Automatica 49 (2013) 261–266

Contents lists available at SciVerse ScienceDirect

Automatica

journal homepage: www.elsevier.com/locate/automatica

Brief paper

Formal analysis of piecewise affine systems through formula-guided refinement☆

Boyan Yordanov [a,1], Jana Tůmová [a,2], Ivana Černá [b], Jiří Barnat [b], Calin Belta [c]

[a] Department of Mechanical Engineering, Boston University, United States

[b] Department of Informatics, Masaryk University, Czech Republic

[c] Department of Mechanical Engineering and Division of Systems Engineering, Boston University, United States

ARTICLE INFO

Article history:
Received 15 July 2011
Received in revised form
15 April 2012
Accepted 10 August 2012
Available online 23 October 2012

Keywords:
Piecewise linear analysis
Temporal logic
Verification
Biotechnology

ABSTRACT

We present a computational framework for identifying a set of initial states from which all trajectories of a piecewise affine (PWA) system with additive uncertainty satisfy a linear temporal logic (LTL) formula over a set of linear predicates in its state variables. Our approach is based on the construction and refinement of finite abstractions of infinite systems. We derive conditions guaranteeing the equivalence of an infinite system and its finite abstraction with respect to a specific LTL formula and propose a method for the construction of such formula-equivalent abstractions. While provably correct, the overall method is conservative and expensive. A tool for PWA systems implementing the proposed procedure using polyhedral operations and analysis of finite graphs is made available. Examples illustrating the analysis of PWA models of gene networks are included.

1. Introduction

Recently, there has been increasing interest in developing computational tools for temporal logic analysis and control of dynamical systems (Fainekos, Kress-Gazit, & Pappas, 2005; Kloetzer & Belta, 2008a; Tabuada & Pappas, 2006). In this paper, we focus on piecewise affine systems (PWA) that evolve along different discrete-time affine dynamics in different polytopic regions of the (continuous) state space. PWA systems are widely used, since they can approximate nonlinear dynamics with arbitrary accuracy, and are equivalent to other classes of hybrid systems (Heemels, De Schutter, & Bemporad, 2001). In addition, techniques for the identification of such models from experimental data are available (Juloski et al., 2005).

We are interested in developing a method for the analysis of PWA systems from LTL specifications that leads to more informative results than simple Yes/No answers. We consider the following problem: given a PWA system with additive, polytopic parameter uncertainties and an LTL formula over an arbitrary set of linear predicates in its state variables, find the largest region of initial states from which all trajectories of the system satisfy the formula. Our approach is based on the construction, iterative refinement and verification of finite abstractions inspired by LTL model checking (Baier, Katoen, & Larsen, 2008) and bisimulation-based refinement (Bouajjani, Fernandez, & Halbwachs, 1991). The construction of abstractions is enabled by our previous results (Yordanov & Belta, 2010), where we showed that finite quotients of PWA systems can be constructed by using polyhedral operations only. Our refinement procedure is guided by formula equivalence, i.e., at each iteration we aim at constructing a finite abstraction of the PWA system that satisfies exactly the same formula.

This work can be seen in the context of literature focused on the construction of finite quotients of infinite systems, and is related to Clarke et al. (2003), Pappas (2003) and Tabuada and Pappas (2006). The embedding of discrete-time systems into transition systems is inspired from Pappas (2003) and Tabuada and Pappas (2006). However, while the focus there is on characterizing the existence of bisimulation quotients or developing control strategies using such quotients for linear systems, in this work we consider an analysis problem and focus on the computation of finite quotients, which are equivalent to the original, infinite PWA system with respect to the satisfaction of a specific LTL formula. The related idea of defining CTL formula-specific equivalences coarser than bisimulation has been explored in Aziz, Shiple, Singhal, Brayton, and Sangiovanni-Vincentelli (2002) in the context of finite state

☆ This work was partially supported by grants ARO W911NF-09-1-0088, NSF CNS-0834260, and AFOSR FA9550-09-1-020 at Boston University and by grants LH11065,GD102/09/H042, and GAP202/11/0312 at Masaryk University. The material in this paper was partially presented at the 49th IEEE Conference on Decision and Control (CDC 2010), December 15–17, 2010, Atlanta, Georgia, USA. This paper was recommended for publication in revised form by Associate Editor Maurice Heemels under the direction of Editor Andrew R. Teel.

E-mail addresses: yordanov@microsoft.com (B. Yordanov), xtumova@fi.muni.cz (J. Tůmová), cerna@fi.muni.cz (I. Černá), barnat@fi.muni.cz (J. Barnat), cbelta@bu.edu (C. Belta).

[1] Currently with Biological Computation Group, Microsoft Research, Cambridge, UK.

[2] Currently with Department of Informatics, Masaryk University, Czech Republic.

0005-1098/$ – see front matter © 2012 Elsevier Ltd. All rights reserved.
doi:10.1016/j.automatica.2012.09.027

systems. A trace-equivalent quotient, which might be coarser than a bisimulation one, can be used to analyze an infinite system with any LTL formulas (e.g. see Henzinger, Majumdar, & Raskin, 2005). Instead, our method can lead to the construction of quotients that are even coarser but are equivalent with the infinite system with respect to a specific formula only. While we focus on the analysis of PWA systems, methods targeting other classes of infinite systems might also benefit from the construction of formula-equivalent quotients, provided that all operations required by the method can be implemented.

