 19, 2014; accepted May 5, 2014. Date of publication August 28, 2014; date of current version November 18, 2014. This work was partially supported by the NSF under grants CNS-0834260 and CNS-1035588 and by the ONR under grant MURI N00014-09-1051 at Boston University, and by Veni grant 10230 at Eindhoven University of Technology. Recommended by Associate Editor G. J. Pappas.

E. Aydin Gol and C. Belta are with Boston University, Boston, MA 02215 USA (e-mail: ebru@bu.edu; cbelta@bu.edu).

X. Ding is with United Technologies Research Center, East Hartford, CT 06108 USA (e-mail: dingx@utrc.utc.com).

M. Lazar is with Eindhoven University of Technology, 5612 AZ Eindhoven, The Netherlands (e-mail: m.lazar@tue.nl).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TAC.2014.2351653

the switched system, and sampling the trajectories originating from the quantized state space. The approximate bisimulation relation guarantees that the trajectories of the abstract model and the original system are close to each other [17]. In [15], the accuracy of the abstraction was defined by the quantization parameter and the Lyapunov function. The method developed in [15] is not limited to linear systems, however, the abstraction is approximate, and the case studies presented in [15] highlight that a considerable accuracy requires a large abstract model. On the other hand, our method relies on efficient computation of one step controllable sets, hence suitable for linear systems. However, the resulting abstract model is exact in the sense that it produces the same set of observations as the original system. The authors of [15] relaxed the incremental stability assumption in their recent work on construction of approximate simulations [18]. Another conceptually related work is [16], where n Lyapunov functions were used for the abstraction of n-dimensional continuous-time Morse-Smale systems (e.g., hyperbolic linear systems) to timed automata. The abstraction proposed therein is weaker than bisimulation, but it can be used to verify safety properties. While both [16] and this work use sublevel sets for abstraction, the main difference between [16] and this approach comes from the usage of *polyhedral Lyapunov functions*, and therefore different classes of systems for which the methods apply. Our approach removes the need for multiple orthogonal Lyapunov functions, and we argue that it allows for a more tractable implementation since the abstraction of timed automata is expensive by itself [10], and polytopic sublevel sets ensure that the abstraction algorithm requires only basic operations with polytopic sets.

Preliminary versions of this work appeared in [19], [20]. In [19], we used polytopic sublevel sets to generate a bisimulation quotient for a discrete autonomous linear system, and in [20] we extended this approach to switched linear systems. Here we expand these preliminary versions by including analysis of complexity, more technical details, e.g., the proofs of the technical results, and illustrative case studies. In addition, we show how the proposed approach can be extended to piecewise linear systems and polytopic difference inclusion systems.

The rest of the paper is organized as follows. We introduce preliminaries in Section II and formulate the problem in Section III. We present the algorithm to generate the bisimulation quotient in Section IV, and analyze the complexity associated with it in Section V. We show in Section VI how the resulting bisimulation quotient can be used to synthesize switching control laws and verify the system behavior against temporal logic formulae. We illustrate the findings of the paper with examples in Section VII and summarize conclusions in Section VIII.

II. Preliminaries

For a set $\mathcal{S}$, $\mathrm{int}(\mathcal{S})$, $|\mathcal{S}|$, and $2^{\mathcal{S}}$ stand for its interior, cardinality, and power set, respectively. For $\lambda \in \mathbb{R}$ and $\mathcal{S} \subseteq \mathbb{R}^n$, let $\lambda\mathcal{S} := \{\lambda x \mid x \in \mathcal{S}\}$. We use $\mathbb{R}$, $\mathbb{R}_+$, $\mathbb{Z}$, and $\mathbb{Z}_+$ to denote the sets of real numbers, non-negative reals, integer numbers, and non-negative integers. For $m, n \in \mathbb{Z}_+$, we use $\mathbb{R}^n$ and $\mathbb{R}^{m\times n}$ to denote the set of column vectors and matrices with n and $m \times n$ real entries. For a vector v or a matrix A, we denote $v^\top$ or $A^\top$ as its transpose, respectively. For a vector $x \in \mathbb{R}^n$, $[x]_i$ denotes the i-th element of x and $\|x\|_\infty = \max_{i=1,\ldots,n} |[x]_i|$ denotes the infinity norm of x, where $|\cdot|$ denotes the absolute value. For a matrix $Z \in \mathbb{R}^{l\times n}$, let $\|Z\|_\infty := \sup_{x\neq 0}(\|Zx\|_\infty/\|x\|_\infty)$ denote its induced matrix infinity norm.

A n-dimensional *polytope* $\mathcal{P}$ (see, e.g., [21]) in $\mathbb{R}^n$ can be described as the convex hull of at least $n+1$ affinely independent points in $\mathbb{R}^n$. Alternatively, $\mathcal{P}$ can be described as the intersection of k, where $k \geq n+1$, closed half spaces, i.e., there exist $k \geq n+1$ and $H_\mathcal{P} \in \mathbb{R}^{k\times n}$, $h_\mathcal{P} \in \mathbb{R}^k$, such that

$$\mathcal{P} = \{x \in \mathbb{R}^n | H_\mathcal{P} x \leq h_\mathcal{P}\}.$$

We assume polytopes in $\mathbb{R}^n$ are n-dimensional unless noted otherwise. The set of boundaries of a polytope $\mathcal{P}$ are called *facets*, denoted by $f(\mathcal{P})$, which are themselves $(n-1)$-dimensional polytopes. A *semi-linear* set (also called a *polyhedron* in literature) in $\mathbb{R}^n$ is defined as finite unions, intersections and complements of sets $\{x \in \mathbb{R}^n \mid a^\top x \sim b, \sim \in \{=, <\}\}$, for some $a \in \mathbb{R}^n$ and $b \in \mathbb{R}$. Note that a convex and bounded semi-linear set is equivalent to a polytope with some (or none) of its facets removed.

A. Transition Systems and Bisimulations

Definition 2.1: A transition system (TS) is a tuple $\mathcal{T} = (Q, \Sigma, \rightarrow, \Pi, h)$, where

- Q is a (possibly infinite) set of states;
- Σ is a set of inputs;
- $\rightarrow \subseteq Q \times \Sigma \times Q$ is a set of transitions;
- Π is a finite set of observations; and
- $h : Q \longrightarrow 2^\Pi$ is an observation map.

We denote $x \xrightarrow{\sigma} x'$ if $(x, \sigma, x') \in \rightarrow$. We assume $\mathcal{T}$ to be non-blocking, i.e., for each $x \in Q$, there exists $x' \in Q$ and $\sigma \in \Sigma$ such that $x \xrightarrow{\sigma} x'$. An *input word* is defined as an infinite sequence $\boldsymbol{\sigma} = \sigma_0\sigma_1 \ldots$ where $\sigma_k \in \Sigma$ for all $k \in \mathbb{Z}_+$. A *trajectory* of $\mathcal{T}$ produced by an input word $\boldsymbol{\sigma} = \sigma_0\sigma_1 \ldots$ and originating at state x_0 is an infinite sequence $\mathbf{x} = x_0x_1 \ldots$ where $x_k \xrightarrow{\sigma_k} x_{k+1}$ for all $k \in \mathbb{Z}_+$. A trajectory $\mathbf{x}$ generates a word $\mathbf{o} = o_0o_1 \ldots$, where $o_k \subseteq \Pi$ is the set of observations of state x_k and defined as $o_k := h(x_k)$ for all $k \in \mathbb{Z}_+$.

The TS $\mathcal{T}$ is *finite* if $|Q| < \infty$ and $|\Sigma| < \infty$, otherwise $\mathcal{T}$ is *infinite*. Moreover, $\mathcal{T}$ is *deterministic* if $x \xrightarrow{\sigma} x'$ implies that there does not exist $x'' \neq x'$ such that $x \xrightarrow{\sigma} x''$; otherwise, $\mathcal{T}$ is called *non-deterministic*. Given a set $\mathcal{Q} \subseteq Q$, we define the set of states $\mathrm{Pre}_\mathcal{T}(\mathcal{Q}, \sigma)$ that reach $\mathcal{Q}$ in one step when input σ is applied as

$$\mathrm{Pre}_\tau(\mathcal{Q}, \sigma) := \{x \in Q | \exists x' \in \mathcal{Q}, x \xrightarrow{\sigma} x'\}. \tag{1}$$

States of a TS can be related by a relation $\sim \subseteq Q \times Q$. For convenience of notation, we denote $x \sim x'$ if $(x, x') \in \sim$.

Definition 2.2: We say that a relation $\sim$ is *observation preserving* if for any $x, x' \in Q$, $x \sim x'$ implies that $h(x) = h(x')$.

For an observation preserving relation $\sim$, the subset $\mathcal{Q} \subseteq Q$ is called an *equivalence class* if $x, x' \in \mathcal{Q} \Leftrightarrow x \sim x'$. We denote by $Q/_\sim$ the set labeling all equivalence classes and define a map $\mathrm{eq} : Q/_\sim \mapsto 2^Q$ such that $\mathrm{eq}(X)$ is the set of states in the equivalence class $\mathcal{Q} \in Q/_\sim$ labeled by X.

Definition 2.3: A finite *partition* P of a set $\mathcal{S}$ is a finite collection of sets $P := \{P_i\}_{i\in I}$, such that $\cup_{i\in I} P_i = \mathcal{S}$ and $P_i \cap P_j = \emptyset$ if $i \neq j$. A finite *refinement* of P is a finite partition P' of $\mathcal{S}$ such that for each $P_i \in P'$, there exists $P_j \in P$ such that $P_i \subseteq P_j$.

A partition naturally induces a relation, and an observation preserving relation induces a quotient TS. One can immediately verify that a refinement of an observation preserving partition is also observation preserving.

Definition 2.4: An observation preserving relation $\sim$ of a TS $\mathcal{T} = (Q, \Sigma, \rightarrow, \Pi, h)$ induces a *quotient transition system* $\mathcal{T}/_\sim = (Q/_\sim, \Sigma, \rightarrow_\sim, \Pi, h_\sim)$, where $Q/_\sim$ is the set labeling all equivalence classes. The transitions of $\mathcal{T}/_\sim$ are defined as $X \xrightarrow{\sigma}_\sim Y$ if and only if there exists $x \in \mathrm{eq}(X)$ and $x' \in \mathrm{eq}(Y)$ such that $x \xrightarrow{\sigma} x'$. The observation map is defined as $h_\sim(X) := h(x)$, where $x \in \mathrm{eq}(X)$.

Definition 2.5: Given a TS $\mathcal{T} = (Q, \Sigma, \rightarrow, \Pi, h)$, a relation $\sim$ is a bisimulation relation of $\mathcal{T}$ if (1) $\sim$ is observation preserving; and (2) for any $x_1, x_2 \in Q, \sigma \in \Sigma$, if $x_1 \sim x_2$ and $x_1 \xrightarrow{\sigma} x_1'$, then there exists $x_2' \in Q$ such that $x_2 \xrightarrow{\sigma} x_2'$ and $x_1' \sim x_2'$.

If $\sim$ is a bisimulation, then the quotient transition system $\mathcal{T}/_\sim$ is called a *bisimulation quotient* of $\mathcal{T}$. In this case, $\mathcal{T}$ and $\mathcal{T}/_\sim$ are said to be *bisimilar*. Bisimulation is a very strong equivalence relation between systems. In fact, it preserves properties expressed in temporal logics such as LTL, CTL and μ-calculus [4]–[6]. As such, it is used as an important tool to reduce the complexity of system verification or controller synthesis, since the bisimulation quotient (which may be finite) can be verified or used for controller synthesis instead of the original system.

B. Polyhedral Lyapunov Functions

Consider an autonomous discrete-time system,

$$x_{k+1} = \Phi(x_k), \quad k \in \mathbb{Z}_+ \tag{2}$$

where $x_k \in \mathbb{R}^n$ is the state at the discrete-time instant k and $\Phi : \mathbb{R}^n \mapsto \mathbb{R}^n$ is an arbitrary map with $\Phi(0) = 0$. Given a state $x \in \mathbb{R}^n$, $x' := \Phi(x)$ is called a *successor* state of x.

Definition 2.6: Let $\lambda \in [0, 1]$. We call a set $\mathcal{P} \subseteq \mathbb{R}^n$ λ-*contractive* (shortly, contractive) if for all $x \in \mathcal{P}$ it holds that $\Phi(x) \in \lambda\mathcal{P}$. For $\lambda = 1$, we call $\mathcal{P}$ a *positively invariant* set.

A function $\alpha : \mathbb{R}_+ \to \mathbb{R}_+$ belongs to class $\mathcal{K}_\infty$ if it is continuous, strictly increasing, $\alpha(0) = 0$ and $\lim_{s\to\infty} \alpha(s) = \infty$.

Theorem 2.1: Let $\mathcal{X}$ be a positively invariant set for (2) with $0 \in \mathrm{int}(\mathcal{X})$. Furthermore, let $\alpha_1, \alpha_2 \in \mathcal{K}_\infty$, $\rho \in (0, 1)$ and $V : \mathbb{R}^n \mapsto \mathbb{R}_+$ such that:

$$\alpha_1(\|x\|) \leq V(x) \leq \alpha_2(\|x\|), \forall\, x \in \mathcal{X} \tag{3}$$

$$V(\Phi(x)) \leq \rho V(x), \forall\, x \in \mathcal{X}. \tag{4}$$

Then system (2) is asymptotically stable in $\mathcal{X}$.

The proof of Theorem 2.1 can be found in [22], [23].

Definition 2.7: A function $V : \mathbb{R}^n \mapsto \mathbb{R}_+$ is called a *Lyapunov function* (LF) in $\mathcal{X}$ if it satisfies (3) and (4). If $\mathcal{X} = \mathbb{R}^n$, then V is called a *global Lyapunov function*.

