49th IEEE Conference on Decision and Control
December 15-17, 2010
Hilton Atlanta Hotel, Atlanta, GA, USA

Formal Analysis of Piecewise Affine Systems through Formula-Guided Refinement

Boyan Yordanov, Jana Tůmová, Calin Belta, Ivana Černá, and Jiří Barnat

Abstract— **We present a computational framework for identifying a set of initial states from which all trajectories of a piecewise affine (PWA) system 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 temporal logic formula and propose methods aimed at the construction of such formula-equivalent abstractions. We show that the proposed procedure can be implemented using polyhedral operations and analysis of finite graphs. While provably correct, the overall method is conservative and expensive. The proposed algorithms have been implemented as a software tool that is available for download. An illustrative example for a PWA gene network model is included.**

I. INTRODUCTION

In control problems, trajectories of "complex" mathematical models of physical systems, such as systems of differential or difference equations, are usually checked against "simple" specifications, such as stability of equilibria and set invariance. In formal verification, "rich" specifications, such as formulas of temporal logics, are checked against "simple" models such as (finite) transition graphs and automata models of software programs and digital circuits [8]. The study of physical systems requires the development of theoretical frameworks and computational tools for bridging in this gap, and therefore allowing for specifying the properties of continuous and hybrid systems in a rich language, with automatic verification and controller synthesis. Recent results include temporal logics for systems with continuous dynamics [9], control of linear systems from temporal logic specifications [18], task specification and controller synthesis in mobile robotics [10], and specification and analysis of qualitative behavior of genetic circuits [3].

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 as models in many areas. They can approximate nonlinear dynamics with arbitrary accuracy, and are equivalent with other classes of hybrid systems [13]. In addition, there exist several techniques for the identification of such models from experimental data (see [14] for a review).

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 GA201/09/1389 and GA201/09/P497 at Masaryk University.
B. Yordanov (yordanov@bu.edu) is with the Dept. of Biomedical Engineering, Boston University. J. Tůmová (xtumova@fi.muni.cz, tumova@bu.edu) is with Dept. of Informatics, Masaryk University and the Dept. of Mechanical Engineering, Boston University. C. Belta (cbelta@bu.edu) is with the Dept. of Mechanical Engineering and the Division of Systems Engineering, Boston University. I. Černá (cerna@fi.muni.cz) and J. Barnat (barnat@fi.muni.cz) are with Dept. of Informatics, Masaryk University

We consider the following problem: given a PWA system and an arbitrary LTL formula over a 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 to this problem is based on the construction and iterative refinement of finite abstractions. The refinement is guided by formula equivalence, *i.e.,* we aim at constructing a finite abstraction of the PWA system that satisfies exactly the same formula. To this goal, we use ideas from LTL model checking [8] and bisimulation - based refinement [4].

This work can be seen in the context of literature focused on the construction of finite quotients of infinite systems, and is related to [17], [18], [7]. The embedding of discrete-time systems into transition systems is inspired from [17], [18]. 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, formula equivalent quotients.

The related idea of defining CTL formula specific equivalences coarser than bisimulation has been explored in [1] in the context of finite state systems. In contrast, we consider infinite systems and LTL formulas. Relying on a temporal logic formula to guide the refinement of an abstraction is also part of CEGAR-based methods for verification [7]. Instead of performing many model checking steps, in this work we aim directly at the construction of formula equivalent finite quotients. In addition, our approach yields more informative results, since we obtain regions of initial conditions for which the system satisfies the specification, instead of simple Yes/No answers. The construction of the abstractions is enabled by our previous results [20], where we showed that finite quotients of PWA systems can be constructed by using polyhedral operations only. Analysis of PWA systems for properties such as invariance and reachability can be performed more efficiently [15] but our method allows for greater expressivity by considering specifications expressed as LTL formulas.

The method presented in this paper was implemented in MATLAB and is available at http://hyness.bu.edu/software.

II. DEFINITIONS AND PRELIMINARIES

Definition 1: A transition system is a tuple $T = (Q, \to, O, o)$, 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.