Our methods differ from counterexample-guided refinement (CEGAR) (Clarke et al., 2003) (a different strategy for specification-based refinement) in two ways. First, instead of performing many model checking steps, in this work we aim directly at the construction of formula-equivalent quotients. Second, we obtain more informative results by identifying satisfying and violating regions (i.e. regions of initial conditions from which all trajectories of the system satisfy or violate the specification). This resembles the construction and refinement of 3-valued abstractions (Bruns & Godefroid, 1999; Chechik & Ding, 2002). While the analysis of PWA systems for properties such as stability, invariance and reachability has been considered previously (Bemporad, Torrisi, & Morari, 2000), our approach allows greater expressivity through LTL specifications.

Compared to our previous approach to the analysis of PWA systems (Yordanov & Belta, 2010), in this paper we develop a more efficient procedure by constructing formula-equivalent abstractions coarser than bisimulation. In addition, we derive conditions under which the analysis results are exact although, in general, the overall procedure is still conservative. A preliminary version of this work was presented in Yordanov, Tumova, Belta, Cerna, and Barnat (2010), where only fixed parameter PWA systems were considered. In this paper, we extend the method from Yordanov et al. (2010) to handle uncertain parameter systems, while introducing additional optimizations to our analysis procedure. The algorithms presented here were implemented as a tool available at http://hyness.bu.edu/software.

2. Preliminaries and notation

We assume familiarity with the following notions (for details see, e.g., Baier et al., 2008) and only introduce some preliminaries.

A *transition system* is a tuple $T = (Q, \rightarrow, O, o)$, where Q is a (possibly infinite) set of states, $\rightarrow \subseteq Q \times Q$ is a transition relation, O is a finite set of observations, and $o : Q \rightarrow O$ is an observation map. We denote by $Pre_T(X) = \{x \in Q \mid \exists x' \in X,\ x \rightarrow x'\}$ the states that reach *region* $X \subseteq Q$ in one step. We denote the *language* of T starting from X as $\mathcal{L}_T(X)$ and use $\mathcal{L}_T$ for $\mathcal{L}_T(Q)$.

We use the graphical notation for the LTL operators (i.e. $\Box$ for always and $\Diamond$ for eventually). Given an LTL formula ϕ over O, we write $T(X) \models \phi$ if all the words from $\mathcal{L}_T(X)$ satisfy ϕ. Let $X_T^\phi = \{x \in Q \mid T(x) \models \phi\}$ denote the largest region of T satisfying ϕ. For a finite T, the set X_T^ϕ can be computed through model-checking (Baier et al., 2008).

We use $\sim_o$ to denote the equivalence relation induced by o over Q (i.e. for $x_1, x_2 \in Q$, $x_1 \sim_o x_2$ iff $o(x_1) = o(x_2)$) and $\sim_b$ to denote the coarsest, observation-preserving bisimulation. The quotient $T/_{\sim_o} = (Q/_{\sim_o}, \rightarrow_{\sim_o}, O, o_{\sim_o})$ simulates T, while T and $T/_{\sim_b}$ are bisimilar. This leads to the following relationships between the region X_T^ϕ for T and its quotients[3]

$$\mathrm{con}(X_{T/_{\sim_o}}^\phi) \subseteq \mathrm{con}(X_{T/_{\sim_b}}^\phi) = X_T^\phi. \tag{1}$$

A *Büchi automaton* is a tuple $\mathcal{B} = (S, S_0, O, \delta_\mathcal{B}, F)$ where S is a finite set of states, $S_0 \subseteq S$ is the set of initial states, O is the input alphabet, $\delta_\mathcal{B} : S \times O \rightarrow 2^S$ is a transition function and $F \subseteq S$ is the set of accepting states. For any LTL formula ϕ, a Büchi automaton denoted by $\mathcal{B}_\phi$ that accepts all and only words satisfying ϕ can be constructed. A *product automaton* $P = T \otimes \mathcal{B}_\phi$ is a Büchi automaton[4] $P = (S_P, S_{P0}, \delta_P, F_P)$ where $S_P = Q \times S$, $S_{P0} = Q \times S_0$, and $F_P = Q \times F$. P accepts all and only words from $\mathcal{L}_T$ satisfying ϕ. We denote the projection of states of P to states of T by $\alpha : S_P \rightarrow Q$ (i.e. for a state (x, s) of P, $\alpha((x, s)) = x \in Q$).

3. Problem formulation and approach

Let $\mathcal{X}_l,\ l \in L$ be a set of open polytopes in $\mathbb{R}^N$, where L is a finite index set, such that $\mathcal{X}_{l_1} \bigcap \mathcal{X}_{l_2} = \emptyset$ for all $l_1, l_2 \in L, l_1 \neq l_2$ and $\mathcal{X} = \cup_{l \in L} cl(\mathcal{X}_l)$ is a closed full-dimensional polytope in $\mathbb{R}^N$ ($cl(\mathcal{X}_l)$ denotes the topological closure of set $\mathcal{X}_l$). A discrete-time piecewise affine (PWA) system is defined as:

$$x_{k+1} = A_l x_k + b_l, \quad x_k \in \mathcal{X}_l,\ l \in L,\ k = 0, 1, \ldots, \tag{2}$$

where, for each mode $l \in L$, parameter b_l is uncertain but known to belong to a polyhedral region $B_l \subset \mathbb{R}^N$. We assume that $\mathcal{X}$ is an invariant for all trajectories of the system and matrix A_l is nonsingular for all $l \in L$.