The parameter ρ is called the *contraction rate* of V. For any $\Gamma > 0$, $\mathcal{P}_\Gamma := \{x \in \mathbb{R}^n \mid V(x) \leq \Gamma\}$ is called a *sublevel set* of V.

For the remainder of this paper we consider LFs defined using the infinity norm, i.e.,

$$V(x) = \|Lx\|_\infty, \quad L \in \mathbb{R}^{l\times n}, l \geq n, l \in \mathbb{Z}_+ \tag{5}$$

where L has full-column rank. Notice that infinity norm Lyapunov functions are a particular type of polyhedral Lyapunov functions. We opted for this type of function to simplify the exposition but in fact, the proposed abstraction method applies to general polyhedral Lyapunov functions defined by Minkowski (gauge) functions of polytopes in $\mathbb{R}^n$ with the origin in their interior.

Proposition 2.1: Suppose that $L \in \mathbb{R}^{l\times n}$ has full-column rank and V as defined in (5) is a global LF for system (2) with contraction rate $\rho \in (0, 1)$. Then for all $\Gamma > 0$ it holds that $\mathcal{P}_\Gamma$ is a polytope and $0 \in \mathrm{int}(\mathcal{P}_\Gamma)$. Moreover, if $\Phi(x) = Ax$ for some $A \in \mathbb{R}^{n\times n}$, then *for all* $\Gamma > 0$ it holds that $\mathcal{P}_\Gamma$ is a ρ-contractive polytope for (2).

The proof of the above result is a straightforward application of results in [24], [25].

In this paper, we will consider switched systems that are stable under arbitrary switching. In this case, the dynamics corresponding to (2) becomes a difference inclusion, i.e., $x' \in \Phi(x)$, $\Phi(x) := \{Ax \mid A \in \mathcal{A}\}$ for some set $\mathcal{A} \subseteq \mathbb{R}^{n\times n}$. It has been shown in [24], [25], that all the above definitions (invariant set, Lyapunov function) and results apply directly to difference inclusions in the absolute sense, i.e., given x, the corresponding conditions must hold for all $x' \in \Phi(x)$.

III. Problem Formulation

In this paper, we consider discrete-time switched linear systems, i.e.,

$$x_{k+1} = A_{\boldsymbol{\sigma}(k)} x_k, \quad \boldsymbol{\sigma}(k) \in \Sigma, k \in \mathbb{Z}_+ \tag{6}$$

where $\boldsymbol{\sigma} : \mathbb{Z}_+ \to \Sigma$ is a switching sequence that selects the active subsystem from a finite index set Σ and $A_i \in \mathbb{R}^{n\times n}$ is a strictly stable (i.e., Schur) matrix for all $i \in \Sigma$. We assume that a common polyhedral Lyapunov function (LF) of the form (5) with contraction rate $\rho \in (0, 1)$ is known for system (6). The algorithm proposed in [25] is employed to construct such a function with a desired contraction rate.

Let $\mathcal{X}$ be a polytope $\mathcal{X} := \{x \mid \|Lx\|_\infty \leq \Gamma_\mathcal{X}\}$ and $\mathcal{D}$ be a polytope $\mathcal{D} := \{x \mid \|Lx\|_\infty \leq \Gamma_\mathcal{D}\}$, where L corresponds to the polyhedral LF (5) of system (6) and we assume that $0 < \Gamma_\mathcal{D} < \Gamma_\mathcal{X}$. Note that $\mathcal{D} \subset \mathcal{X}$ and $0 \in \mathrm{int}(\mathcal{D}) \subset \mathrm{int}(\mathcal{X})$. We call $\mathcal{X}$ the working set and $\mathcal{D}$ the target set. We are interested in synthesis of control strategies and verification of the system behavior within $\mathcal{X}$ with respect to polytopic regions in the state space, until the target set $\mathcal{D}$ is reached (since $\mathcal{D}$ is positively invariant, the system trajectory will be confined within $\mathcal{D}$ after $\mathcal{D}$ is reached).

We assume that there exists a set $\mathcal{R}$ of polytopes indexed by a finite set R, i.e., $\mathcal{R} := \{\mathcal{R}_i\}_{i\in R}$, where $\mathcal{R}_i \subseteq \mathcal{X} \setminus \mathcal{D}$ for all $i \in R$, and $\mathcal{R}_i \cap \mathcal{R}_j = \emptyset$ for any $i \neq j$. The set $\mathcal{R}$ represents regions of interest in the relevant state space, and the polytopes in $\mathcal{R}$ are considered as observations of (6). Therefore, a trajectory of (6) $x_0 x_1 \ldots$ produces an infinite sequence of observations

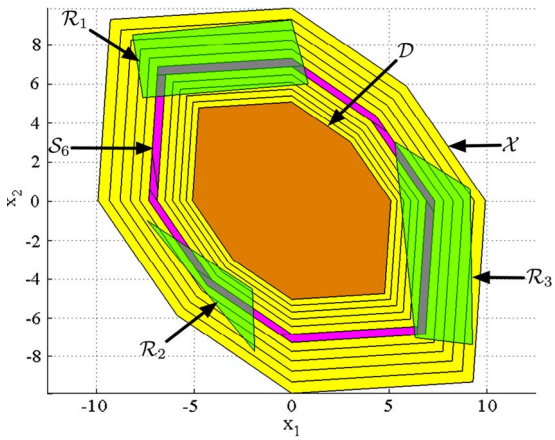

Fig. 1. An example in $\mathbb{R}^2$ of the working set $\mathcal{X}$, the target set $\mathcal{D}$ (in brown), a set of observational relevant polytopes $\mathcal{R} = \{\mathcal{R}_1, \mathcal{R}_2, \mathcal{R}_3\}$ (in transparent green), sublevel sets with $N = 11$ and one slice $\mathcal{S}_6$ (in purple).

$o_0 o_1 \ldots$, such that o_i is the index of the polytope in $\mathcal{R}$ visited by state x_i, or $o_i = \emptyset$ if x_i is in none of the polytopes.

Example 3.1: Consider a system as in (6), $\Sigma = \{1, 2\}$, $A_1 = \begin{pmatrix} -0.65 & 0.32 \\ -0.42 & -0.92 \end{pmatrix}$ and $A_2 = \begin{pmatrix} 0.65 & 0.32 \\ -0.42 & -0.92 \end{pmatrix}$. The algorithm proposed in [25] is employed to construct a global polyhedral LF of the form (5), where

$$L = \begin{pmatrix} -0.0625 & 0.6815 & 0.9947 & 0.9947 \\ 1 & 1 & -0.6868 & -0.0678 \end{pmatrix}^\top$$

and $\rho = 0.94$. We chose $\Gamma_{\mathcal{X}} = 10$ and $\Gamma_{\mathcal{D}} = 5.063$. (see Fig. 1 for polytopes $\mathcal{X}$, $\mathcal{D}$, and a set of polytopes $\mathcal{R}$.) ■

The semantics of the system can be formalized through an embedding of (6) into a transition system, as follows.

Definition 3.1: Let $\mathcal{X}$, $\mathcal{D}$, and $\mathcal{R} = \{\mathcal{R}_i\}_{i \in R}$ be given. The embedding transition system for (6) is a transition system $\mathcal{T}_e = (Q_e, \Sigma, \rightarrow_e, \Pi, h_e)$ where

- $Q_e := \mathcal{X}$;
- Σ is the same as the index set given in (6);
- 1) If $x \in \mathcal{X} \setminus \mathcal{D}$, then $x \xrightarrow{\sigma}_e x'$ if and only if $x' = A_\sigma x$, i.e., x' is the state at the next time-step after applying the dynamics of (6) at x when subsystem σ is active;
  2) If $x \in \mathcal{D}$, $x \xrightarrow{\sigma}_e x$ for all $\sigma \in \Sigma$ (since the target set $\mathcal{D}$ is already reached, we consider the behavior of the system thereafter no longer relevant);
- $\Pi = R \cup \{\Pi_{\mathcal{D}}\}$, i.e., the set of observations is the set of labels of regions, plus the label $\Pi_{\mathcal{D}}$ for $\mathcal{D}$;
- 1) $h_e(x) := \{i\}$ if and only if $x \in \mathcal{R}_i$;
  2) $h_e(x) := \emptyset$ if and only if $x \in \mathcal{X} \setminus (\mathcal{D} \cup \bigcup_{i \in R} \mathcal{R}_i)$;
  3) $h_e(x) := \{\Pi_{\mathcal{D}}\}$ if and only if $x \in \mathcal{D}$.

Note that $\mathcal{T}_e$ is deterministic and it has an infinite number of states. Moreover, $\mathcal{T}_e$ exactly captures the system dynamics under (6) in the relevant state space $\mathcal{X} \setminus \mathcal{D}$, since a transition of $\mathcal{T}_e$ naturally corresponds to the evolution of the discrete-time system in one time-step. Indeed, within $\mathcal{X} \setminus \mathcal{D}$, the trajectory of $\mathcal{T}_e$ produced by an input word $\boldsymbol{\sigma}$ from a state $x \in \mathcal{X} \setminus \mathcal{D}$ is exactly the same as the trajectory of system (6) from x under the switching sequence $\boldsymbol{\sigma}$.

We now formulate the main problem considered in this paper.

Problem 3.1: Let a system (6) with a polyhedral Lyapunov function of the form (5), sets $\mathcal{X}$, $\mathcal{D}$, and $\{\mathcal{R}_i\}_{i \in R}$ be given. Compute a finite observation preserving partition P such that its induced relation $\sim$ is a bisimulation of the embedding transition system $\mathcal{T}_e$, and obtain the corresponding bisimulation quotient $\mathcal{T}_e/_\sim$.

Remark 3.1: The above assumptions on the sets $\mathcal{X}$, $\mathcal{D}$, and $\{\mathcal{R}_i\}_{i \in R}$ are made for simplicity of presentation. The problem formulation and the approach described in the rest of the paper can be easily extended to arbitrary positively invariant sets $\mathcal{X}$ and $\mathcal{D}$ with $\mathcal{D} \subseteq \mathcal{X}$, i.e., not obtained as the sublevel sets of (5), by considering the largest sublevel set that is included in $\mathcal{D}$ and the smallest sublevel set that includes $\mathcal{X}$ ($\Gamma_{\mathcal{D}}$ and $\Gamma_{\mathcal{X}}$ can be made arbitrarily small and arbitrary large, respectively, so as to capture any compact subset of $\mathbb{R}^n$). Also, the set of polytopes of interest $\{\mathcal{R}_i\}_{i \in R}$ can be relaxed to a finite set of linear predicates in x as defined in [26].

IV. Bisimulation Quotient

Starting from a polyhedral Lyapunov function $V(x) = \|Lx\|_\infty$ with a contraction rate $\rho \in (0, 1)$ as described in Section II-B for system (6), we first generate a sequence of polytopic sublevel sets of the form $\mathcal{P}_\Gamma := \{x \in \mathbb{R}^n \mid \|Lx\|_\infty \leq \Gamma\}$ as follows. Recall that $\mathcal{X} = \mathcal{P}_{\Gamma_{\mathcal{X}}}$ and $\mathcal{D} = \mathcal{P}_{\Gamma_{\mathcal{D}}}$ for some $0 < \Gamma_{\mathcal{D}} < \Gamma_{\mathcal{X}}$. We define a finite sequence $\bar{\Gamma} := \Gamma_0, \ldots, \Gamma_N$, where

$$\Gamma_{i+1} = \rho^{-1}\Gamma_i, \; i = 0, \ldots, N-2 \tag{7}$$

$\Gamma_0 := \Gamma_{\mathcal{D}}$, $\Gamma_N := \Gamma_{\mathcal{X}}$, and

$$N := \arg\min_N \{\rho^{-N}\Gamma_0 | \rho^{-N}\Gamma_0 \geq \Gamma_{\mathcal{X}}\}. \tag{8}$$

This choice of N guarantees that $\mathcal{P}_{\Gamma_{N-1}}$ is the largest sublevel set defined via (7) that is a subset of $\mathcal{X}$. Since Γ_N is exactly $\Gamma_{\mathcal{X}}$, $\mathcal{P}_{\Gamma_N}$ is exactly $\mathcal{X}$.

The sequence $\bar{\Gamma}$ generates a sequence of sublevel sets $\bar{\mathcal{P}}_\Gamma := \mathcal{P}_{\Gamma_0}, \ldots, \mathcal{P}_{\Gamma_N}$. From the definition of the sublevel sets and $\bar{\Gamma}$, we have that

$$\mathcal{P}_{\Gamma_0} \subset \ldots \subset \mathcal{P}_{\Gamma_N}. \tag{9}$$

Next, we define a *slice* of the state space as follows:

$$\mathcal{S}_i := \mathcal{P}_{\Gamma_i} \setminus \mathcal{P}_{\Gamma_{i-1}}, \; i = 1, \ldots, N. \tag{10}$$

For convenience, we also denote $\mathcal{S}_0 := \mathcal{P}_{\Gamma_0}$ (although $\mathcal{S}_0$ is not a slice in between two sublevel sets). We immediately see that the sets $\{\mathcal{S}_i\}_{i=0,\ldots,N}$ form a partition of $\mathcal{X}$. Note that the slices are bounded semi-linear sets (see Section II).