A transition $(x, x') \in \rightarrow$ is also denoted by $x \rightarrow x'$. The transition system T is *finite* if its set of states Q is finite and *infinite* otherwise. The transition system T is *deterministic* if, for all $x \in Q$, there exists at most one $x' \in Q$ such that $x \rightarrow x'$. Finally, T is called *non-blocking* if, for every state $x \in Q$, there exists $x' \in Q$ such that $x \rightarrow x'$. In this paper only non-blocking transition systems are considered.

A *trajectory* or *run* of T starting from x_0 is an infinite sequence $x_0x_1x_2\dots$ with the property that $x_i \in Q$, and $x_i \rightarrow x_{i+1}$, for all $i \geq 0$. A trajectory $x_0x_1x_2\dots$ defines a *word* $o(x_0)o(x_1)o(x_2)\dots$. The set of all words generated by the set of all trajectories starting at $x \in Q$ is called the *language* of T originating at x and is denoted by $\mathcal{L}_T(x)$.

A subset $X \subseteq Q$ is called a *region* of T. The set of all trajectories originating in X is denoted by $T(X)$ and the set of all words generated by runs in $T(X)$ is called the language of T originating at X and is denoted by $\mathcal{L}_T(X) = \bigcup_{x\in X} \mathcal{L}_T(x)$. For an arbitrary region X, we define the set of states $Pre_T(X)$ that reach X in one step as

$$Pre_T(X) = \{x \in Q \mid \exists x' \in X,\ x \rightarrow x'\} \tag{1}$$

The observation map o of a transition system T induces an observational equivalence relation $\sim$ over the set of states Q. We say that states $x_1, x_2 \in Q$ are equivalent (written as $x_1 \sim x_2$) if and only if $o(x_1) = o(x_2)$. The equivalence relation naturally induces a *quotient transition system* $T/_\sim = (Q/_\sim, \rightarrow_\sim, O, o_\sim)$. $Q/_\sim$ is the quotient space (the set of all equivalence classes). Given an equivalence class $X \in Q/_\sim$ [1], we denote the set of all equivalent states in that class by $con(X) \subseteq Q$ (con stands for concretization map). Since all states $x \in Q$ in an equivalence class $X \in Q/_\sim$ have the same observation, $o_\sim(X)$ is well defined and given by $o_\sim(X) = o(x), x \in con(X)$. The transition relation $\rightarrow_\sim$ is defined as follows: for $X_1, X_2 \in Q/_\sim$, $X_1 \rightarrow_\sim X_2$ if and only if there exist $x_1 \in con(X_1)$ and $x_2 \in con(X_2)$ such that $x_1 \rightarrow x_2$. It is easy to see that

$$\forall X \in Q/_\sim, \mathcal{L}_T(con(X)) \subseteq \mathcal{L}_{T/_\sim}(X). \tag{2}$$

The quotient transition system $T/_\sim$ is said to *simulate* the original system T.

Definition 2: The equivalence relation $\sim$ induced by the observation map o is a bisimulation of a transition system $T = (Q, \rightarrow, O, o)$ if, for all states $x_1, x_2 \in Q$, if $x_1 \sim x_2$ and $x_1 \rightarrow x_1'$, then there exist $x_2' \in Q$ such that $x_2 \rightarrow x_2'$ and $x_1' \sim x_2'$.

If $\sim$ is a bisimulation, then the quotient transition system $T/_\sim$ is called a *bisimulation quotient* of T, and the transition systems T and $T/_\sim$ are called *bisimilar*, denoted as $T/_\sim \simeq T$. An immediate consequence of bisimulation is language equivalence, *i.e.,*

$$\forall X \in Q/_\sim, \mathcal{L}_T(con(X)) = \mathcal{L}_{T/_\sim}(X). \tag{3}$$

[1]with a slight abuse of notation, we use symbol X to denote states of $T/_\sim$ (*i.e.* $X \in Q/_\sim$) and regions of T (*i.e.* $X \subseteq Q$) but the precise meaning should be clear from the context

Using the $Pre_T()$ operator defined in Equation (1), a characterization of bisimulation can be given as follows: the equivalence relation $\sim$ is a bisimulation if and only if for all equivalence classes $X' \in Q/_\sim$, $Pre_T(con(X'))$ is either empty or a finite union of equivalence classes. Equivalently, the bisimulation property (Def. 2) is violated at $X \in Q/_\sim$ if there exists a state $X' \in Q/_\sim$, such that

$$\emptyset \subset con(X) \cap Pre_T(con(X')) \subset con(X). \tag{4}$$

This leads to an iterative procedure for the construction of the coarsest bisimulation $\sim$, known as the "bisimulation algorithm" [4], which in general does not terminate but if it does then $T/_\sim$ is a finite bisimulation quotient.