We are interested in properties of (2) specified over the polytopes from its definition.[5] More specifically, given LTL formula ϕ over L we seek the largest initial region from which all trajectories of the PWA system satisfy ϕ. To formalize this problem, we define the semantics of PWA systems through a transition system embedding[6]:

Definition 1. Let $T_e = (Q_e, \rightarrow_e, O_e, o_e)$, where $Q_e = \cup_{l \in L} \mathcal{X}_l$; $x \rightarrow_e x'$ iff there exist $l \in L, b_l \in B_l$ such that $x \in \mathcal{X}_l$ and $x' = A_l x + b_l$; $O_e = L$; $o_e(x) = l$ iff $x \in \mathcal{X}_l$. Given $X \subseteq Q_e$, all trajectories of system (2) originating in X satisfy an LTL formula ϕ iff $T_e(X) \models \phi$.

From Definition 1, the main problem considered in this paper is formalized as the computation of $X_{T_e}^\phi$. Since T_e has an infinite number of states, it cannot be analyzed directly, which motivates us to consider the generalized problem:

Problem 1. Given an infinite transition system T and an LTL formula ϕ over its set of observations O, find X_T^ϕ.

We assume that the quotient $T/_{\sim_o}$ can be constructed (in Yordanov and Belta (2010) we showed that this is indeed the case for T_e). Then, the set $X_{T/_{\sim_o}}^\phi$ can be computed and used as in Eq. (1) to obtain an under-approximation of X_T^ϕ, leading to a conservative solution to Problem 1. The bisimulation algorithm (Bouajjani et al., 1991) can then be used to decrease this conservatism (a similar idea was applied in Chutinan and Krogh (2001) for ACTL). The equivalence relations $\sim_i$ produced at the i-th iteration of the bisimulation algorithm (i.e. $\sim_0 = \sim_o$ and $\sim_{i+1}$ refines $\sim_i$) provide approximations of X_T^ϕ with increasing accuracy (i.e. $\mathrm{con}(X_{T/_{\sim_i}}^\phi) \subseteq \mathrm{con}(X_{T/_{\sim_{i+1}}}^\phi)$). If the algorithm terminates at step k (which cannot be guaranteed for general infinite systems), the bisimulation $\sim_k = \sim_b$ leads to an exact solution as shown in Eq. (1).

[3] The *concretization* con() maps a region of a quotient to its corresponding region in the concrete system T.

[4] The singleton input alphabet of P is omitted.

[5] These polytopes can capture arbitrary linear predicates.

[6] Such a formalization was also used in Yordanov and Belta (2010) where several additional remarks were considered.

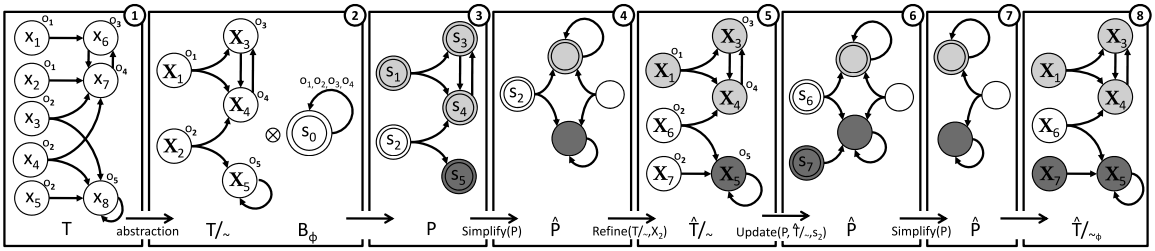

Fig. 1. Algorithm 2 is applied to transition system T (Fig. 1.1) to construct the ϕ-equivalent quotient (Fig. 1.8) for LTL formula $\phi = \Box\neg o_5$. At each step, the sets $S_\top$ and $S_\bot$ (or their projections to $T/_\sim$) are shaded dark or light, respectively, while "dummy" states are shown without labels. The resulting quotient $\hat{T}/_{\sim_\phi}$ (Fig. 1.8) is not a bisimulation quotient but still allows the computation of the largest satisfying (and violating) set of T (note that state X_6 in Fig. 1.8 is inherently uncertain and cannot be refined).

Inspired by this, in Yordanov and Belta (2010) we developed a procedure that refined specific quotient states, possibly leading to the construction of an equivalence relation $\sim$ that was coarser than bisimulation but provided the exact solution[7]

$$\mathrm{con}(X^\phi_{T/_\sim}) = X^\phi_T. \tag{3}$$

We formalize the conditions guaranteeing that Eq. (3) holds (an exact solution to Problem 1 is computed) as follows[8]

Definition 2. An observation preserving equivalence relation is a ϕ-equivalence of T (denoted by $\sim_\phi$) iff $\forall x_1, x_2 \in Q$ such that $x_1 \sim_\phi x_2, T(x_1) \models \phi \Leftrightarrow T(x_2) \models \phi$.

Proposition 1. *Given a transition system T and an LTL formula ϕ, Eq. (3) holds iff $\sim$ is a ϕ-equivalence of T.*

Proof ($\Rightarrow$)**.** Let $\sim$ be a ϕ-equivalence. From Definition 2 it follows that $\forall x \in Q$ such that $T(x) \models \phi, x \in \mathrm{con}(X), X \in T/_\sim$ we have $T/_\sim(X) \models \phi$. Then, $\forall x \in Q, x \in X^\phi_T \Leftrightarrow x \in \mathrm{con}(X^\phi_{T/_\sim})$ and therefore $X^\phi_T = \mathrm{con}(X^\phi_{T/_\sim})$. [$\Leftarrow$] Assume that $\sim$ is not a ϕ-equivalence. Then, $\exists x_1, x_2 \in Q$ such that $x_1 \sim x_2, T(x_1) \models \phi$ and $T(x_2) \not\models \phi$. Considering the equivalence class $X \in Q/_\sim$ such that $x_1, x_2 \in \mathrm{con}(X)$ we have $T/_\sim(X) \not\models \phi$. Then $x_1 \in X^\phi_T$ but $x_1 \notin \mathrm{con}(X^\phi_{T/_\sim})$ and therefore $X^\phi_T \neq \mathrm{con}(X^\phi_{T/_\sim})$. □