Example 4.1 (Example 3.1 Continued): Consider the system given in Example 3.1. The sequence $\bar{\Gamma}$ is computed from $\Gamma_{\mathcal{X}}$, $\Gamma_{\mathcal{D}}$, and ρ as described above, which resulted in $N = 11$. The polytopic sublevel sets $\bar{\mathcal{P}}_\Gamma := \mathcal{P}_{\Gamma_0}, \ldots, \mathcal{P}_{\Gamma_{11}}$ are shown in Fig. 1. ■

Proposition 4.1: Assume that the set of slices $\{\mathcal{S}_i\}_{i=0,\ldots,N}$ is obtained from a sequence $\bar{\Gamma}$ satisfying (7). Given a state x in the i-th slice, i.e., $x \in \mathcal{S}_i$, where $1 \leq i \leq N$, its successor state ($x' = A_\sigma x$, $\sigma \in \Sigma$) satisfies $x' \in \mathcal{S}_j$ for some $j < i$.

Proof: From Proposition 2.1, we have that $\mathcal{P}_{\Gamma_i}$ are ρ-contractive. By the definition of a ρ-contractive set (Definition 2.6), we have that $x' = A_\sigma x \in \rho\mathcal{P}_{\Gamma_i} = \{x \in \mathbb{R}^n \mid \|Lx\|_\infty \leq \rho\Gamma_i\}$ for all $\sigma \in \Sigma$. From (7), we have $\rho\Gamma_i = \Gamma_{i-1}$. Therefore $\mathcal{P}_{\Gamma_{i-1}} = \{x \in \mathbb{R}^n \mid \|Lx\|_\infty \leq \Gamma_{i-1}\}$ implies that $\mathcal{P}_{\Gamma_{i-1}} = \{x \in \mathbb{R}^n \mid \|Lx\|_\infty \leq \rho\Gamma_i\}$ and hence $\mathcal{P}_{\Gamma_{i-1}} = \rho\mathcal{P}_{\Gamma_i}$ and $x' \in \mathcal{P}_{\Gamma_{i-1}}$. From the definition of slices (10), $x' \in \mathcal{S}_j$ for some $j < i$. ■

The state space of $\mathcal{T}_e$ (which is the working set $\mathcal{X}$) can be naturally partitioned as

$$P_{\mathcal{X}} := \left\{ \{\mathcal{R}_i\}_{i\in R}, \mathcal{X} \setminus \left(\mathcal{D} \cup \bigcup_{i\in R} \mathcal{R}_i \right), \mathcal{D} \right\}. \quad (11)$$

The relation induced from partition $P_{\mathcal{X}}$ is observation preserving (see Section II-A).

The proposed abstraction algorithm computes the bisimulation quotient by iteratively refining an observation preserving partition with respect to one step controllable sets. We first explain two procedures, `ComputePre` and `RefineUpdate`, which are used by the main abstraction algorithm. The procedure $\texttt{ComputePre}(\tilde{\mathcal{P}}, \sigma)$ takes as input $\tilde{\mathcal{P}}$, which is a bounded semi-linear set (e.g., a slice), and $\sigma \in \Sigma$, which is the switching input, and returns the set $\mathrm{Pre}_{\mathcal{T}_e}(\tilde{\mathcal{P}}, \sigma)$. In general, if $\tilde{\mathcal{P}}$ is a semi-linear set, then $\mathrm{Pre}_{\mathcal{T}_e}(\tilde{\mathcal{P}}, \sigma)$ is also a semi-linear set and it can be computed via quantifier elimination [27]. In particular, the computation of $\mathrm{Pre}_{\mathcal{T}_e}$ for a bounded semi-linear set $\tilde{\mathcal{P}}$ falls into one of the following cases:

(i) If $\tilde{\mathcal{P}}$ is a polytope, then $\mathrm{Pre}_{\mathcal{T}_e}(\tilde{\mathcal{P}}, \sigma)$ is computed as:

$$\mathrm{Pre}_{\tau_e}(\tilde{\mathcal{P}}, \sigma) = \{x \in \mathbb{R}^n | H_{\tilde{\mathcal{P}}} A_\sigma x \leq h_{\tilde{\mathcal{P}}}\} \quad (12)$$

which is a possibly degenerate polytope in $\mathbb{R}^n$. Note that (12) applies to a polytope $\tilde{\mathcal{P}}$ of any dimension;

(ii) If $\tilde{\mathcal{P}}$ is a union of polytopes, one can use a standard convex decomposition method to decompose $\tilde{\mathcal{P}}$ into a set of polytopes $\{\mathcal{P}_i\}_{i\in I}$ (see, e.g., [21]), and compute $\mathrm{Pre}_{\mathcal{T}_e}(\tilde{\mathcal{P}}, \sigma)$ as $\cup_{i\in I}\mathrm{Pre}_{\mathcal{T}_e}(\mathcal{P}_i, \sigma)$ using (12);

(iii) If $\tilde{\mathcal{P}}$ is a convex and bounded semi-linear set, then $\tilde{\mathcal{P}} = \mathcal{P} \setminus \cup_{i\in I}\mathcal{F}_i$ for some polytope $\mathcal{P}$ and its facet $\mathcal{F}_i \in f(\mathcal{P})$. Since $\mathcal{T}_e$ is deterministic, we have $\mathrm{Pre}_{\mathcal{T}_e}(\tilde{\mathcal{P}}, \sigma) = \mathrm{Pre}_{\mathcal{T}_e}(\mathcal{P}, \sigma) \setminus \mathrm{Pre}_{\mathcal{T}_e}(\cup_{i\in I}\mathcal{F}_i, \sigma)$, where the second term can be computed as described in case (*ii*);

(iv) If $\tilde{\mathcal{P}}$ is a general (non-convex) bounded semi-linear set, then again it can be decomposed into convex and bounded semi-linear sets $\tilde{\mathcal{P}} = \cup_{i\in I}\tilde{\mathcal{P}}_i$. Then $\mathrm{Pre}_{\mathcal{T}_e}(\tilde{\mathcal{P}}, \sigma) = \cup_{i\in I}\mathrm{Pre}(\tilde{\mathcal{P}}_i, \sigma)$, and each $\mathrm{Pre}(\tilde{\mathcal{P}}_i, \sigma)$ can be computed as described in case (*iii*).

As summarized above, we see that $\texttt{ComputePre}(\tilde{\mathcal{P}}, \sigma)$ can always be implemented by convex decompositions and repeated applications of (12), and thus $\texttt{ComputePre}(\tilde{\mathcal{P}}, \sigma)$ only requires basic operations with polytopic sets.

The procedure $\texttt{RefineUpdate}(P, \mathcal{T}, \tilde{\mathcal{P}}, \sigma, q)$ (outlined in Algorithm 1) refines a partition P with respect to set $\tilde{\mathcal{P}}$, where $\tilde{\mathcal{P}} = \texttt{ComputePre}(\mathrm{eq}(q), \sigma)$. It then updates $\mathcal{T}$. If P consists of only bounded semi-linear sets and $\tilde{\mathcal{P}}$ is a semi-linear set, then the resulting refinement P^+ consists of only bounded semi-linear sets. This fact allows us to compute $\mathrm{Pre}_{\mathcal{T}_e}(\tilde{\mathcal{P}}, \sigma)$ with $\texttt{ComputePre}(\tilde{\mathcal{P}}, \sigma)$ only taking bounded semi-linear sets as inputs.

Algorithm 1 $[P^+, \mathcal{T}^+] = \texttt{RefineUpdate}(P, \mathcal{T}, \tilde{\mathcal{P}}, \sigma, q)$

Input: A TS $\mathcal{T} = (Q, \Sigma, \to, \Pi, h)$, a partition P where $\mathrm{eq}(q') \in P$ for all $q' \in Q$, and $\tilde{\mathcal{P}} = \texttt{ComputePre}(\mathrm{eq}(q), \sigma)$ for some $q \in Q$, $\sigma \in \Sigma$.
Output: P^+ is a finite refinement of P with respect to $\tilde{\mathcal{P}}$, $\mathcal{T}^+$ is a TS updated from $\mathcal{T}$.
1: Set $P^+ = P$ **and** $\mathcal{T}^+ = \mathcal{T}$.
2: **for all** $q' \in Q^+$ such that $\mathrm{eq}(q') \cap \tilde{\mathcal{P}} \neq \emptyset$ **do**
3: Replace q' in Q^+ by $\{q_1, q_2\}$ and set $\mathrm{eq}(q_1) = \mathrm{eq}(q') \cap \tilde{\mathcal{P}}$, $\mathrm{eq}(q_2) = \mathrm{eq}(q') \setminus \tilde{\mathcal{P}}$.
4: Replace $\mathrm{eq}(q')$ in P^+ by $\{\mathrm{eq}(q_1), \mathrm{eq}(q_2)\}$.
5: Replace each $(q', \sigma', q'') \in \to^+$ by $\{(q_i, \sigma', q'')\}_{i=1,2}$.
6: Add transition (q_1, σ, q) to $\to^+$.
7: **end for**

We now present the abstraction algorithm (see Algorithm 2) that computes the bisimulation quotient. The main idea is to start with $P_{\mathcal{X}}$ (11), refine the partition according to $\{\mathcal{S}_i\}_{i=0,\ldots,N}$ so that it is a refinement to both $P_{\mathcal{X}}$, and $\{\mathcal{S}_i\}_{i=0,\ldots,N}$, and then iteratively refine using the Pre operator (1) until the bisimulation quotient is obtained. Starting with $P_{\mathcal{X}}$ is necessary so that the partition is observation preserving. The second step guarantees that each element in the partition is included in a slice. The third step allows us to ensure that at iteration i of the algorithm, the bisimulation quotient for states within $\mathcal{P}_{\Gamma_i}$ is completed.

Algorithm 2 Abstraction algorithm

Input: System dynamics (6), polyhedral LF $V(x) = \|Lx\|_\infty$ with a contractive rate ρ, sets $\mathcal{X}$, $\mathcal{D}$, and $\{\mathcal{R}_i\}_{i\in R}$.
Output: Bisimulation quotient $\mathcal{T}_e/_\sim$ of the embedding transition system $\mathcal{T}_e$ and the corresponding observation pre serving partition P.
1: Obtain $P_{\mathcal{X}}$ as in (11).
2: Generate the sequence of sublevel sets $\bar{\mathcal{P}}_\Gamma = \mathcal{P}_{\Gamma_0}, \ldots, \mathcal{P}_{\Gamma_N}$ and slices $\mathcal{S}_0, \ldots, \mathcal{S}_N$ as defined in (10).
3: Set $P_0 = \{\tilde{\mathcal{P}}_1 \cap \tilde{\mathcal{P}}_2 \mid \tilde{\mathcal{P}}_1 \in P_{\mathcal{X}}, \tilde{\mathcal{P}}_2 \in \{\mathcal{S}_i\}_{i=0,\ldots,N}, \tilde{\mathcal{P}}_1 \cap \tilde{\mathcal{P}}_2 \neq \emptyset\}$.
4: Initialize $\mathcal{T}_e/_{\sim_0}$ by setting $Q_e/_{\sim_0}$ as the set labeling P_0. Set transitions only for the state $q_{\mathcal{D}} \in Q_e/_{\sim_0}$ where $\mathrm{eq}(q_{\mathcal{D}}) = \mathcal{S}_0 = \mathcal{D}$ with $q_{\mathcal{D}} \xrightarrow{\sigma}_{\sim_0} q_{\mathcal{D}}$ for all $\sigma \in \Sigma$.
5: **for** each $i = 0, \ldots, N-1$ **do**
6: Set $\mathcal{T}_e/_{\sim_{i+1}} = \mathcal{T}_e/_{\sim_i}$ **and** $P_{i+1} = P_i$.
7: **for** each $q \in Q_e/_{\sim_i}$ where $\mathrm{eq}(q) \subseteq \mathcal{S}_i$ **do**
8: **for** each $\sigma \in \Sigma$ **do**
9: Find $\tilde{\mathcal{P}} = \texttt{ComputePre}(\mathrm{eq}(q), \sigma)$.
10: Set $[P_{i+1}, \mathcal{T}_e/_{\sim_{i+1}}] = \texttt{RefineUpdate}(P_{i+1}, \mathcal{T}_e/_{\sim_{i+1}}, \tilde{\mathcal{P}}, \sigma, q)$.
11: **end for**
12: **end for**
13: **end for**
14: Return $\mathcal{T}_e/_{\sim_N}$ and P_N.

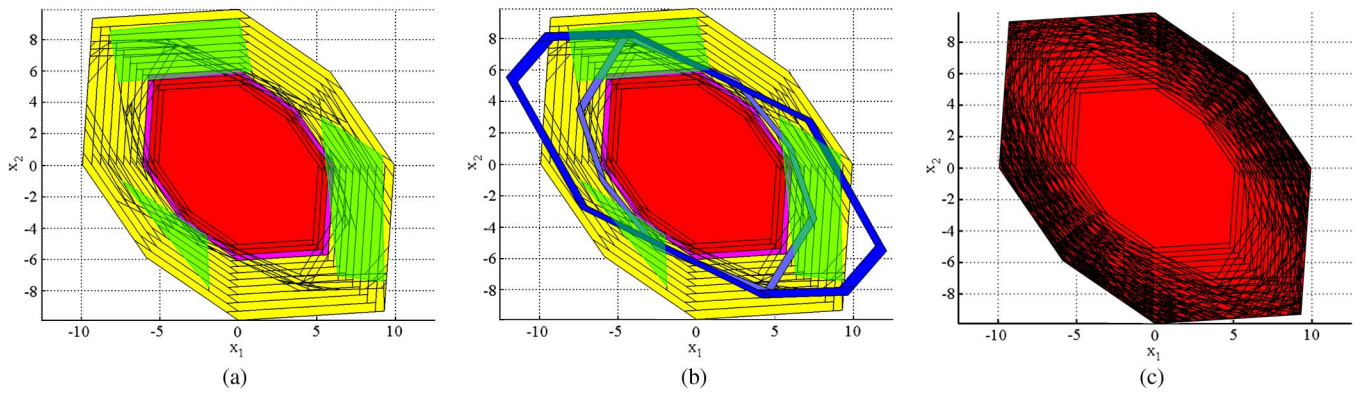

Fig. 2. The observed regions are shown in transparent green in (a) and (b). (a) At the end of the third iteration ($i = 2$), the bisimulation quotient for states within $\mathcal{P}_{\Gamma_3}$ is completed, which are shown in red $\mathcal{P}_{\Gamma_2}$) and purple ($\mathcal{S}_3$). (b) In the forth iteration, the states within $\mathcal{P}_{\Gamma_{11}} \setminus \mathcal{P}_{\Gamma_3}$ are partitioned according to $\mathrm{Pre}_{\mathcal{T}_e}(\tilde{\mathcal{P}}, \sigma), \tilde{\mathcal{P}} \in \mathcal{S}_3$. $\mathcal{S}_3$ is shown in purple; $\mathrm{Pre}_{\mathcal{T}_e}(\mathcal{S}_3, 1)$ and $\mathrm{Pre}_{\mathcal{T}_e}(\mathcal{S}_3, 2)$ are shown in light and dark blue. (c) At the last iteration where $i = 10$, the algorithm is completed. The state space covered by the bisimulation quotient is shown in red, covering all of $\mathcal{X}$.