To specify temporal logic properties for system trajectories, in this paper we use Linear Temporal Logic (LTL) formulas [8]. LTL formulas are inductively defined over the set of observations O, by using the standard Boolean operators (*i.e.,* $\neg$ (negation), $\vee$ (disjunction), $\wedge$ (conjunction)) and the temporal operators $\bigcirc$ ("next"), $\mathcal{U}$ ("until"), $\Box$ ("always"), $\Diamond$ ("eventually"). Each LTL formula ϕ over an alphabet O defines a language $\mathcal{L}_\phi$ of words that satisfy ϕ. Given a finite transition system $T = (Q, \rightarrow, O, o)$ and an LTL formula ϕ over O, an off-the-shelf model checker, such as `NuSMV` [6] or `DiVinE` [2], can be used to check whether the language $\mathcal{L}_T(x)$ satisfies ϕ, for all $x \in Q$. For a region $X \subseteq Q$, we write $T(X) \vDash \phi$ if all the words from $\mathcal{L}_T(X)$ satisfy ϕ. Let

$$X_T^\phi = \{x \in Q \mid T(x) \vDash \phi\}. \tag{5}$$

Note that, X_T^ϕ is the largest region of T satisfying ϕ (*i.e.* if $x \notin X_T^\phi$, then there exists a word in $\mathcal{L}_T(x)$ that violates ϕ).

If $T/_\sim$ is a quotient of T, then for any equivalence class $X \in Q/_\sim$ and formula ϕ, we have:

$$T/_\sim(X) \vDash \phi \Rightarrow T(con(X)) \vDash \phi \tag{6}$$

In addition, if $\sim$ is a bisimulation, then

$$T/_\sim(X) \vDash \phi \Leftrightarrow T(con(X)) \vDash \phi \tag{7}$$

Properties (6) and (7) (which follow immediately from (2) and (3)) allow one to model check finite quotients and extend the results to the (possibly infinite) original transition system.

Definition 3: 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 nondeterministic transition function and $F \subseteq S$ is the set of accepting (final) states.

The semantics of a Büchi automaton is defined over infinite input words. A run of $\mathcal{B}$ over a word $w = o_1o_2o_3\dots \in O^\omega$ is a sequence $\rho = s_0s_1s_2\dots$, where $s_0 \in S_0$ and $(s_{i-1}, o_i, s_i) \in \delta_\mathcal{B}$ for all $i \geq 1$. Let $\inf(\rho)$ denote the set of states that appear in the run ρ infinitely often. A run ρ of $\mathcal{B}$ is accepting if and only if $\inf(\rho) \cap F \neq \emptyset$. In other words, an input word w is accepted by $\mathcal{B}$ if and only if there exists at least one run over w that visits F infinitely often. We denote by $\mathcal{L}_\mathcal{B}$ the language accepted by $\mathcal{B}$, i.e. the set of all words accepted by $\mathcal{B}$. An LTL formula ϕ can always be translated into a Büchi automaton $\mathcal{B}_\phi$ using off-the-shelf tools such as

LTL2BA [12], such that $\mathcal{L}_{\mathcal{B}_\phi} = \mathcal{L}_\phi$. A Büchi automaton is deterministic if S_0 and $\delta_{\mathcal{B}}(s, o)$ are singletons for all $s \in S$ and $o \in O$.

III. 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 closure of set $\mathcal{X}_l$). A discrete-time piecewise affine (PWA) system is defined as:

$$x_{k+1} = A_l x_k + b_l,\ x_k \in \mathcal{X}_l,\ l \in L,\ k = 0, 1, 2, \ldots. \quad (8)$$

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 (8) specified in terms of the polytopes from its definition. Informally, the semantics of system (8) can be understood in the following sense: a trajectory of the system $x_0 x_1 x_2 \ldots$ produces an infinite word $l_0 l_1 l_2 \ldots$, where $l_i \in L$ is the index of the polytope visited at step i (*i.e.* $x_i \in \mathcal{X}_{l_i}$). An LTL formula over L can then be interpreted over trajectories of the system (see Sec. II). In the following, we give a formal definition of the semantics through an embedding into a transition system.