The bisimulation $\sim_b$ is a ϕ-equivalence for all LTL formulas ϕ and, therefore, bisimulation is a sufficient condition guaranteeing Eq. (3). However, since we are interested in the analysis of T for a specific LTL formula ϕ, it can be too restrictive (e.g. see Fig. 1). Proposition 1 shows that ϕ-equivalence is necessary and sufficient for Eq. (3). Thus, Problem 1 reduces to the computation of $\mathrm{con}(X^\phi_{T/_{\sim_\phi}})$.

4. Formula-guided refinement

In this section we develop an algorithm for the computation of ϕ-equivalent quotients of (possibly infinite) transition systems, leveraging ideas from the bisimulation algorithm and automata-based model checking. We assume that, given a transition system T, the finite quotient $T/_{\sim_o}$ is computable (as will be the case for T_e) and consider $T/_\sim$ where $\sim$ is observation-preserving (i.e. $\sim$ refines $\sim_o$). To simplify the presentation, we also assume that the LTL formula ϕ can be translated into a deterministic Büchi automaton $\mathcal{B}_\phi$ over the set of observations O. Thus, we focus only on a fragment of LTL (e.g. no deterministic $\mathcal{B}_\phi$ exists for $\phi = \Diamond\Box o$, $o \in O$) but full expressivity can be achieved at a higher computational cost by using a deterministic Rabin automata translation (Baier et al., 2008). Since the computation of ϕ-equivalent quotients is guided by formula ϕ, it is most natural to perform the computation in the product automata $P = T/_\sim \otimes \mathcal{B}_\phi$, which captures both the system and the specification.

Let $S_\top \subseteq S_P$ denote the set of states of P such that all runs originating in $S_\top$ are accepted by P. $S_\top$ can be computed efficiently using a method described in Kloetzer and Belta (2008b)—a subset $F_\top \subseteq F_P$ of accepting states from which infinitely many revisits to F_P are guaranteed is first identified; $S_\top$ is then the set of states from which a visit to $F_\top$ is guaranteed. The largest satisfying set $X^\phi_{T/_\sim}$ of $T/_\sim$ can be computed as the projection $\alpha(S_\top \cap S_{P0}) \subseteq Q/_\sim$. Similarly, we can compute a set of states $S_\bot \subseteq S_P$ such that no runs originating in $S_\bot$ are accepted by P. The projection $\alpha(S_\bot \cap S_{P0}) \subseteq Q/_\sim$ corresponds to $X^{\neg\phi}_{T/_\sim}$, i.e. the largest region of $T/_\sim$ from which all runs violate ϕ.

Let $S_? = S_P \setminus (S_\top \cup S_\bot)$ denote the subset of states from which some but not all runs are accepted by P. The projection $X^?_{T/_\sim} = \alpha(S_? \cap S_{P0}) \subseteq Q/_\sim$ corresponds to states of $T/_\sim$ where both runs satisfying ϕ and $\neg\phi$ originate. We partition the set of states $S_?$ into subsets $S_\prec$ and $S_\div$, where states of $T/_\sim$ from the projection $\alpha(S_\prec \cap S_{P0})$ are inherently uncertain[9] in T, while refinement should be targeted to states from $\alpha(S_\div \cap S_{P0})$. We characterize $S_\prec$ in the following proposition:

Proposition 2. *Given a state $(X, s) \in S_P$, we have $(X, s) \in S_\prec$ if all of the following properties hold:* (i) $(X, s) \in S_?$ *(i.e. both satisfying and violating runs originate there),* (ii) $\forall (X', s') \in \delta_P((X, s)), (X', s') \in S_\top \cup S_\bot \cup S_\prec$ *(i.e. all successors states of (X, s) have been characterized in P),* (iii) $T/_\sim = \mathrm{Refine}(T/_\sim, X)$ *(i.e. state (X, s) cannot be refined further).*

Proof. All successors of (X, s) are characterized in P and, therefore, none of the successors of $X = \alpha((X, s))$ will be considered for further refinement in $T/_\sim$. Since applying $\textsc{Refine}(T/_\sim, X)$ does not affect X, then for all states X' such that $X \to_\sim X'$ we have $\forall x \in \mathrm{con}(X), \exists x' \in \mathrm{con}(X')$ such that $x \to x'$. Therefore, $(X, s) \in S_\prec$ and X is inherently uncertain. □

Proposition 3. *The equivalence relation $\sim$ is a ϕ-equivalence of T if and only if $S_\div \cap S_{P0} = \emptyset$.*

Proof. For necessity, assume $S_\div \cap S_{P0} \neq \emptyset$ where (X, s) is a state of P such that $(X, s) \in S_\div$. Then, $X = \alpha((X, s))$ is a state of $T/_\sim$ such that $\exists x_1, x_2 \in \mathrm{con}(X), T(x_1) \models \phi, T(x_2) \models \neg\phi$ and therefore $\sim$ is not a ϕ-equivalence. For sufficiency, assume $S_\div \cap S_{P0} = \emptyset$. Then, $\forall X \in Q/_\sim$ either (i) $\forall x \in \mathrm{con}(X), T(x) \models \phi$, (ii) $\forall x \in \mathrm{con}(X), T(x) \models \neg\phi$ or (iii) $\forall x \in \mathrm{con}(X), T(x) \not\models \phi$ and $T(x) \not\models \neg\phi$. Therefore $\sim$ is a ϕ-equivalence. □

[7] Similarly to the bisimulation algorithm, the termination of our refinement strategy could not be guaranteed.