The correctness of Algorithm 2 will be shown by an inductive argument. Given a sublevel set $\mathcal{P}_{\Gamma_i}$ and a partition P_i as obtained in Algorithm 2, we define $\tilde{P}_i$ as

$$\tilde{P}_i := \{\tilde{\mathcal{P}} \in P_i | \tilde{\mathcal{P}} \subseteq \mathcal{P}_{\Gamma_i}\}. \tag{13}$$

From Algorithm 2, we see that P_0 partitions all the slices, and since P_i is a finite refinement of P_0, we can directly see that $\tilde{P}_i$ is a partition of $\mathcal{P}_{\Gamma_i}$. Let us define an embedding transition system $\mathcal{T}_e(i)$ as a subset of $\mathcal{T}_e$ with set of states $\{x \in Q_e \mid x \in \mathcal{P}_{\Gamma_i}\}$ and let us state the following result.

Proposition 4.2: At the completion of the i-th iteration (of the outer loop) of Algorithm 2 (where P_{i+1} is obtained), if $\sim_i$ induced by $\tilde{P}_i$ as defined in (13) is a bisimulation of $\mathcal{T}_e(i)$, then $\sim_{i+1}$ induced by $\tilde{P}_{i+1}$ is a bisimulation of $\mathcal{T}_e(i+1)$.

Proof: We show that at the end of i-th iteration, each transition originating at a state $q \in Q_e/_{\sim_{i+1}}$ with $\mathrm{eq}(q) \subseteq \mathcal{P}_{\Gamma_{i+1}}$ satisfies the bisimulation requirement (Definition 2.5). By Proposition 4.1, for each $x \in \mathcal{S}_{i+1}$ and $\sigma \in \Sigma$, $x' = A_\sigma x$ must be in a slice with a lower index and thus $x' \in \mathcal{T}_e(i)$. Let $x \in \mathrm{eq}(q) \in P_i$. If $x' \in \mathcal{S}_i$, then we have $x \in \tilde{\mathcal{P}} = \texttt{ComputePre}(\mathrm{eq}(q'), \sigma)$ (from step 9 of Algorithm 2) for some $q' \in Q_e/_{\sim_i}$. The `RefineUpdate` procedure replaces $\mathrm{eq}(q)$ with $\mathrm{eq}(q_1) = \mathrm{eq}(q) \cap \tilde{\mathcal{P}}$ and $\mathrm{eq}(q_2) = \mathrm{eq}(q) \setminus \tilde{\mathcal{P}}$, and updates $\mathcal{T}_e/_{\sim_{i+1}}$. We note from the definition of Pre operator (1) that for any $x \in \mathrm{eq}(q_1)$, $x' = A_\sigma x \in \mathrm{eq}(q')$, thus for any $x_1, x_2 \in \mathrm{eq}(q_1)$, $x_1 \sim x_2$, $A_\sigma x_1 \sim A_\sigma x_2$. Moreover, the same argument holds for any subset of $\mathrm{eq}(q_1)$. Therefore, the transitions given in steps 5 and 6 of Algorithm 1 satisfy the bisimulation requirement. On the other hand, if $x' \notin S_i$, then $x' \in S_j$ for some $j < i$ and x is already in a set $\mathrm{eq}(q)$, where $q \xrightarrow{\sigma}_{\sim_{i+1}} q'$ for some q' satisfying the bisimulation requirement. Therefore, step 9 of Algorithm 2 provides exactly the transitions needed for all states in $\mathcal{S}_{i+1}$ and thus, $\sim_{i+1}$ induced by $\tilde{P}_{i+1}$ is a bisimulation of $\mathcal{T}_e(i+1)$. ■

Theorem 4.1: Algorithm 2 returns a solution to Problem 3.1.

Proof: From Algorithm 1, we have that P_i is a refinement of $P_\mathcal{X}$ for any $i = 0, \ldots, N$. Therefore, P_N and its induced relation $\sim_N$ are observation preserving.

At step 4 of Algorithm 2, we set $q \xrightarrow{\sigma}_e Q_{\sim_0} q$, $\forall\, \sigma \in \Sigma$ where $\mathrm{eq}(q) = \mathcal{D}$. From the definition of $\mathcal{T}_e$, we see that since $\mathcal{D}$ is the only state, $\sim_0$ induced by $\tilde{P}_0$ is a bisimulation of $\mathcal{T}_e(0)$. Using Proposition 4.2 and induction, at iteration $N-1$, we have that $\sim_N$ induced by $\tilde{P}_N$ is a bisimulation of $\mathcal{T}_e(N)$. Note that $\tilde{P}_N$ is exactly P_N, $\mathcal{P}_{\Gamma_N}$ is exactly $\mathcal{X}$ and $\mathcal{T}_e(N)$ is exactly $\mathcal{T}_e$. Therefore $\sim_N$ induced by P_N is a bisimulation of $\mathcal{T}_e$. ■

At each iteration i, the number of updated sets is finite as the partition P_i and the set of inputs Σ are finite, and therefore, we have:

Corollary 4.1: A solution to Problem 3.1 can be generated in a finite number of steps, which is determined by the contraction rate of the Lyapunov function.

Example 4.2 (Example 4.1 Continued): Algorithm 2 is applied on the same setting as in Example 4.1 to compute the bisimulation quotient. P_3 and P_{11} are shown in Fig. 2.

A. Extensions

Although the focus of the paper is on switched linear systems with polyhedral Lyapunov functions, the presented approach can also be applied to other classes of discrete-time systems with different Lyapunov functions, if

1) the sublevel sets of the Lyapunov function are semi-linear sets,
2) the pre-image of a bounded semi-linear set is computable and is also a semi-linear set, and
3) the dynamical system has a finite set of controls.

The first condition guarantees that the slices (see (10)) are semi-linear sets, and therefore the initial partition is composed of semi-linear sets. The second condition allows us to compute pre-images throughout the algorithm. Finally the last condition is necessary since the pre-images of the partition elements are computed for each control input (line 8 of Algorithm 2).

For example, consider discrete-time piecewise linear systems described by

$$x_{k+1} = A_\sigma x_k, \qquad x_k \in \mathcal{X}_\sigma, \sigma \in \Sigma, k \in \mathbb{Z}_+ \tag{14}$$

where Σ is a finite index set of modes (different dynamics), $P_{pwl} = \{\mathcal{X}_\sigma\}_{\sigma \in \Sigma}$ is a partition of $\mathcal{X}$ and each $\mathcal{X}_\sigma$ is a semi-linear set. Under certain conditions, a Lyapunov function with piecewise polytopic sublevel sets for system (14) exists [28]. System (14) with a piecewise polyhedral Lyapunov function satisfies the properties given above. The extension to such

systems requires to refine the initial partition P_0 according to P_{pwl}. Then, the proposed algorithms can be used to construct a quotient transition system $\mathcal{T}_e/_\sim$. In this case, in step 2 of Algorithm 1 it is sufficient to refine a state q' only if $\mathrm{eq}(q') \subset \mathcal{X}_\sigma$ since only mode σ can be active in $\mathrm{eq}(q')$. By eliminating some of the transitions of $\mathcal{T}_e/_\sim$ according to P_{pwl}, i.e., $q \xrightarrow{\sigma}_\sim q'$ only if $\mathrm{eq}(q) \subseteq \mathcal{X}_\sigma$, we obtain a bisimulation quotient $\mathcal{T}_e^{pwl}/_\sim$ for the corresponding embedding transition system. This extension is illustrated by an example in Section VII.

V. Complexity

Our algorithm involves computations of pre-images of bounded semi-linear sets through linear dynamics, intersections and set differences for semi-linear sets at each iteration. The number of operations performed, and hence the complexity of the algorithm, scale linearly with the size of the resulting partition $|P_N|$. Therefore, we derive an upper bound on $|P_N|$ with respect to the number of slices, observations and controls.

Let s_i be the number of sets in partition P_N that are included in slice $\mathcal{S}_i$, i.e.,

$$s_i := |\{\tilde{\mathcal{P}} \in P_N | \tilde{\mathcal{P}} \subseteq \mathcal{S}_i\}|.$$

Similarly, s_i^0 denotes the number of sets in the initial partition P_0 that are included in slice $\mathcal{S}_i$. In the subsequent analysis, r is used to denote the number of observations within $\mathcal{X} \setminus \mathcal{D}$ ($r = |R| + 1$), and e is used to denote the number of input symbols ($e = |\Sigma|$).

Lemma 5.1: The number of sets in the resulting partition P_N that are included in slice $i \geq 1$, s_i, is less than or equal to

$$\overline{s}_i = r \left(\sum_{k=0}^{i-1} s_k \right)^e. \tag{15}$$

Proof: A set $\tilde{\mathcal{P}} \in P_0$ with $\tilde{\mathcal{P}} \subseteq \mathcal{S}_i$ is partitioned only if there exists $\sigma \in \Sigma$ and $\tilde{\mathcal{P}}' \in \mathcal{S}_j$ for some $j < i$ such that $\tilde{\mathcal{P}} \cap \mathrm{Pre}(\tilde{\mathcal{P}}', \sigma) \neq \emptyset$ and $\tilde{\mathcal{P}} \setminus \mathrm{Pre}(\tilde{\mathcal{P}}', \sigma) \neq \emptyset$ (step 2 of Algorithm 1). Therefore,

(a) for any two states $q_1, q_2 \in Q_e/_\sim$ with $\mathrm{eq}(q_1), \mathrm{eq}(q_2) \subset \mathcal{S}_i$ if $h(q_1) = h(q_2)$, there exists $\sigma \in \Sigma$ such that $q_1 \xrightarrow{\sigma}_\sim q_1'$, $q_2 \xrightarrow{\sigma}_\sim q_2'$, and $q_1' \neq q_2'$.

From Proposition 4.1 and the bisimulation requirement (Definition 2.5), we have that

(b) for each $q \in Q_e/_\sim$ with $\mathrm{eq}(q) \subseteq \mathcal{S}_i$ and for each $\sigma \in \Sigma$, there exists a state q' with $\mathrm{eq}(q') \subseteq \mathcal{S}_j$ for some $j < i$ such that $q \xrightarrow{\sigma}_\sim q'$.

Given properties (a) and (b), the number of sets obtained from partitioning a set $\tilde{\mathcal{P}} \in P_0$ with $\tilde{\mathcal{P}} \subseteq \mathcal{S}_i$ is bounded by the number of permutations of size e, with unrestricted repetitions, taken from a set of size $\sum_{k=0}^{i-1} s_k$.

The given bound is obtained by observing that $s_i^0 \leq r$ for all $i = 1, \ldots, N$, since the initial partition P_0 is obtained by refining the coarsest observation preserving partition $P_\mathcal{X}$ (see (11)) according to slice partition. ■

The bound is computed through a combinatorial perspective by utilizing the contractive property of the system. Even though the bound is attainable, the underlying dynamics is not considered explicitly. Therefore, in many cases the bound is not attained, for example see Example 5.1.

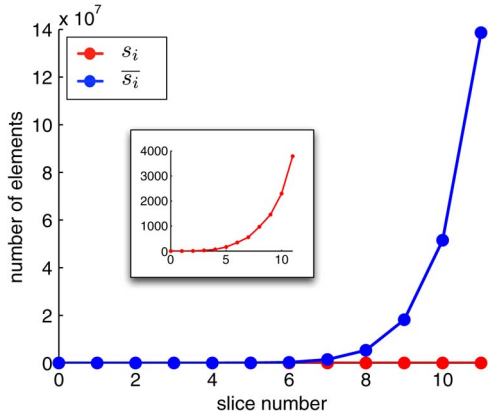

Fig. 3. Comparison of the number of elements in a slice, s_i, and the corresponding bound, $\overline{s_i}$ (15).

Remark 5.1: P_N is the coarsest refinement of P_0 satisfying the bisimulation requirement. This claim follows from statement-(a) of the proof of Lemma 5.1, and can easily be proven by an inductive argument on the partitions of the sublevel sets, i.e., $\tilde{P}_i$ as defined in (13).

Example 5.1 (Example 4.2 Continued): The number of sets in partition P_{11} according to slice numbers and the corresponding bounds computed as in (15) are shown in Fig. 3. ■

By using Lemma 5.1, we derive a bound on the number of sets in P_N that are included in sublevel set $\mathcal{P}_{\Gamma_i}$ in closed form, i.e., the new bound depends only on sublevel set number i, the size of the control set e and the number of observations r.