Definition 4: The embedding transition system $T_e = (Q_e, \rightarrow_e, O_e, o_e)$ for the PWA system from Eqn. (8) is defined as $Q_e = \cup_{l \in L} \mathcal{X}_l$, $x \rightarrow_e x'$ if and only if there exist $l \in L$ such that $x \in \mathcal{X}_l$ and $x' = A_l x + b_l$, $O_e = L$, and $o_e(x) = l$ if and only if $x \in \mathcal{X}_l$.

Note that the embedding T_e has an infinite number of states and is always deterministic and non-blocking.

Definition 5: Given a subset $X \subseteq Q_e$, we say that all trajectories of system (8) originating in X satisfy formula ϕ if and only if $T_e(X)$ satisfies ϕ.

Problem 1: Given a PWA system (8) and an LTL formula ϕ over L, find the largest region of initial states, from which all trajectories of the system satisfy ϕ.

The solution to Problem 1 amounts to the computation of $X^\phi_{T_e}$ (see Eqn. (5)). Since T_e has an infinite number of states, it cannot be analyzed directly. Our approach is based on the construction of a finite abstraction that is equivalent to the initial system with respect to the satisfaction of a specific temporal logic formula. We develop the notion of formula equivalent quotients in Sec. IV and describe an algorithm for their computation in Sec. V, where the results are valid for general deterministic infinite transition systems. We describe the construction of a formula equivalent finite quotient for T_e in Sec. VI, which leads to the solution of Problem 1. As it will become clear later, our approach is conservative, in the sense that, we can only "try" to find the formula equivalent quotient of T_e and the satisfying region $X^\phi_{T_e}$ but, in general, we can only guarantee to obtain subsets of the latter.

IV. Analysis of Infinite Transition Systems

In this section, we consider the following problem:

Problem 2: Given an infinite transition system T (Def. 1) and an LTL formula ϕ over its set of observations O, find X^ϕ_T (Eqn. (5)).

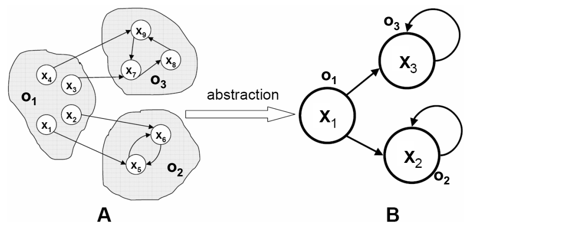

Fig. 1: The transition system in (A) forms three equivalence classes under observational equivalence and the resulting finite quotient (B) is clearly not a bisimulation quotient (Eqn. (4) is violated at state X_1). However, the quotient is ϕ-equivalent for LTL formula $\phi = \bigcirc(o_2 \vee o_3)$ and can be equivalently used for model checking ϕ instead of the original system.

We assume that, given the observational equivalence relation $\sim$ (see Sec. II), the finite quotient $T/_\sim$ is computable (its computation for T_e is discussed in Sec. VI). Then, $X^\phi_{T/_\sim}$ can be computed by model checking $T/_\sim$ from each state $X \in Q/_\sim$. From Eqn. (6) it follows that $con(X^\phi_{T/_\sim}) \subseteq Q$ is a satisfying region in T but, in general, $con(X^\phi_{T/_\sim}) \subseteq X^\phi_T$, so only a subset of the largest satisfying region is obtained.

If $\sim$ is a bisimulation of T then from Eqn. (7) it follows that for any LTL formula ϕ