[8] Such a characterization was not provided in Yordanov and Belta (2010).

[9] Both satisfying and violating runs originate in each $x \in \mathrm{con}(X)$ for an inherently uncertain $X \in Q/_\sim$ (e.g. see X_6 in Fig. 1.8). These runs cannot be separated via refinement.

In general, the set $S_{\div}$ is nonempty but can be made empty if accepting and non-accepting runs from each state $(X, s) \in S_{\div}$ are separated through refinement—from Proposition 3 this provides a strategy for the construction of ϕ-equivalent quotients. Since the structure of P is completely determined by $\mathcal{B}_\phi$ and $T/_\sim$ but $\mathcal{B}_\phi$ is fixed, states in P can only be refined through refinement of $T/_\sim$. We refine a state $(X, s) \in S_{\div}$ by applying $\textsc{Refine}(T/_\sim, \alpha(X, s))$ (Algorithm 1). This procedure is inspired by the bisimulation algorithm (Bouajjani et al., 1991) and refines $T/_\sim$ locally at a state $X \in Q/_\sim$, returning the refined quotient $\hat{T}/_\sim$.

All outgoing transitions from newly formed states are implicitly induced through the refinement but incoming transitions must be updated (Algorithm 1, line 8): $X'' \rightarrow_\sim X$ is replaced with $X'' \hat{\rightarrow}_\sim X'$ if there exists a newly formed state $X' \in \mathbb{X}_r$ such that $\text{con}(X'') \cap Pre_T(\text{con}(X')) \neq \emptyset$ and removed otherwise. If a self loop at state X was present in $T/_\sim$, all pairwise transitions between subsets from $\mathbb{X}_r$ (including self loops) are possible in $\hat{T}/_\sim$ and must be recomputed. All subsets of a refined state inherit the observation of the parent and, therefore, $\hat{o}_\sim$ is easily updated. Finally, changes made through refinement in the quotient $T/_\sim$ are projected to the product P by applying a function $\hat{P} = \textsc{Update}(P, \hat{T}/_\sim, (X, s))$, rather than by recomputing it as $\hat{P} = \hat{T}/_\sim \otimes \mathcal{B}_\phi$.

Algorithm 1 $\hat{T}/_\sim = \textsc{Refine}(T/_\sim, X)$

1: Initialize $\mathbb{X}_r := \{X\}$
2: **while** there exist $X_r \in \mathbb{X}_r, X' \in Post_{T/_\sim}(X)$ such that $\emptyset \subset con(X_r) \cap Pre_T(con(X')) \subset con(X_r)$ **do**
3: Construct states X_1, X_2 such that:
4: $con(X_1) := con(X_r) \cap Pre_T(con(X'))$
5: $con(X_2) := con(X_r) \setminus Pre_T(con(X'))$
6: $\mathbb{X}_r := (\mathbb{X}_r \setminus X_r) \cup \{X_1, X_2\}$
7: **end while**
8: $\hat{Q}/_\sim := (Q/_\sim \setminus X) \cup \mathbb{X}_r$; update $\hat{\rightarrow}_\sim$ and $\hat{o}_\sim$
9: **return** $\hat{T}/_\sim = (\hat{Q}/_\sim, \hat{\rightarrow}_\sim, O, \hat{o}_\sim)$

The product automaton P can be simplified by respectively replacing each set $S_\top$, $S_\perp$ and $S_\prec$ by a single dummy state s_1, s_2 or s_3, such that $\delta_P(s_1) = \{s_1\}$, $\delta_P(s_2) = \{s_2\}$, $\delta_P(s_3) = \{s_1, s_2\}$ and $s_1 \in F_P$. For a state $s \in S_P \setminus (S_\top \cup S_\perp \cup S_\prec)$ a transition to a dummy state is included if a transition to a state from the corresponding set was present (e.g. if there existed a state $s' \in S_\top$ such that $s' \in \delta_P(s)$ then $s_1 \in \delta_P(s)$). This simplification is performed by the function $\hat{P} = \textsc{Simplify}(P)$, which reduces the number of states of P and leads to faster computation.

The overall method is summarized in Algorithm 2 and an example of its application is shown in Fig. 1. The procedure is not guaranteed to terminate but termination can be enforced by imposing a limit on the number of iterations or refining up to certain state granularity.[10] In such cases only an under-approximation of the solution to Problem 1 is obtained, which can be improved by adjusting these limits. Additional optimizations based on the decomposition of the product P into strongly connected components (SCC)—a common technique in model checking—are possible, since SCCs respect the characterization of states of P as $S_\top$, $S_\perp$, or $S_?$.

Remark 1. In Yordanov and Belta (2010), we constructed both the product automaton $P_\phi = T/_\sim \otimes \mathcal{B}_\phi$ and $P_{\neg\phi} = T/_\sim \otimes \mathcal{B}_{\neg\phi}$ and used model checking to find $X^{\phi}_{T/_\sim}$ and $X^{\neg\phi}_{T/_\sim}$. Here, we avoid the construction of $P_{\neg\phi}$ and apply methods inspired by Büchi games (Kloetzer & Belta, 2008b) on P_ϕ to find these sets.