Theorem 5.1: Let $p_i = |\{\tilde{\mathcal{P}} \,|\, \tilde{\mathcal{P}} \in P_N, \tilde{\mathcal{P}} \subseteq \mathcal{P}_{\Gamma_i}\}|$ for all $i = 0, \ldots, N$. Then $p_0 = 1$ and

$$p_i \leq (r+1)^{\sum_{j=0}^{i-1} e^j}, \qquad i = 1, \ldots, N. \tag{16}$$

Proof: As $\mathcal{P}_{\Gamma_0}$ is not partitioned, the claim holds for $i = 0$, i.e., $p_0 = 1$. We prove the claim for $i \geq 1$ by induction. The definitions of p_i, s_i, and $\overline{s_i}$ imply that

$$p_{i+1} = p_i + s_{i+1} \leq p_i + \overline{s_{i+1}}.$$

From (15) $\overline{s}_1 = r$, and hence the claim holds for $i = 1$ as $p_1 \leq 1 + r$. Assume that inequality (16) holds for p_k for some $k \geq 1$. By Lemma 5.1, we have that $\overline{s_{k+1}} = r p_k^e$. Therefore

$$p_{k+1} \leq p_k + r p_k^e < (r+1) p_k^e.$$

Using the inductive hypothesis on p_k, the left hand side can be rewritten as

$$\begin{aligned} p_{k+1} &< (r+1) \left((r+1)^{\sum_{j=0}^{k-1} e^j} \right)^e \\ p_{k+1} &< (r+1) \left((r+1)^{\sum_{j=1}^{k} e^j} \right) \\ p_{k+1} &< (r+1)^{\sum_{j=0}^{k} e^j}. \end{aligned}$$

Thus, inequality (16) holds for p_{k+1}, and by induction we conclude that (16) holds for all $i = 1, \ldots, N$. ■

The size, p_N, of the resulting partition of the working set $\mathcal{X}$ is double exponential with N when $e > 1$ (switched systems). Therefore when $e > 1$ the number of Pre operations performed, p_{N-1}, is double exponential with $N-1$. It is easy to verify from (16) that the bound is exponential with N for linear systems, i.e., $e = 1$. Note that the derived bound is an upper bound for the worst case, i.e., when $\overline{s}_i = s_i$ for all $i = 0, \ldots, N$.

Remark 5.2: The computational complexity increases with the number of sublevel sets, N, which is computed from the working set $\mathcal{X}$, the target set $\mathcal{D}$, and the contraction rate ρ of the Lyapunov function (see (8)). Therefore, the amount of computation can be adjusted by the choice of the working set $\mathcal{X}$ and the target set $\mathcal{D}$ for a given Lyapunov function. For example, the computation time can be decreased by choosing $\mathcal{D}$ as the largest sublevel set that does not intersect with the regions of observations. In addition, using a Lyapunov function with a minimal contraction rate can decrease the computation time. Computation of such Lyapunov functions are out of the scope of this paper, but for example, minimization of the contraction rate is possible via the algorithm presented in [28].

VI. Temporal Logic Synthesis and Verification

After we obtain a bisimulation quotient for system (6), we can solve verification and controller synthesis problems from temporal logic specifications such as CTL^*, CTL, and LTL. The asymptotic stability assumption implies that all trajectories of (6) sink in $\mathcal{D}$. For this reason, we will focus on the syntactically co-safe fragment of LTL, which includes all specifications of LTL where satisfactions of trajectories can be determined by a finite prefix. Since we are interested in the behavior of (6) until $\mathcal{D}$ is reached, scLTL is a sufficiently rich specification language.

A detailed description of the syntax and semantics of scLTL is beyond the scope of this paper and can be found in, for example, [7], [29]. Roughly, an scLTL formula is built up from a set of atomic propositions Π, standard Boolean operators $\neg$ (negation), $\vee$ (disjunction), $\wedge$ (conjunction), $\Rightarrow$ (implication) and temporal operators X (next), U (until), and F (eventually).[1] The semantics of scLTL formulae is given over infinite words $\mathbf{o} = o_0 o_1 \ldots$, where $o_i \in 2^{\Pi}$ for all i. We write $\mathbf{o} \models \phi$ if the word $\mathbf{o}$ satisfies the scLTL formula ϕ. We say a trajectory $\mathbf{q}$ of a transition system $\mathcal{T}$ satisfies scLTL formula ϕ, if the word generated by $\mathbf{q}$ (see Definition 2.1) satisfies ϕ.

Example 6.1: Again, consider the setting in Example 3.1 with $\mathcal{R} = \{\mathcal{R}_i\}_{i=\{1,2,3\}}$. We now consider a specification in scLTL over $\{\mathcal{R}_1, \mathcal{R}_2, \mathcal{R}_3, \Pi_{\mathcal{D}}\}$. For example, the specification "*A system trajectory never visits $\mathcal{R}_2$ and eventually visits $\mathcal{R}_1$. Moreover, if it visits $\mathcal{R}_3$ then it must not visit $\mathcal{R}_1$ at the next time step*" can be translated to a scLTL formula:

$$\phi := (\neg\mathcal{R}_2 \cup \Pi_{\mathcal{D}}) \wedge \mathrm{F}\mathcal{R}_1 \wedge ((\mathcal{R}_3 \Rightarrow \mathrm{X}\neg\mathcal{R}_1) \cup \Pi_{\mathcal{D}}). \quad (17)$$

A. Synthesis of Switching Strategies

In this section, we assume that we can choose the dynamics A_σ, $\sigma \in \Sigma$ to be applied at each step k. Our goal is to find the largest set of initial states and a switching sequence (i.e., a sequence of elements from Σ to be applied at each step) for each initial state such that all the corresponding trajectories of system (6) satisfy a temporal logic specification. As a switched system is deterministic, it produces a unique trajectory for a given initial state and a switching sequence. Formally, we consider the following problem:

Problem 6.1: Consider system (6) with a polyhedral Lyapunov function in the form of (5), sets $\mathcal{X}$, $\mathcal{D}$, and $\{\mathcal{R}_i\}_{i\in R}$, and a scLTL formula ϕ over $R \cup \{\Pi_{\mathcal{D}}\}$. Find the largest set $\mathcal{X}^S \subseteq \mathcal{X}$ and a function $\Omega : \mathcal{X}^S \mapsto \Sigma^*$ such that the trajectory of system (6) initiated from a state $x_0 \in \mathcal{X}^S$ under the switching sequence $\Omega(x_0)$ satisfies ϕ.

Our solution to Problem 6.1 proceeds by finding a bisimulation quotient $\mathcal{T}_e/_\sim$ of the embedding transition system $\mathcal{T}_e$ using Algorithm 2. Then we translate ϕ to a Finite State Automaton (FSA), defined below.

Definition 6.1: A deterministic finite state automaton (FSA) is a tuple $\mathcal{A} = (S_{\mathcal{A}}, S_{\mathcal{A}0}, \Sigma, \delta_{\mathcal{A}}, F_{\mathcal{A}})$ where

- $S_{\mathcal{A}}$ is a finite set of states;
- $S_{\mathcal{A}0} \subseteq S_{\mathcal{A}}$ is a set of initial states;
- Σ is an input alpabet;
- $\delta_{\mathcal{A}} : S_{\mathcal{A}} \times \Sigma \to S_{\mathcal{A}}$ is a transition function;
- $F_{\mathcal{A}} \subseteq S_{\mathcal{A}}$ is a set of final states.

A word $\boldsymbol{\sigma} = \sigma_0 \ldots \sigma_{d-1}$ over Σ generates a trajectory $s_0 \ldots s_d$, where $s_0 \in S_{\mathcal{A}0}$ and $\delta(s_i, \sigma_i) = s_{i+1}$ for all $i = 0, \ldots, d-1$. $\mathcal{A}$ accepts word $\boldsymbol{\sigma}$ if $s_d \in F_{\mathcal{A}}$.

For any scLTL formula ϕ over Π, there exists a FSA $\mathcal{A}$ with input alphabet 2^{Π} that accepts the prefixes of all and only the satisfying words [7], [30].

Definition 6.2: Given a transition system $\mathcal{T} = (Q, \Sigma, \to, \Pi, h)$ and a FSA $\mathcal{A} = (S_{\mathcal{A}}, S_{\mathcal{A}0}, 2^{\Pi}, \delta_{\mathcal{A}}, F_{\mathcal{A}})$, their product automaton, denoted by $\mathcal{PA} = \mathcal{T} \times \mathcal{A}$, is a tuple $\mathcal{PA} = (S_{\mathcal{PA}}, S_{\mathcal{PA}0}, \Sigma, \to_{\mathcal{PA}}, F_{\mathcal{PA}})$ where

- $S_{\mathcal{PA}} = Q \times S_{\mathcal{A}}$;
- $S_{\mathcal{PA}0} = Q \times S_{\mathcal{A}0}$;
- $\to_{\mathcal{PA}} \subseteq S_{\mathcal{PA}} \times \Sigma \times S_{\mathcal{PA}}$ is the set of transitions, defined by: $((q,s),\sigma,(q',s')) \in \to_{\mathcal{PA}}$ iff $q \xrightarrow{\sigma} q'$ and $\delta_{\mathcal{A}}(s, h(q)) = s'$;
- $F_{\mathcal{PA}} = Q \times F_{\mathcal{A}}$.

We denote $s_{\mathcal{PA}} \xrightarrow{\sigma}_{\mathcal{PA}} s'_{\mathcal{PA}}$ if $(s_{\mathcal{PA}}, \sigma, s'_{\mathcal{PA}}) \in \to_{\mathcal{PA}}$. A trajectory $\mathbf{p} = (q_0, s_0) \ldots (q_d, s_d)$ of $\mathcal{PA}$ produced by input word $\boldsymbol{\sigma} = \sigma_0 \ldots \sigma_{d-1}$ is a finite sequence such that $(q_0, s_0) \in S_{\mathcal{PA}0}$ and $(q_k, s_k) \xrightarrow{\sigma_k}_{\mathcal{PA}} (q_{k+1}, s_{k+1})$ for all $k = 0, \ldots, d-1$. $\mathbf{p}$ is called accepting if $(q_d, s_d) \in F_{\mathcal{PA}}$.

By the construction of $\mathcal{PA}$ from $\mathcal{T}$ and $\mathcal{A}$, $\mathbf{p}$ produced by $\boldsymbol{\sigma}$ is accepting if and only if $\mathbf{q} = \gamma_{\mathcal{T}}(\mathbf{p})$ satisfies the scLTL formula corresponding to $\mathcal{A}$ [29], where $\gamma_{\mathcal{T}}(\mathbf{p})$ is the projection of a trajectory $\mathbf{p}$ of $\mathcal{PA}$ onto $\mathcal{T}$ by simply removing the automaton part of the state in $s_{\mathcal{PA}} \in S_{\mathcal{PA}}$.

We construct the product $\mathcal{PA}$ between the quotient transition system $\mathcal{T}_e/_\sim$ obtained from Algorithm 2 and FSA $\mathcal{A}$ corresponding to specification formula ϕ. By performing a graph search on $\mathcal{PA}$, we can find the largest subset $S^S_{\mathcal{PA}}$ of $S_{\mathcal{PA}}$ and a feedback control function $\Omega_{\mathcal{PA}} : S^S_{\mathcal{PA}} \mapsto \Sigma$ such that the trajectories of $\mathcal{PA}$ originating in $S^S_{\mathcal{PA}}$ in closed loop with $\Omega_{\mathcal{PA}}$

[1]The difference between LTL and scLTL is the lack of globally operator in scLTL. Moreover, the negation can only be used in conjunction with the propositions in a scLTL formula.

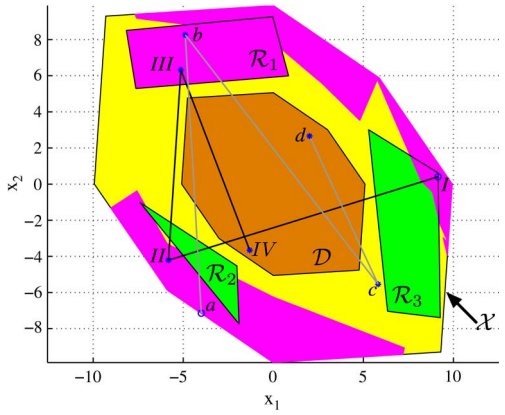

Fig. 4. $\mathcal{X}^S$ is shown in purple. $\mathcal{X}$, $\mathcal{D}$, $\{\mathcal{R}_i\}_{i\in R}$ and two sample trajectories are indicated by their labels.

reach $F_{\mathcal{PA}}$. Then, we define the set of satisfying initial states of system (6) from $S^S_{\mathcal{PA}}$ as

$$\mathcal{X}^S = \{\mathrm{eq}(q) | (q,s) \in (S_{\mathcal{PA}0} \cap S^S_{\mathcal{PA}})\}. \tag{18}$$

Since $\mathcal{PA}$ is deterministic, $\Omega_{\mathcal{PA}}$ defines a unique input word for each $(q_0,s_0) \in S^S_{\mathcal{PA}}$. Moreover, an input word of $\mathcal{PA}$ directly maps to a switching sequence for system (6). Formally, the switching sequence $\Omega : \mathcal{X}^S \mapsto \Sigma^*$ is obtained by "projecting" $\Omega_{\mathcal{PA}}$ from $\mathcal{PA}$ to $\mathcal{T}$ as follows:

$$\Omega(x) = \Omega_{\mathcal{PA}}((q_0,s_0)) \ldots \Omega_{\mathcal{PA}}((q_{d-1},s_{d-1})) \tag{19}$$

where $x \in eq(q_0)$, $s_0 \in S_{\mathcal{A}0}$, $(q_i,s_i) \xrightarrow{\Omega_{\mathcal{PA}}((q_i,s_i))}_{\mathcal{PA}} (q_{i+1},s_{i+1})$, for each $i = 0,\ldots,d-1$ and $(q_d,s_d) \in F_{\mathcal{PA}}$.