$$con(X^\phi_{T/_\sim}) = X^\phi_T. \quad (9)$$

A solution to Problem 2 can then be obtained by computing the coarsest bisimulation $\sim$ of T using the bisimulation algorithm (see Sec. II) and model checking the bisimulation quotient $T/_\sim$ from each state to compute $X^\phi_{T/_\sim}$. However, such a procedure would only work for the particular case when T admits a finite bisimulation quotient $T/_\sim$. Alternatively, the equivalence relation produced at each step of the bisimulation algorithm can be used to construct finite simulation quotients, which can then be model checked against an LTL formula. A similar idea was used in [5] for the universal fragment ACTL of CTL. We followed this approach in [20], where we combined state refinement, inspired by the bisimulation algorithm, with model checking in an iterative procedure (see Remark 1 for additional details). At each step, the set $con(X^\phi_{T/_\sim}) \subseteq X^\phi_T$ provided an under-approximation of the solution to Problem 2. This under-approximation could be improved by performing additional iterations but the termination of the algorithm with an exact solution could not be guaranteed. In the following, we consider conditions guaranteeing that Eqn. (9) holds and therefore an exact solution to Problem 2 can be computed. As already stated, bisimulation is one such condition, but as it will become clear later, it is unnecessarily strong.

Definition 6: Given an (infinite) transition system T and an LTL formula ϕ, an observation preserving equivalence relation $\sim$ is a ϕ-equivalence of T if and only if for all states $x_1, x_2 \in Q$ such that $x_1 \sim x_2$, $T(x_1) \vDash \phi \Leftrightarrow T(x_2) \vDash \phi$

We denote a ϕ-equivalence relation as $\sim_\phi$ and refer to the quotient $T/_{\sim_\phi}$ as ϕ-equivalent quotient. From Eqn. (7) it follows that a bisimulation relation $\sim$ is a ϕ-equivalence for all LTL formulas ϕ. Bisimulation is a sufficient condition guaranteeing that Eqn. (9) holds but since we are interested in the analysis of T for a specific LTL formula ϕ it can be too restrictive. An example where a formula equivalent quotient is not a bisimulation quotient is shown in Fig. 1.

Proposition 1: Given a transition system T and an LTL formula ϕ, Eqn. (9) holds iff $\sim$ is a ϕ-equivalence of T. Prop. 1 shows that ϕ-equivalence is a necessary and sufficient condition for Eqn. (9) and its proof is available in [21]. Then, Problem 2 reduces to the computation of $con(X^{\phi}_{T/\sim_\phi})$, where $T/\sim_\phi$ is a finite, ϕ-equivalent quotient for T. We discuss the computation of formula equivalent quotients in Sec. V.

Remark 1: The method we presented in [20] involved the iterative model checking and refinement of simulation quotients of an infinite transition system. By model checking with both an LTL formula and its negation we were able to target refinement to the specific set of states from which some but not all runs satisfied the formula. Although our method was originally inspired by the bisimulation algorithm, this optimization led to the construction of formula equivalent quotients in the cases when the algorithm terminated. However, a large number of model checking and refinement steps was required. As it will become clear in Sec. V our current approach aims directly at the construction of formula equivalent quotients and is more efficient.

V. Formula Guided Refinement

In this section we develop an algorithm for the computation of ϕ-equivalent quotients of (possibly infinite) transitions systems, leveraging ideas from the bisimulation algorithm and automata-based model checking. We assume that given a deterministic transition system T and the observational equivalence relation $\sim$, the finite quotient $T/_\sim$ is computable (as will be the case for T_e). For the sake of presentation we also assume that LTL formula ϕ can be translated into a deterministic Büchi automaton $\mathcal{B}_\phi$ over the set of observations O. Although this restricts the specification to a fragment of LTL, our method can be easily modified to handle full LTL expressivity by translating the specification into a deterministic Rabin automaton instead. Since the computation of ϕ-equivalent quotients is guided by formula ϕ, it is most natural to perform the computation in the product automata $P = T/_\sim \otimes \mathcal{B}_\phi$ (Def. 7), where both the structure of the system ($T/_\sim$) and the specification ($\mathcal{B}_\phi$) is captured.