[10] When quotient states are represented by polytopes as for PWA systems (Yordanov & Belta, 2010), the radii of the inscribed spheres provide a convenient quotient state granularity metric.

Algorithm 2 Given T and ϕ, compute $X^{\phi}_{\hat{T}/_\sim}, X^{\neg\phi}_{\hat{T}/_\sim}$

1: Construct $T/_{\sim_o}$; translate ϕ to deterministic Büchi automaton $\mathcal{B}_\phi$; construct $P = T/_{\sim_o} \otimes \mathcal{B}_\phi$
2: Initialize $\hat{T}/_\sim := T/_{\sim_o}$ and $\hat{P} := P$
3: **repeat**
4: Compute $S_\top$ and $S_\perp$ in $\hat{P}$, $S_? := S_{\hat{P}} \setminus (S_\top \cup S_\perp)$
5: Compute $S_\prec \subseteq S_?$, $S_{\div} := S_? \setminus S_\prec$
6: **for all** $(X, s) \in S_{\div}$ **do**
7: **if** X not yet refined in $\hat{T}/_\sim$ **then**
8: $\hat{T}/_\sim := \textsc{Refine}(\hat{T}/_\sim, X)$
9: **end if**
10: $\hat{P} := \textsc{Update}(\hat{P}, \hat{T}/_\sim, (X, s))$
11: **end for**
12: $\hat{P} := \textsc{Simplify}(\hat{P})$
13: **until** $\hat{P}$ not updated during an iteration
14: **return** $X^{\phi}_{\hat{T}/_\sim} = \alpha(S_\top \cap S_{P0}), X^{\neg\phi}_{\hat{T}/_\sim} = \alpha(S_\perp \cap S_{P0})$

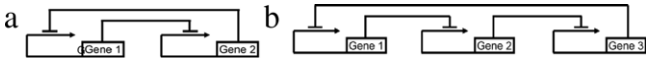

Fig. 2. Schematic representations of: A the genetic toggle switch (Gardner et al., 2000); and B the Repressilator (Elowitz & Leibler, 2000).

Complexity: An LTL formula ϕ of size $|\phi|$ can be translated into a Büchi automaton of size $2^{\mathcal{O}(|\phi|)}$ (Baier et al., 2008) resulting in a product automaton with $O(|Q/_\sim| \cdot 2^{\mathcal{O}(|\phi|)})$ states. The characterization of the states of P discussed in Section 4 has complexity $\mathcal{O}(|S_P| \cdot |\delta_P|)$. Whenever state $X \in Q/_\sim$ is refined, updating P can add up to $2^{|Post_{T/_\sim}(X)|} \cdot |\mathcal{B}_\phi|$ states but the growth of P is controlled through simplification. When ϕ cannot be translated into a deterministic Büchi automaton, a deterministic Rabin automaton of double-exponential size with respect to the size of the formula can be used. In this case, the computation of sets $S_\top$ and $S_\perp$ can also become more expensive, since a Rabin acceptance condition must be considered. Although the theoretical complexity of the solution did not improve over our previous results (Yordanov & Belta, 2010), a significant computational reduction can be reached in practice (see Section 5, Table 1) by avoiding the construction of two product automata as in Yordanov and Belta (2010), reducing the size of P and targeting refinement to specific states.

5. Implementation and case study

The method described above was implemented as a Matlab tool (available at http://hyness.bu.edu/software). The tool takes as input a PWA system (Eq. (2)) and an LTL formula and produces a set of satisfying initial regions. The construction of quotients for PWA systems is implemented as described in Yordanov and Belta (2010), where quotient regions are represented as (unions of) polytopes and the computation of *Pre*(), intersection, set difference and emptiness checks are performed using polyhedral operations using routines from the MPT toolbox (Kvasnica, Grieder, & Baotić, 2004).

To demonstrate the described approach and compare it to our method from Yordanov and Belta (2010) we analyze two PWA models inspired by classical synthetic gene networks (Elowitz & Leibler, 2000; Gardner, Cantor, & Collins, 2000) (Fig. 2). Gene regulation is modeled by ramp functions, which are PWA functions defined by two threshold values, inducing three regions of different dynamics. At low repressor concentrations (below threshold 1) the regulated gene is fully expressed, at high repressor concentrations

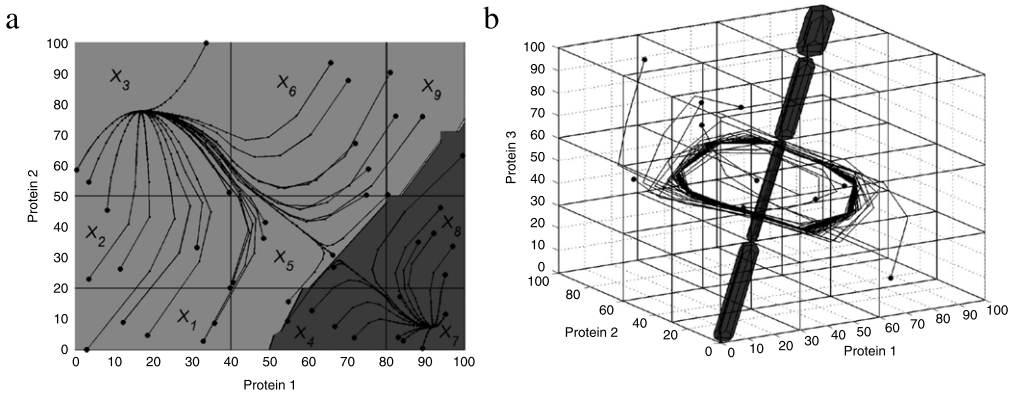

Fig. 3. A: Simulated trajectories of the toggle switch PWA model go towards one of two stable equilibria located in regions $\mathcal{X}_3$ and $\mathcal{X}_7$ (initial states are marked by open circles). Trajectories originating in the light gray region are guaranteed to satisfy specification $\Diamond 3$, while trajectories originating in the dark gray region are guaranteed to satisfy $\Diamond 7$. B: Simulated trajectories of the repressilator PWA model oscillate visiting region of high and low concentrations of Protein 3 (initial states are marked by open circles). Trajectories originating anywhere but the shaded region are guaranteed to satisfy specification $\Box(\Diamond\phi_1 \wedge \Diamond\phi_2)$.