Proposition 6.1: $\mathcal{X}^S$ as defined in (18) and function Ω as defined in (19) solve Problem 6.1.

Proof: For each $x \in \mathcal{X}^S$, there exists $(q_0,s_0) \in S^S_{\mathcal{PA}}$ such that $x \in \mathrm{eq}(q_0)$ and $s_0 \in S_{\mathcal{A}0}$ by (18). By the construction of $\mathcal{PA}$ and the definition of Ω (19), the trajectory of $\mathcal{T}_e/_\sim$ originating at q_0 and generated by input word $\Omega(x)$ satisfies ϕ. Then by the bisimulation relation the trajectories of (6) originating in $\mathrm{eq}(q_0)$ and generated by switching sequence $\Omega(x)$ satisfy ϕ.

We prove that $\mathcal{X}^S$ is the largest set of satisfying initial states by contradiction. Assume that there exists $x_0 \notin \mathcal{X}^S$ such that a trajectory $\mathbf{x} = x_0 \ldots x_d$ originating at x_0 of system (6) produced by switching sequence $\boldsymbol{\sigma} = \sigma_0 \ldots \sigma_{d-1}$ satisfies ϕ, and $x_0 \in \mathrm{eq}(q_0)$ where $q_0 \in Q_e/_\sim$. Then by the bisimulation relation 1) there exists a trajectory $\mathbf{q} = q_0 \ldots q_d$ of $\mathcal{T}_e/_\sim$ such that $x_i \in \mathrm{eq}(q_i)$, $q_i \xrightarrow{\sigma_i}_e /_\sim q_{i+1}$ for all $i = 0,\ldots,d-1$ and $x_d \in \mathrm{eq}(q_d)$, 2) $\mathbf{q}$ satisfies ϕ. However, we know that on the product $\mathcal{PA} = \mathcal{T}_e/_\sim \times \mathcal{A}$, $F_{\mathcal{PA}}$ is not reachable from $\{(q_0,s) \mid s \in S_{\mathcal{A}0}\}$. Hence, a trajectory $\mathbf{p}$ originating in $\{(q_0,s) \mid s \in S_{\mathcal{A}0}\}$ cannot be accepting on $\mathcal{PA}$, and by construction of $\mathcal{PA}$ [29] $\gamma_{\mathcal{T}_e/_\sim}(\mathbf{p})$ as a trajectory of $\mathcal{T}_e/_\sim$ cannot satisfy formula ϕ, which yields a contradiction. ■

Example 6.2 (Example 6.1 Continued): For the example specification ϕ (17), we obtained the solution to Problem 6.1. The FSA has 6 states and the quotient TS obtained from Algorithm 2 has 9677 states. The set of initial states $\mathcal{X}^S$ is shown in Fig. 4.

B. Verification Under Arbitrary Switching

In this section, we consider the problem of verifying system (6) under arbitrary switching, i.e., at every time-step a subsystem is arbitrarily chosen from the set Σ.

Problem 6.2: Consider system (6) with a polyhedral Lyapunov function in the form of (5), sets $\mathcal{X}$, $\mathcal{D}$, and $\{\mathcal{R}_i\}_{i\in R}$, and a scLTL formula ϕ over $R \cup \{\Pi_{\mathcal{D}}\}$. Find the largest set $\mathcal{X}^{AS} \subseteq \mathcal{X}$ such that all trajectories of system (6) originating in $\mathcal{X}^{AS}$ satisfy ϕ under arbitrary switching.

Note that system (6) under arbitrary switching is uncontrolled and non-deterministic. Therefore, we define an embedding transition system $\mathcal{T}^A_e = \{Q_e, \Sigma^A, \rightarrow^A_e, h_e\}$ for the arbitrary switching setup from the embedding transition system $\mathcal{T}_e = \{Q_e, \Sigma, \rightarrow_e, h_e\}$ (Definition 3.1) by adapting the input set and the set of transitions as follows:

- $\Sigma^A = \{\epsilon\}$,
- $\rightarrow^A_e = \{(q,\epsilon,q') \mid \exists \sigma \in \Sigma, (q,\sigma,q') \in \rightarrow_e\}$.

We denote $q \rightarrow^A_e q'$ if $(q,\epsilon,q') \in \rightarrow^A_e$. We use ϵ as a "dummy" input because the transitions of $\mathcal{T}^A_e$ are not controlled. Note that $\mathcal{T}^A_e$ is infinite and non-deterministic. Moreover, $\mathcal{T}^A_e$ exactly captures dynamics of system (6) under arbitrary switching in the relevant state space $\mathcal{X} \setminus \mathcal{D}$.

Our solution to Problem 6.2 parallels the solution we proposed for Problem 6.1. We first convert the bisimulation quotient $\mathcal{T}_e/_\sim = \{Q_e/_\sim, \Sigma, \rightarrow_e /_\sim, h_e/_\sim\}$ of $\mathcal{T}_e$ obtained from Algorithm 2 to $\mathcal{T}^A_e/_\sim = \{Q_e/_\sim, \Sigma^A, \rightarrow^A_e /_\sim, h_e/_\sim\}$ as follows:

- $\Sigma^A = \{\epsilon\}$,
- $\rightarrow^A_e /_\sim = \{(q,\epsilon,q') \mid \exists \sigma \in \Sigma, (q,\sigma,q') \in \rightarrow_e /_\sim\}$.

In this case, we have a particular bisimulation relation. The embedding and the quotient transition systems have a single input that labels all the transitions.

Proposition 6.2: $\mathcal{T}^A_e/_\sim$ is a bisimulation quotient of $\mathcal{T}^A_e$.

Proof: Let $q_1, q_2 \in \mathrm{eq}(q)$, $q'_1 \in \mathrm{eq}(q')$, and $q_1 \rightarrow^A_e q'_1$, where $q, q' \in Q_e/_\sim$, and $q_1, q_2, q'_1 \in Q_e$. To prove the bisimulation property we need to show that there exists $q'_2 \in \mathrm{eq}(q')$ such that $q_2 \rightarrow^A_e q'_2$.

If $q_1 \rightarrow^A_e q'_1$, then there exists $\sigma \in \Sigma$ such that $q_1 \xrightarrow{\sigma}_e q'_1$, i.e., $q'_1 = A_\sigma q_1$. Steps 9 and 10 of Algorithm 2 guarantee that $\mathrm{eq}(q) \subseteq \mathrm{Pre}_{\mathcal{T}_e}(\mathrm{eq}(q'),\sigma)$. Therefore, for all $q_i \in \mathrm{eq}(q)$, $A_\sigma q_i \in \mathrm{eq}(q')$, and hence for all $q_i \in \mathrm{eq}(q)$, $q_i \rightarrow^A_e q_j$ for some $q_j \in \mathrm{eq}(q')$. ■

Parallel to our solution to Problem 6.1, we construct a FSA $\mathcal{A}$ corresponding to specification formula ϕ, and then we take the product $\mathcal{PA}^A = (S^A_{\mathcal{PA}}, S^A_{\mathcal{PA}0}, \Sigma^A, \rightarrow^A_{\mathcal{PA}}, F^A_{\mathcal{PA}})$ between $\mathcal{T}^A_e/_\sim$ and $\mathcal{A}$ as described in Definition 6.2. Note that $\mathcal{PA}^A$ is non-deterministic as $\mathcal{T}^A_e/_\sim$ is non-deterministic.

To finally solve Problem 6.2, we employ the approach proposed in [31], [32] to solve reachability problems on non-deterministic finite systems. We define an operator J_i:

$$J_{i+1}(s_{\mathcal{PA}}) = \min(J_i(s_{\mathcal{PA}}), \max_{s_{\mathcal{PA}} \rightarrow^A_{\mathcal{PA}} s'_{\mathcal{PA}}} J_i(s'_{\mathcal{PA}}) + 1)$$

initialized with $J_0(s_{\mathcal{PA}}) = \infty$ for all $s_{\mathcal{PA}} \in S^A_{\mathcal{PA}} \setminus F^A_{\mathcal{PA}}$ and $J_0(s_{\mathcal{PA}}) = 0$ for all $s_{\mathcal{PA}} \in F^A_{\mathcal{PA}}$. We iteratively compute J_{i+1} from J_i until the fixed point of the operator is reached, i.e., $J_i(s_{\mathcal{PA}}) = J_{i+1}(s_{\mathcal{PA}})$ for all $s_{\mathcal{PA}} \in S^A_{\mathcal{PA}}$. The fixed point of

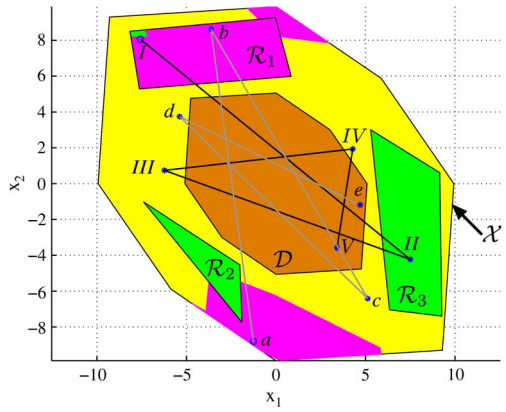

Fig. 5. $\mathcal{X}^{AS}$ is shown in purple. $\mathcal{X}$, $\mathcal{D}$, $\{\mathcal{R}_i\}_{i\in R}$ and two sample trajectories are indicated by labeling.

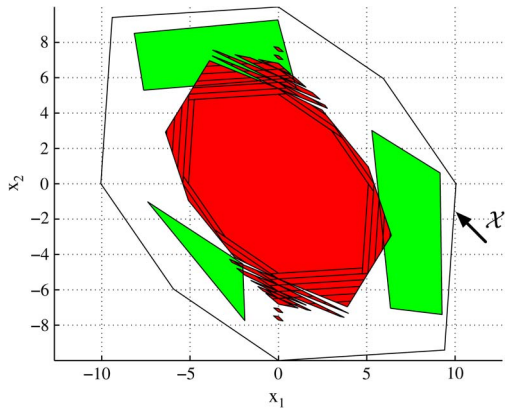

Fig. 6. $\cup_{q\in Q_e^\forall}\mathrm{eq}(q)$ is shown in red.

the operator always exists, since the number of states, $|S^A_{\mathcal{PA}}|$, is finite.

Proposition 6.3: Let $S^{AS}_{\mathcal{PA}} = \{s_{\mathcal{PA}} \in S^A_{\mathcal{PA}} \mid J(s_{\mathcal{PA}}) < \infty\}$ and define $\mathcal{X}^{AS} = \{\mathrm{eq}(q) \mid (q,s) \in (S^A_{\mathcal{PA}0} \cap S^{AS}_{\mathcal{PA}})\}$. Then $\mathcal{X}^{AS}$ solves Problem 6.2.

Proof: For each $x \in \mathcal{X}^{AS}$, there exists $(q_0, s_0) \in S^{AS}_{\mathcal{PA}}$ such that $x \in \mathrm{eq}(q_0)$ and $s_0 \in S_{\mathcal{A}0}$. The fixed point computation guarantees that every trajectory of $\mathcal{PA}^A$ originating at (q_0, s_0) reaches $F^A_{\mathcal{PA}}$. Then, the construction of $\mathcal{PA}^A$ and the bisimulation relation guarantee that all of the trajectories of (6) originating in $\mathrm{eq}(q_0)$ satisfy ϕ.

If $x_0 \notin \mathcal{X}^{AS}$, we need to show that there exists a trajectory $\mathbf{x} = x_0 \dots x_d$ of (6) that violates ϕ. Let $x_0 \in \mathrm{eq}(q_0)$. If $x_0 \notin \mathcal{X}^{AS}$, then for all $s_0 \in S_{\mathcal{A}0}$ there exists a trajectory $\mathbf{p} = (q_0, s_0) \dots (q_d, s_d)$ of $\mathcal{PA}^A$ that can not reach $F^A_{\mathcal{PA}}$, otherwise (q_0, s_0) would be included in $S^{AS}_{\mathcal{PA}}$. Since $\mathbf{p}$ can not reach $F^A_{\mathcal{PA}}$, $\mathbf{q} = \gamma_{\mathcal{T}}(\mathbf{p})$ violates ϕ. By the bisimulation property, there exists a trajectory $\mathbf{x} = x_0 \dots x_d$ of (6) that produces the same word as $\mathbf{q}$, and hence $\mathbf{x}$ violates ϕ. ■

Example 6.3 (Example 6.1 Continued): For the example specification ϕ as in (17), we obtained the solution to Problem 6.2. $\mathcal{X}^{AS}$ and sample trajectories are shown in Fig. 5. Note that, by definition, this is a subset of the set of initial states found for the synthesis problem (see Fig. 4).