Definition 7: The *product automaton* $P = T/_\sim \otimes \mathcal{B}_\phi$ of a finite transition system $T/_\sim = (Q/_\sim, \rightarrow_\sim, O, o_\sim)$ and a Büchi automaton $\mathcal{B}_\phi = (S, S_0, O, \delta_{\mathcal{B}_\phi}, F)$ accepting the language $\mathcal{L}_\phi$ for some LTL formula ϕ is defined as $P = (S_P, S_{P0}, \delta_P, F_P)$. $S_P = Q/_\sim \times S$ is the set of states of P, $S_{P0} = Q/_\sim \times S_0$ is the set of initial states, and $F_P = Q/_\sim \times F$ is the set of accepting states. The transition function is δ_P where for a $(X, s) \in S_P$, $\delta_P((X, s)) = \{(X', s') \in S_P \mid X \rightarrow_\sim X' \text{ and } s' = \delta_{\mathcal{B}_\phi}(s, o(X))\}$.
The product automaton is a nondeterministic Büchi automaton with input alphabet containing only one element, which is therefore omitted. An accepting run $r_P = (X_1, s_1)(X_2, s_2)\ldots$ in P can be projected into a run $r_{T/_\sim} = X_1X_2\ldots$ of $T/_\sim$, such that $o(X_1)o(X_2)\ldots$ is accepted by $\mathcal{B}_\phi$ [19] and therefore satisfies ϕ. Let us denote by $\alpha : S_P \rightarrow Q/_\sim$, $\alpha(X, s) = X$, the projection of states of product automaton P to the states of $T/_\sim$.

The set $X^{\phi}_{T/_\sim}$ can be computed as the projection $\alpha(S_Y \cap S_{P0}) \subseteq Q/_\sim$ where $S_Y \subseteq S_P$ is the set of states in P from which all runs are accepting (see Sec. II). S_Y can be efficiently computed following the method that we described in [16]. Specifically, we first identify a subset $F_Y \subseteq F$ of accepting states, from which infinitely many revisits to F are guaranteed. S_Y is then a set of states from which a visit to F_Y is guaranteed in zero or more steps.

We can also easily identify a set of states $S_N \subseteq S_P$ of P from which no runs are accepting. The projection $\alpha(S_N \cap S_{P0}) \subseteq Q/_\sim$ corresponds to $X^{\neg\phi}_{T/_\sim}$ (*i.e.* the largest set of states of $T/_\sim$ from which no runs satisfy ϕ).

Let $S_? \subseteq S_P, S_? = S_P \setminus (S_Y \cup S_N)$ be the set of states from which some but not all runs are accepting in P. The projection $\alpha(S_? \cap S_{P0}) \subseteq Q/_\sim$ corresponds to states of $T/_\sim$ where both runs satisfying ϕ and $\neg\phi$ originate and, therefore, the ϕ-equivalence property (Def. 6) is violated at those states.

Proposition 2: The equivalence relation $\sim$ is a ϕ- equivalence of a deterministic transition system T if and only if $(S_? \cap S_{P0}) = \emptyset$. Then, $S_? = \emptyset$ guarantees that $\sim$ is a ϕ-equivalence.
A proof for Prop. 2 is available in [21]. In general, the set $S_?$ is nonempty but can be made empty if accepting and non-accepting runs from each state $(X, s) \in S_?$ are separated through refinement. Following from Prop. 2 and the discussion presented in Sec. IV this provides a solution to Problem 2. Since the structure of P is completely determined by $\mathcal{B}_\phi$ and $T/_\sim$ and $\mathcal{B}_\phi$ is fixed, the only way to refine states in P is through refinement of $T/_\sim$. We refine a state $(X, s) \in S_?$ by applying the procedure Refine $(T/_\sim, \alpha(X, s))$, followed by Update $(P, (X, s))$, which simply projects changes made during Refine $(T/_\sim, \alpha(X, s))$ into the product P.

Algorithm 1 Refine$(T/_\sim, X)$

```
1: initialize $\hat{Q}/_\sim = Q/_\sim \setminus X$
2: for each state $X'$ such that $X \rightarrow_\sim X'$ do
3:     construct state $X_{new}$ such that
       $con(X_{new}) = con(X) \cap Pre_T(con(X'))$
4:     add state $X_{new}$ to $\hat{Q}/_\sim$
5: end for
6: update $\hat{\rightarrow}_\sim$ and $\hat{o}_\sim$
7: return $\hat{T}/_\sim = (\hat{Q}/_\sim, \hat{\rightarrow}_\sim, O, \hat{o}_\sim)$
```