Table 1
Computational results from the case study (results obtained using the approach from Yordanov and Belta (2010) are given in parentheses). All computation was performed on a desktop computer with a 3.0 GHz AMD Athlon II X4 640 processor and 4 GB of memory. Computation times (with a 20 iterations limit) are reported as "time". The relative volumes (as a percentage of the total state space) of the identified satisfying and violating regions are reported as "% sat". and "% viol"., respectively. The number of states in the initial quotient and the quotient after all refinement steps are reported as $|Q/_{\sim}|$ and $|\hat{Q}/_{\sim}|$, respectively. The number of states in the product automaton after all refinement, updating and simplification is reported as $|\hat{S}_P|$ (note that, for the method from Yordanov and Belta (2010), no product simplification is involved and the number of states of both P_ϕ and $P_{\neg\phi}$ are reported (see Remark 1)).

System	Toggle switch		Repressilator
Spec.	$\Diamond 3$	$\Diamond 7$	$\Box(\Diamond\phi_1 \wedge \Diamond\phi_2)$
Time	29 (97) s	28.9 (96) s	17 (153) min
% sat.	79.7 (79.8)	20.1 (20.1)	99.62 (99.76)
% viol.	20.1 (20.1)	79.7 (79.8)	0 (0)
$\lvert Q/_{\sim}\rvert$	9	9	27
$\lvert\hat{Q}/_{\sim}\rvert$	423 (463)	419 (459)	2845 (2970)
$\lvert\hat{S}_P\rvert$	7 (926, 463)	7 (918, 459)	17 (8910, 8910)

(above threshold 2) expression is only basal and the response between the two thresholds is graded. Due to space constraints, the dynamics of the PWA models are illustrated only through the simulated trajectories in Fig. 3(A) and (B), which show that the characteristic bistability (Gardner et al., 2000) and oscillations (Elowitz & Leibler, 2000) are captured. The first system includes two mutually inhibiting genes (Gardner et al., 2000) and acts as a switch, allowing only one of the genes to be expressed depending on initial conditions. It is modeled as a two dimensional ($N = 2$) PWA model with nine rectangular regions ($L = \{1, \ldots, 9\}$, $\mathcal{X}_1, \ldots, \mathcal{X}_9$ in Fig. 3(A)). We seek the maximal sets of initial conditions guaranteeing that all trajectories of the system eventually reach regions $\mathcal{X}_3$ or $\mathcal{X}_7$, respectively, specified using LTL formulas $\Diamond 3$ and $\Diamond 7$. Biologically, these regions correspond to the states of the system where one gene is fully expressed, while the other is expressed only basally (the two positions of the switch). The second network includes three inhibiting genes and has been experimentally shown to produce oscillations (Elowitz & Leibler, 2000). A three dimensional ($N = 3$) PWA model with 27 hyper-rectangular regions denoted by $\mathcal{X}_1, \ldots \mathcal{X}_{27}$ was constructed (Fig. 3(B)). We are interested in testing whether all initial conditions lead to oscillatory behavior and define sub-formulas ϕ_1 and ϕ_2, which are satisfied when concentrations of the protein produced from gene 3 are respectively low and high (i.e. ϕ_1 is the disjunction of all regions $\mathcal{X}_l$ such that $\forall x \in \mathcal{X}_l, [0\ 0\ 1]x < 30$ and, similarly, ϕ_2 is the disjunction of all regions $\mathcal{X}_l$ such that $\forall x \in \mathcal{X}_l, [0\ 0\ 1]x > 60$). By analyzing the system with LTL formula $\Box(\Diamond\phi_1 \wedge \Diamond\phi_2)$ we search for the maximal set of initial conditions guaranteeing that trajectories of the system keep oscillating between low and high concentrations of protein 3. Results obtained by applying the analysis methods described in this paper and in Yordanov and Belta (2010) to both systems are summarized in Table 1, while all identified satisfying regions are shown in Fig. 3.

References

Aziz, A., Shiple, T., Singhal, V., Brayton, R., & Sangiovanni-Vincentelli, A. (2002). Formula-dependent equivalence for compositional CTL model checking. *Formal Methods in System Design*, *21*, 193–224.

Baier, C., Katoen, J-P., & Larsen, K. G. (2008). *Principles of model checking*. MIT Press.

Bemporad, A., Torrisi, F. D., & Morari, M. (2000). Optimization-based verification and stability characterization of piecewise affine and hybrid systems. In *LNCS*: *vol. 1790. HSCC*.

Bouajjani, A., Fernandez, J.-C., & Halbwachs, N. (1991). Minimal model generation. In *LNCS*: *vol. 531. CAV* (pp. 197–203).

Bruns, G., & Godefroid, P. (1999). Model checking partial state spaces with 3-valued temporal logics. In *LNCS*: *vol. 1633. CAV* (pp. 274–287).