C. Verification for Polytopic Difference Inclusions

In this section, we consider switched systems with infinitely many subsystems. Assume that the active subsystem of system (6) is chosen from the convex set $\mathcal{A} = co\{A_\sigma \mid \sigma \in \Sigma\}$ rather than the finite set $\{A_\sigma \mid \sigma \in \Sigma\}$, and let $\mathcal{T}_e^\infty$ be the corresponding embedding transition system with input set Σ^∞. In this case Algorithm 2 cannot be used to construct a finite bisimulation quotient of the embedding transition system, since the input set Σ^∞ is not finite. The computation of a bisimulation quotient for this setting requires to compute equivalence classes also in the control space. Quantifier elimination can be used to compute pre-images and the corresponding equivalence classes in the control space. However, such approaches result in intractable implementations. Moreover, existence of finite bisimulations for such systems is not guaranteed. Here, we focus on an arbitrary switching setup and show that we can compute a deterministic bisimulation quotient for a subset of the working set $\mathcal{X}$. Let $\mathcal{T}_e^\forall = \{Q_e^\forall, \{\epsilon\}, \rightarrow_e^\forall, h_e\}$ be the embedding transition system of the switched system under arbitrary switching, i.e., $(x, \epsilon, x') \in \rightarrow_e^\forall$ if $\exists A \in \mathcal{A}$ and $x' = Ax$. $\mathcal{T}_e^\forall$ has a single input and is non-deterministic. We will construct a bisimulation quotient of a subset of $\mathcal{T}_e^\forall$.

Let $\mathcal{T}_e$ be the embedding transition system for the finite input set Σ, and $\mathcal{T}_e/_\sim = \{Q_e/_\sim, \Sigma, \rightarrow_e /_\sim, h_e/_\sim\}$ be the bisimulation quotient of $\mathcal{T}_e$ constructed using Algorithm 2. We define $\mathcal{T}_e^\forall/_\sim = \{Q_e^\forall/_\sim, \{\epsilon\}, \rightarrow_e^\forall /_\sim, h_e/_\sim\}$ from $\mathcal{T}_e/_\sim$ as follows:

- $\rightarrow_e^\forall /_\sim = \{(q, \epsilon, q') \mid \forall \sigma \in \Sigma, (q, \sigma, q') \in \rightarrow_e /_\sim\}$,
- $Q_e^\forall/_\sim = \{q_0 \mid q_0 \in Q_e/_\sim, \exists d \in \mathbb{Z}_+, \forall i = 0, \dots, d-1, (q_i, \epsilon, q_{i+1}) \in \rightarrow_e^\forall /_\sim, \mathcal{D} = \mathrm{eq}(q_d)\}$.

There is a transition $(q, \epsilon, q') \in \rightarrow_e^\forall /_\sim$, if q has a unique successor in $\mathcal{T}_e/_\sim$, i.e. $|\{q' \mid \exists \sigma \in \Sigma, (q, \sigma, q') \in \rightarrow_e /_\sim\}| = 1$. $Q_e^\forall/_\sim$ is a subset of $Q_e/_\sim$, and for each $q_0 \in Q_e^\forall/_\sim$, $\mathcal{T}_e/_\sim$ produces the same trajectory $q_0 q_1 \dots$ for any input sequence. In other words, the behavior of $\mathcal{T}_e/_\sim$ within $Q_e^\forall/_\sim$ is predictable under arbitrary switching. By construction $\mathcal{T}_e^\forall/_\sim$ is deterministic. In addition, each state $q \in Q_e^\forall/_\sim$ has a single outgoing transition (q, ϵ, q') satisfying that

$$\mathrm{eq}(q) \subseteq \bigcup_{\sigma\in\Sigma} \mathrm{Pre}(\mathrm{eq}(q'), \sigma). \tag{20}$$

Moreover, by using the convexity of set $\mathcal{A}$ ($\mathcal{A} = co\{A_\sigma \mid \sigma \in \Sigma\}$), it can be shown that for a set $\mathcal{P}$:

$$\bigcap_{\sigma\in\Sigma} \mathrm{Pre}(\mathcal{P}, \sigma) = \bigcap_{\sigma\in\Sigma^\infty} \mathrm{Pre}(\mathcal{P}, \sigma). \tag{21}$$

Finally, from (20) and (21), we conclude that all the transitions of $\mathcal{T}_e^\forall/_\sim$ satisfy the bisimulation requirement and $\mathcal{T}_e^\forall/_\sim$ is a bisimulation quotient of $\mathcal{T}_e^\forall$ for the states within $\cup_{q\in Q_e^\forall}\mathrm{eq}(q)$.

Example 6.4: For the setting in Example 3.1, we obtained $\mathcal{T}_e^\forall/_\sim$ from $\mathcal{T}_e/_\sim$. $\cup_{q\in Q_e^\forall}\mathrm{eq}(q)$ is shown in Fig. 6.

VII. Implementation and Case Studies

The methods described in this paper were implemented in MATLAB as a software package, which is freely downloadable from hyness.bu.edu/Software.html. The examples presented above and the following case studies were generated by using

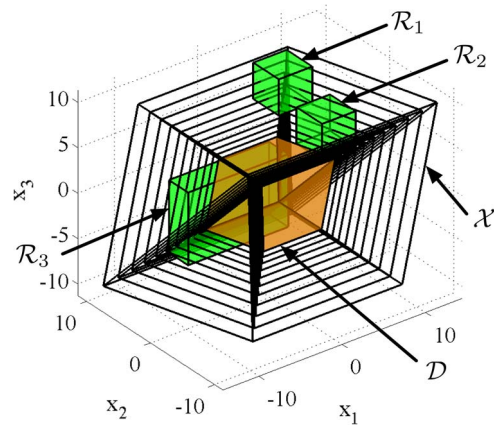

Fig. 7. The sets $\mathcal{X}$, $\mathcal{D}$, $\mathcal{R}_1$, $\mathcal{R}_2$, $\mathcal{R}_3$, and the sublevel sets $\{\mathcal{P}_{\Gamma_i}\}_{i=0,\ldots,10}$ of the Lyapunov function for Case Study 1.

the software package on an iMac with an Intel Core i5 processor at 2.8 GHz with 8 GB of memory. Algorithm 2 was completed in 2 hours for Example 3.1. Once the bisimulation quotient was constructed, controller synthesis and verification were both completed in 2 minutes.

A complete case study of a switched linear system in $\mathbb{R}^2$ was presented as a running example throughout the paper. Here, two additional case studies are presented: a linear system in $\mathbb{R}^3$ and a piecewise linear system in $\mathbb{R}^2$.

A. Case Study 1

In this case study, we apply the proposed methods to a 3-dimensional discrete-time linear system:

$$x_{k+1} = A_1 x_k, \text{where } A_1 = \begin{bmatrix} 0.384 & 0.394 & 0.240 \\ 0 & 0.442 & -0.442 \\ -0.100 & 0 & 0.780 \end{bmatrix}. \quad (22)$$

The system is of the form (6) with $\Sigma = \{1\}$. Note that the system is autonomous, i.e., there is only one dynamics and therefore no control input. The system is asymptotically stable and $\|Lx\|_\infty$ is a Lyapunov function for the system with contraction rate $\rho = 0.91$, where

$$L = \begin{bmatrix} -0.8835 & 0.1165 & 0.1165 \\ -0.8835 & -0.1165 & -0.1165 \\ 0.8835 & -0.1165 & -0.1165 \\ 0.8835 & 0.1165 & 0.1165 \\ 0 & -1.0000 & -0.1165 \\ 0 & -1.0000 & -0.1165 \\ 0 & 1.0000 & 0.1165 \\ 0 & 1.0000 & 0.1165 \\ 0 & -0.1165 & -1.0000 \\ 0 & -0.1165 & -1.0000 \\ 0 & 0.1165 & 1.0000 \\ 0 & 0.1165 & 1.0000 \end{bmatrix}.$$

The working set and the target set are $\mathcal{X} := \{x \in \mathbb{R}^3 \mid \|Lx\|_\infty \leq 10.0\}$ and $\mathcal{D} = \{x \in \mathbb{R}^3 \mid \|Lx\|_\infty \leq 3.8942\}$, respectively. The sublevel sets $\{\mathcal{P}_{\Gamma_i}\}_{i=0,\ldots,10}$ are computed as explained in Section IV. These sublevel sets and the sets of observations, $\mathcal{R} = \{\mathcal{R}_1, \mathcal{R}_2, \mathcal{R}_3\}$, are shown in Fig. 7.

Algorithm 2 was used to compute a finite bisimulation quotient of the corresponding embedding transition system. The quotient TS had 14096 states and was computed in 35 minutes.

The following specification was considered: "*A system trajectory either visits $\mathcal{R}_1$ and then $\mathcal{R}_2$, or $\mathcal{R}_3$ before visiting $\mathcal{D}$.*" The specification was formally stated as the following scLTL formula:

$$\phi := (\neg \Pi_{\mathcal{D}} \cup (\mathcal{R}_1 \wedge \mathcal{R}_3) \wedge ((\neg \mathcal{R}_1 \wedge \mathrm{F}\mathcal{R}_2) \cup \Pi_{\mathcal{D}}). \quad (23)$$

The largest set of satisfying initial states of system (22) for formula Φ (23) was computed by following the methods explained in Section VI. The FSA had 5 states. Note that both the verification and synthesis problems result in the same set for this case study, i.e., $\mathcal{X}^S = \mathcal{X}^{AS}$, since system (22) is autonomous. The computation of $\mathcal{X}^S$ took 1.2 seconds. The volume of $\mathcal{X}^S$ is 17.4% of the volume of $\mathcal{X} \setminus \mathcal{D}$. The set of initial states and sample trajectories are shown in Fig. 8.

B. Case Study 2

In this case study, we show how the proposed method to construct a bisimulation quotient can be applied to a piecewise linear system. The system is adapted from [28], where stabilizing static feedback control laws for discrete-time piecewise affine (PWA) systems are synthesized. The synthesis framework involves computation of piecewise linear Lyapunov functions that admit piecewise polytopic sublevel sets. Here, we consider the stable closed-loop system which is described by (14) with:

$$\begin{aligned} A_1 = A_5 &= \begin{bmatrix} 0.0546 & -0.7764 \\ 0.0212 & -0.8521 \end{bmatrix} \\ A_2 = A_6 &= \begin{bmatrix} -0.0700 & -0.8150 \\ 0.0700 & -0.7300 \end{bmatrix} \\ A_3 = A_7 &= \begin{bmatrix} -0.9200 & -0.0200 \\ 0.7580 & -0.0200 \end{bmatrix} \\ A_4 = A_8 &= \begin{bmatrix} -0.9200 & 0.0200 \\ 0.7580 & -0.0200 \end{bmatrix}. \end{aligned} \quad (24)$$

We define the operating regions, $\{\mathcal{X}_\sigma\}_{\sigma\in\Sigma}$, and the working set, $\mathcal{X} = \cup_{\sigma\in\Sigma}\mathcal{X}_\sigma$, with respect to the conic partition of $\mathbb{R}^2$, $\{\Omega_\sigma\}_{\sigma\in\Sigma}$, used in [28] and the piecewise linear Lyapunov function of system:

$$V(x) = \|L_\sigma x\|_\infty, \quad \text{if } x \in \Omega_\sigma, \quad L_\sigma \in \mathbb{R}^{l\times n}, l \geq n, l \in \mathbb{Z}_+. \quad (25)$$

The matrices, $\{L_\sigma\}_{\sigma\in\Sigma}$, of the Lyapunov function (25) are omitted due to space reasons, but can be found in [28]. We set $\Gamma_{\mathcal{X}} = 19.75$ ($P_{\Gamma_{\mathcal{X}}} = \{x \in \mathbb{R}^n \mid V(x) \leq \Gamma_{\mathcal{X}}\}$ as before) $\Gamma_{\mathcal{D}} = 10$, and $\mathcal{X}_\sigma = \mathcal{X} \cap \Omega_\sigma$ for all $\sigma \in \Sigma$. The operating regions of the system and the sets $\{\mathcal{X}_\sigma\}_{\sigma\in\Sigma}$ and $\mathcal{D}$ are shown in Fig. 9.

The sublevel sets are not polytopic, however, the slices are still bounded-semi linear sets and can be computed as explained in Section IV. These slices and the regions of interests $\mathcal{R}$ are shown in Fig. 10. To find a bisimulation quotient for the system, we first refine partition $P_{\mathcal{X}}$ (11) according to partition $P_{pwl} = \{\mathcal{X}_\sigma\}_{\sigma\in\Sigma}$. This additional refinement step guarantees that each

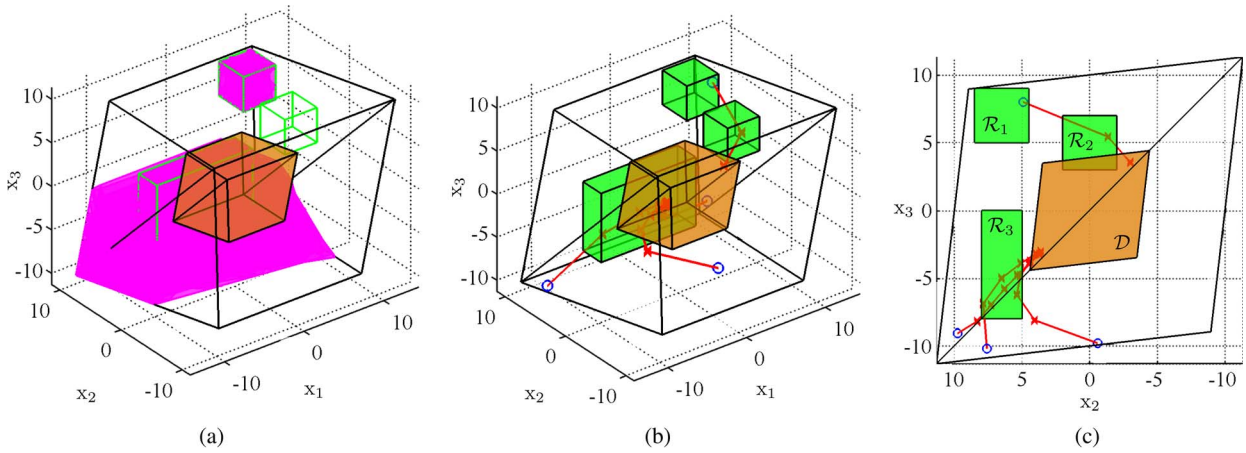

Fig. 8. Case Study 1: (a) $\mathcal{X}^S$ is shown in purple. (b) Four sample trajectories. The initial states are marked by circles. (c) The same trajectories as in (b) are shown under a different view angle (projected on $x_2 - x_3$ plane).