The refinement procedure Refine$(T/_\sim, X)$ (Algorithm 1) is inspired by the bisimulation algorithm (see Sec. II). Unlike the bisimulation algorithm, which refines the equivalence relation $\sim$ globally, Refine$(T/_\sim, X)$ refines the quotient $T/_\sim$ locally at a state $X \in Q/_\sim$. By considering all its successors, state $X \in Q/_\sim$ is partitioned so that each resulting subset X_{new} can make a transition to only one of the original successor states X'. An outgoing transition $X_{new} \hat{\rightarrow}_\sim X'$ of a newly formed state X_{new}, where $con(X_{new}) = con(X) \cap Pre_T(con(X'))$ is thus implicitly induced. The incoming transitions of X_{new} are updated as follows: each transition $X' \rightarrow_{\sim} X$ is replaced with $X' \hat{\rightarrow}_{\sim} X_{new}$ if $con(X') \cap Pre_T(con(X_{new})) \neq \emptyset$. All subsets of a refined state inherit the observation of the parent and, therefore, $\hat{o}_{\sim}$ is easily updated.

The overall method discussed in this section is summarized in Algorithm 2. Since the regions of T contain, in general, an infinite number of states, the algorithm might perform an infinite number of refinement steps. To ensure the algorithm terminates, we refine a state only if it corresponds to a "large enough" region of T (see Sec. VI for such a measure for T_e). This means that $S_?$ might be nonempty when we force the algorithm to terminate, and we cannot guarantee an exact solution to Problem 2.

It is important to note that if a state X is refined in $\hat{T}/_{\sim}$, not every state (X, s) is necessarily refined in $\hat{P}$. There is a one-to-many correspondence between the refined product $\hat{P}$ and the refined quotient $\hat{T}/_{\sim}$, and the projection α is of the type $\alpha : S_{\hat{P}} \rightarrow 2^{\hat{Q}/\sim_{\phi}}$. This might lead to a major reduction in the computational complexity of the solution: the product automaton $\hat{P}$ after refinement might be significantly smaller than the product automaton $\hat{T}/_{\sim} \otimes \mathcal{B}$, which we used for model checking in our old approach [20]. An example illustrating this idea is available in [21].

Algorithm 2 COMPUTE $X^{\phi}_{\hat{T}/\sim}$

```
1:  Construct $T/_{\sim}$, such that $\sim$ is observational equivalence
2:  Construct deterministic BA $\mathcal{B}_{\phi}$, such that $\mathcal{L}_{\mathcal{B}_{\phi}} = \mathcal{L}_{\phi}$
3:  Build product automaton $P = T/_{\sim} \otimes \mathcal{B}_{\phi}$
4:  Initialize $\hat{T}/_{\sim} = T/_{\sim}$, $\hat{P} = P$
5:  repeat
6:    Compute $S_Y$ and $S_N$ in $\hat{P}$
7:    $S_? = S_{\hat{P}} \setminus (S_Y \cup S_N)$
8:    for all $(X, s) \in S_?$ do
9:      if $X$ is large enough and not refined in $\hat{T}/_{\sim}$ then
10:       $\hat{T}/_{\sim}$ =REFINE($\hat{T}/_{\sim}, X$)
11:     end if
12:     if $X$ is refined in $\hat{T}/_{\sim}$ then
13:       UPDATE($\hat{P}, (X, s)$)
14:     end if
15:   end for
16: until $\hat{P}$ not updated during previous iteration
17: return $X^{\phi}_{\hat{T}/\sim} = \alpha(S_Y \cap S_{P0})$
```

VI. FINITE QUOTIENTS OF PWA SYSTEMS

Through the embedding of the PWA system (Eqn. (8)) into an infinite transition system T_e (Def. 4), we reduced Problem 1 to Problem 2. In Sec. IV we developed the notion of formula equivalent finite quotients of infinite systems and showed that such quotients can be used to provide the solution to Problem 2. Based on the assumptions that finite quotients can be constructed and each step of the refinement procedure can be implemented, we proposed an algorithm for the refinement of such quotients in order to obtain formula equivalence in Sec. V. In this section, we discuss the construction of the quotients, which completes the solution to Problem 1. Note that this is a summary of results from [20].

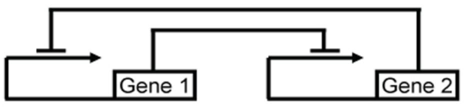

Fig. 2: Schematic representations of the genetic toggle switch [11].