Chechik, M., & Ding, W. (2002). Lightweight reasoning about program correctness. *Information Systems Frontiers*, *4*(4), 363–377.

Chutinan, A., & Krogh, B. H. (2001). Verification of infinite-state dynamic systems using approximate quotient transition systems. *IEEE Transactions on Automatic Control*, *46*(9), 1401–1410.

Clarke, E., Fehnker, A., Han, Z., Krogh, B., Ouaknine, J., Stursberg, O., & Theobald, M. (2003). Abstraction and counter-example-guided refinement in model checking of hybrid systems. *International Journal of Foundations of Computer Science*, *14*(4), 583–604.

Elowitz, M. B., & Leibler, S. (2000). A synthetic oscillatory network of transcriptional regulators. *Nature*, *403*, 335–338.

Fainekos, G. E., Kress-Gazit, H., & Pappas, G. J. (2005). Hybrid controllers for path planning: a temporal logic approach. In *IEEE CDC*.

Gardner, T. S., Cantor, C. R., & Collins, J. J. (2000). Construction of a genetic toggle switch in Escherichia coli. *Nature*, *403*(6767), 339–342.

Heemels, W. P. M. H., De Schutter, B., & Bemporad, A. (2001). Equivalence of hybrid dynamical models. *Automatica*, *37*(7), 1085–1091.

Henzinger, T. A., Majumdar, R., & Raskin, J.-F. (2005). A classification of symbolic transition systems. *ACM Transactions on Computational Logic*, *6*(1), 1–32.

Juloski, A. Lj., Heemels, W. P. M. H., Ferrari-Trecate, G., Vidal, R., Paoletti, S., & Niessen, J. H. G. (2005). Comparison of four procedures for the identification of hybrid systems. In *LNCS*: *vol. 3414. HSCC* (pp. 354–369).

Kloetzer, M., & Belta, C. (2008a). A fully automated framework for control of linear systems from temporal logic specifications. *IEEE Transactions on Automatic Control*, *53*(1), 287–297.

Kloetzer, M., & Belta, C. (2008b). Dealing with non-determinism in symbolic control. In *LNCS*: *vol. 4981. HSCC* (pp. 287–300). Berlin, Heidelberg: Springer.

Kvasnica, M., Grieder, P., & Baotić, M. (2004). Multi-parametric toolbox, MPT.

Pappas, G. J. (2003). Bisimilar linear systems. *Automatica*, *39*(12), 2035–2047.
Tabuada, P., & Pappas, G. (2006). Linear time logic control of discrete-time linear systems. *IEEE Transactions on Automatic Control*, *51*(12), 1862–1877.
Yordanov, B., & Belta, C. (2010). Formal analysis of discrete-time piecewise affine systems. *IEEE Transactions on Automatic Control*, *55*(12), 2834–2840.
Yordanov, B., Tůmová, J., Belta, C., Černá, I., & Barnat, J. (2010). Formal analysis of piecewise affine systems through formula-guided refinement. In *IEEE CDC* (pp. 5899–5904).

Boyan Yordanov received the B.A. degrees in biochemistry and computer science from Clark University, Worcester, MA, in 2005 and the M.S. and Ph.D. degrees in biomedical engineering from Boston University, Boston, MA, in 2009 and 2011, respectively.

He was a Postdoctoral Researcher in the Department of Mechanical Engineering, Boston University, in 2011 and is currently a Postdoctoral Researcher at the Biological Computation Group as part of the Computational Science Laboratory at Microsoft Research, Cambridge, UK His research interests are in the areas of analysis and design of biological systems, synthetic biology, temporal logic, formal methods, and hybrid systems.

Jana Tůmová received the B.S. and M.Sc. degrees in applied computer science and computer science from Masaryk University, Brno, Czech Republic, in 2006 and 2009, respectively. She is currently pursuing the Ph.D. degree in computer science at Masaryk University. Her research interests include formal methods, temporal logics, model checking, and controller synthesis.

Ivana Černá received the M.Sc. and Ph.D. degrees in computer science from Comenius University, Bratislava, Slovak Republic, in 1986 and 1992, respectively.

She is a Professor at the Faculty of Informatics, Masaryk University, Brno, Czech Republic. Her research interests include theory of communicating and parallel systems, formal verification and verification tools, algorithm design, and analysis. She is a coauthor of algorithms implemented in a parallel and distributed verification tool DiVinE.

Jiří Barnat received the M.Sc. and Ph.D. degrees in computer science from Masaryk University, Brno, Czech Republic, in 2000 and 2005, respectively.

He is an Associate Professor at the Faculty of Informatics, Masaryk University. His research interests include parallel algorithms, parallel and distributed methods in formal verification, and platform dependent algorithm engineering. He is currently leading the ParaDiSe research group at the Faculty of Informatics. He is a coauthor of the parallel and distributed verification tool DiVinE.

Calin Belta received the B.S. and M.Sc. degrees in control and computer science from the Technical University of Iasi, Iasi, Romania, the M.Sc. degree in electrical engineering from Louisiana State University, Baton Rouge, and the M.Sc. and Ph.D. degrees in mechanical engineering from the University of Pennsylvania, Philadelphia.

He is currently an Associate Professor of Mechanical Engineering, Systems Engineering, and Bioinformatics at Boston University, Boston, MA. His research interests include analysis and control of hybrid systems, motion planning and control, and bio-molecular networks. Dr. Belta is an Associate Editor for the *SIAM Journal on Control and Optimization* (SICON). He received the AFOSR Young Investigator Award in 2008 and the NSF CAREER Award in 2005.