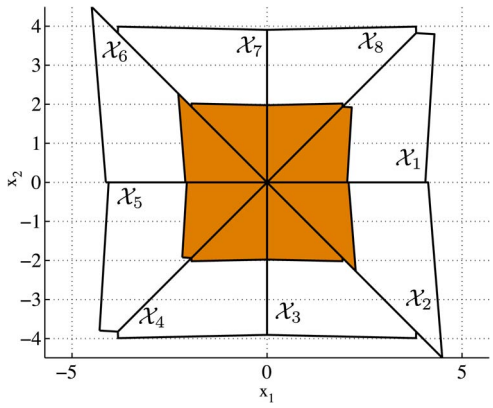

Fig. 9. Case Study 2: $\{\mathcal{X}_\sigma\}_{\sigma\in\Sigma}$ and $\mathcal{D}$ (shown in brown).

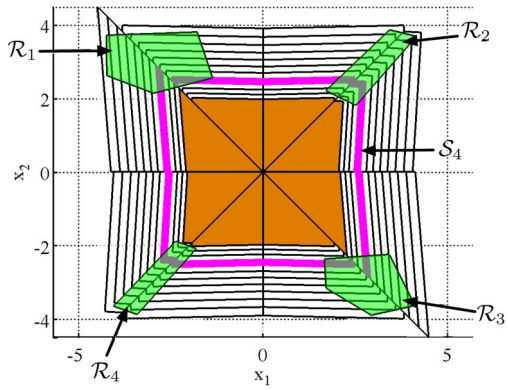

Fig. 10. Case Study 2: Slices $\{\mathcal{S}_i\}_{i=0,\ldots,11}$ and regions of observations $\mathcal{R} = \{\mathcal{R}_i\}_{i=1,\ldots,4}$. $\mathcal{S}_4$ is shown in purple.

$\mathcal{P}$ in the refined partition $P_{\mathcal{X}}^{pwl}$ is included in a set $\mathcal{X}_\sigma$, and therefore, only one mode can be active in $\mathcal{P}$. As each set in $P_{\mathcal{X}}^{pwl}$ is a bounded semi-linear set, Algorithm 2 is used to compute a quotient transition system $\mathcal{T}_e/_\sim$. By eliminating some of the transitions according to P_{pwl}, i.e., $q \xrightarrow{\sigma}_\sim q'$ only if $\mathrm{eq}(q) \subseteq \mathcal{X}_\sigma$, we obtain a bisimulation quotient $\mathcal{T}_e^{pwl}/_\sim$ for the piecewise linear system. The computation took 11 minutes. Note that each state q of $\mathcal{T}_e^{pwl}/_\sim$ has a single outgoing transition, and the system is not controlled.

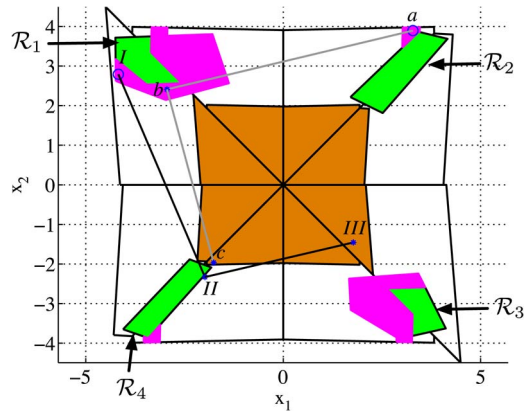

Fig. 11. Case Study 2: Satisfying initial states are shown in purple. Two sample trajectories are indicated by labeling.

We consider the specification: "*A system trajectory never visits* $\mathcal{R}_2$ *and* $\mathcal{R}_4$, *and eventually visits* $\mathcal{R}_1$ *or* $\mathcal{R}_3$," which translates to the following scLTL formula:

$$\phi := (\neg(\mathcal{R}_2 \wedge \mathcal{R}_4) \cup \Pi_{\mathcal{D}}) \wedge \mathrm{F}(\mathcal{R}_1 \wedge \mathcal{R}_3). \tag{26}$$

The set of satisfying initial states of the system is found by using the bisimulation quotient $\mathcal{T}_e^{pwl}/_\sim$ as explained in Section VI. As in the previous case study, both verification and synthesis problems result in the same set, which is shown in Fig. 11. The computation took 0.4 seconds.

VIII. Conclusion

In this paper, we presented a method to abstract the behavior of a switched linear system within a positively invariant subset of $\mathbb{R}^n$ to a finite transition system via the construction of a bisimulation quotient. We employed polyhedral Lyapunov functions to guide the partitioning of the state space and showed that the construction requires polytopic operations only. We showed how this method can be used to synthesize switching sequences and to verify the behavior of the system under arbitrary switching from specifications given as scLTL formulae over polytopic sets in the state space of the system. We also describe how this general approach can be extended to verify piecewise linear systems and systems with difference inclusion dynamics.

REFERENCES

[1] C. Belta *et al.*, "Symbolic planning and control of robot motion [grand challenges of robotics]," *IEEE Robot. Autom. Mag.*, vol. 14, no. 1, pp. 61–70, Mar. 2007.

[2] S. G. Loizou and K. J. Kyriakopoulos, "Automatic synthesis of multiagent motion tasks based on LTL specifications," in *Proc. IEEE Conf. Decision Control*, Paradise Islands, Bahamas, 2004, pp. 153–158.

[3] G. Batt *et al.*, "Validation of qualitative models of genetic regulatory networks by model checking: Analysis of the nutritional stress response in escherichia coli," *Bioinformatics*, vol. 21, no. S1, pp. i19–i28, Jun. 2005.

[4] R. Milner, *Communication and Concurrency*. Upper Saddle River, NJ, USA: Prentice-Hall, 1989.

[5] M. C. Browne, E. M. Clarke, and O. Grumberg, "Characterizing finite kripke structures in propositional temporal logic," *Theor. Comput. Sci.*, vol. 59, no. 1/2, pp. 115–131, Jul. 1988.

[6] J. M. Davoren and A. Nerode, "Logics for hybrid systems," *Proc. IEEE*, vol. 88, no. 7, pp. 985–1010, Jul. 2000.

[7] O. Kupferman and M. Y. Vardi, "Model checking of safety properties," *Formal Methods Syst. Des.*, vol. 19, no. 3, pp. 291–314, Nov. 2001.

[8] P. Tabuada and G. J. Pappas, "Linear time logic control of discrete-time linear systems," *IEEE Trans. Autom. Control*, vol. 51, no. 12, pp. 1862–1877, Dec. 2006.

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

[10] R. Alur and D. L. Dill, "A theory of timed automata," *Theor. Comput. Sci.*, vol. 126, no. 2, pp. 183–235, Apr. 1994.

[11] S. Sankaranarayanan and A. Tiwari, "Relational abstractions for continuous and hybrid systems," in *Computer Aided Verification*, vol. 6806, G. Gopalakrishnan and S. Qadeer, Eds. Berlin, Germany: Springer-Verlag, 2011, ser. Lecture Notes in Computer Science, pp. 686–702.

[12] J. Piovesan, H. Tanner, and C. Abdallah, "Discrete asymptotic abstractions of hybrid systems," in *Proc. IEEE Conf. Decision Control*, 2006, pp. 917–922.

[13] B. Yordanov and C. Belta, "Formal analysis of discrete-time piecewise affine systems," *IEEE Trans. Autom. Control*, vol. 55, no. 12, pp. 2834–2840, Dec. 2010.

[14] H. Lin and P. Antsaklis, "Stability and stabilizability of switched linear systems: A survey of recent results," *IEEE Trans. Autom. Control*, vol. 54, no. 2, pp. 308–322, Feb. 2009.

[15] A. Girard, G. Pola, and P. Tabuada, "Approximately bisimilar symbolic models for incrementally stable switched systems," *IEEE Trans. Autom. Control*, vol. 55, no. 1, pp. 116–126, Jan. 2010.

[16] C. Sloth and R. Wisniewski, "Verification of continuous dynamical systems by timed automata," *Formal Methods Syst. Des.*, vol. 39, no. 1, pp. 47–82, Aug. 2011.

[17] A. Girard and G. Pappas, "Approximation metrics for discrete and continuous systems," *IEEE Trans. Autom. Control*, vol. 52, no. 5, pp. 782–798, May 2007.

[18] M. Zamani, G. Pola, M. Mazo, and P. Tabuada, "Symbolic models for nonlinear control systems without stability assumptions," *IEEE Trans. Autom. Control*, vol. 57, no. 7, pp. 1804–1809, Jul. 2012.

[19] X. C. Ding, M. Lazar, and C. Belta, "Formal abstraction of linear systems via polyhedral Lyapunov functions," in *Proc. IFAC Conf. Anal. Design Hybrid Syst.*, Eindhoven, The Netherlands, Jun. 2012, pp. 88–93.

[20] E. A. Gol, X. C. Ding, M. Lazar, and C. Belta, "Finite bisimulations for switched linear systems," in *Proc. IEEE Conf. Decision Control*, 2012, pp. 7632–7637.

[21] B. Grünbaum, *Convex Polytopes*. New York, NY, USA: Springer-Verlag, 2003.

[22] Z. P. Jiang and Y. Wang, "A converse Lyapunov theorem for discrete-time systems with disturbances," *Syst. Control Lett.*, vol. 45, no. 1, pp. 49–58, Jan. 2002.

[23] M. Lazar, "Model predictive control of hybrid systems: Stability and robustness," Ph.D. dissertation, Eindhoven Univ. Technol., Eindhoven, The Netherlands, 2006.

[24] F. Blanchini, "Ultimate boundedness control for uncertain discrete-time systems via set-induced Lyapunov functions," *IEEE Trans. Autom. Control*, vol. 39, no. 2, pp. 428–433, Feb. 1994.

[25] M. Lazar, "On infinity norms as Lyapunov functions: Alternative necessary and sufficient conditions," in *Proc. IEEE Conf. Decision Control*, Atlanta, GA, USA, 2010, pp. 5936–5942.

[26] M. Kloetzer and C. Belta, "A fully automated framework for control of linear systems from temporal logic specifications," *IEEE Trans. Autom. Control*, vol. 53, no. 1, pp. 287–297, Feb. 2008.

[27] J. Bochnak, M. Coste, and M. F. Roy, *Real Algebraic Geometry*. Berlin, Germany: Springer-Verlag, 1998.

[28] M. Lazar and A. Jokić, "On infinity norms as Lyapunov functions for piecewise affine systems," in *Hybrid Systems: Computation and Control*. New York, NY, USA: ACM Press, 2010, pp. 131–140.

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

[30] Latvala, "Efficient model checking of safety properties," in *Proc. 10th Int. SPIN Workshop Model Checking Softw.*, Portland, OR, USA, 2003, pp. 74–88.

[31] M. Kloetzer and C. Belta, "Dealing with non-determinism in symbolic control," in *Hybrid Systems: Computation and Control*, M. Egerstedt and B. Mishra, Eds. Berlin, Germany: Springer-Verlag, 2008, ser. Lecture Notes in Computer Science, pp. 287–300.

[32] E. Asarin and O. Maler, "As soon as possible: Time optimal control for timed automata," in *Hybrid Systems: Computation and Control*. Berlin, Germany: Springer-Verlag, 1999, pp. 19–30.

Ebru Aydin Gol received the B.S. degree in computer engineering from Orta Dogu Teknik Üniversitesi, Ankara, Turkey, in 2008, the M.S. degree in computer science from Ecole Polytechnique Fèdèrale de Lausanne, Lausanne, Switzerland, in 2010, and the Ph.D. degree in systems engineering at Boston University, Boston, MA, USA in 2014. Her research interests include verification and control of dynamical systems, optimal control, and synthetic biology.

Xuchu (Dennis) Ding received the B.S., M.S., and Ph.D. degree in electrical and computer engineering from the Georgia Institute of Technology, Atlanta, in 2004, 2007 and 2009, respectively. During 2010 to 2011 he was a postdoctoral research fellow at Boston University. In 2011 he joined United Technologies Research Center as a Senior Research Scientist. His research interests include hierarchical mission planning with formal guarantees under dynamic and rich environments, optimal control of hybrid systems, and coordination of multi-agent networked systems.

Mircea Lazar (born in Iasi, Romania, 1978) received the M.Sc. and Ph.D. degrees in control engineering from the Technical University "Gh. Asachi" of Iasi, Romania (2002) and the Eindhoven University of Technology, The Netherlands (2006), respectively. For the PhD thesis he received the European Embedded Control Institute (EECI) PhD award. Since 2006 he has been an Assistant Professor in the Control Systems group of the Electrical Engineering Faculty at the Eindhoven University of Technology. His research interests lie in stability theory, scalable Lyapunov methods and formal methods, and model predictive control.

Calin Belta is an Associate Professor in the Department of Mechanical Engineering, Department of Electrical and Computer Engineering, and the Division of Systems Engineering at Boston University. His research focuses on dynamics and control theory, with particular emphasis on hybrid and cyber-physical systems, formal synthesis and verification, and applications in robotics and systems biology. Calin Belta is a Senior Member of the IEEE and an Associate Editor for the SIAM Journal on Control and Optimization (SICON). He received the Air Force Office of Scientific Research Young Investigator Award and the National Science Foundation CAREER Award.