From the definitions of the observational equivalence relation $\sim$, induced by the observation map o (Sec. II) and T_e (Def. 4), the initial set of states $Q_e/_{\sim}$, of the finite quotient $T_e/_{\sim}$, is simply the set of observations $Q_e/_{\sim} = O_e = L$ and the observation map is identity. Given a state $l \in Q_e/_{\sim}$, $con(l) = \mathcal{X}_l$ is a polytope from the system definition (Eqn. (8)). In order to finish the construction of the quotient, we need to find the set of transitions $\rightarrow_{e,\sim}$. By the definition of $\rightarrow_{\sim}$ (Sec.II) and Eqn. (1), given $l, l' \in Q_e/_{\sim}$, we have:

$$l \rightarrow_{e,\sim} l' \Leftrightarrow \mathcal{X}_l \cap Pre_{T_e}(\mathcal{X}_{l'}) \neq \emptyset \qquad (10)$$

The transition relation $\rightarrow_{e,\sim}$ can then be constructed using polyhedral operations only, since

$$\mathcal{X}_l \cap Pre_{T_e}(\mathcal{X}_{l'}) = \mathcal{X}_l \cap A_l^{-1}(\mathcal{X}_{l'} - b_l). \qquad (11)$$

In order to implement REFINE$(T_e/_{\sim}, l)$ we need to be able to construct a state l_{new}, such that given $l' \in Q_e/_{\sim}$ where $l \rightarrow_{\sim} l'$, $con(l_{new}) = con(l) \cap Pre_{T_e}(con(l'))$ (see Algorithm 1), which is computable using Eqn. (11). Outgoing transitions of a refined state are implicitly induced as discussed in Sec. V and the computation of Eqn. (10) can be used to recompute incoming transitions whenever refinement is applied. In order to decide if the region $con(l)$ is large enough to undergo additional refinement (as discussed in Sec. V), we compute the radius of its inscribed sphere and apply refinement if it is larger than a certain predefined limit ϵ.

VII. IMPLEMENTATION AND CASE STUDY

The method described in this paper was implemented in MATLAB, where all polyhedral operations were performed using routines from the MPT toolbox [15]. The tool takes as input a PWA system (as defined in Eqn. (8)) and an LTL formula and produces a set of satisfying initial regions. The tool is made public and is available through our web site at http://hyness.bu.edu/software.

We developed a PWA model for a network of two mutually repressing genes (Fig. 2), inspired by the genetic toggle switch [11]. Gene regulation is modeled by ramp functions, which are piecewise affine 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 (above threshold 2) expression is only basal and the response between the two thresholds is graded. The PWA model captures the characteristic bistability of the system, allowing only one of the genes to be expressed depending on initial conditions (Fig. 3-A). Since there are two repressors, two ramp functions are used in a two dimensional ($N = 2$)

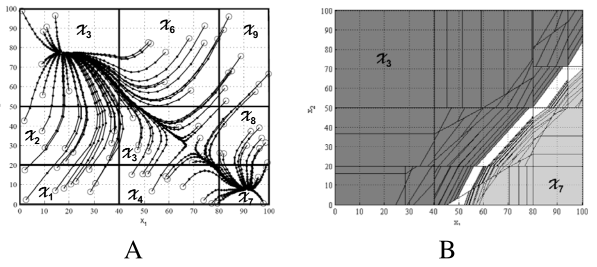

Fig. 3: A) Simulated trajectories of the toggle switch PWA model go towards one of two stable equilibria located in regions $\mathcal{X}_3$ and $\mathcal{X}_7$ (initial states are marked by open circles). B) Results from the analysis of the toggle switch PWA model indicate that trajectories originating in the dark gray region are guaranteed to satisfy specification $\phi_1 = \Diamond 3$, while trajectories originating in the light gray region are guaranteed to satisfy $\phi_2 = \Diamond 7$.

PWA model and, therefore, the system has a total of nine rectangular regions (denoted $\mathcal{X}_1, \ldots, \mathcal{X}_9$ with the set of labels $L = \{1, \ldots, 9\}$). Dynamics 3 and 7 have unique, asymptotically stable equilibria inside rectangles $\mathcal{X}_3$ and $\mathcal{X}_7$ (see Fig. 3-A). Biologically, the equilibria correspond to the two modes of the system (each gene can be fully expressed, while the other is expressed only basally). An interesting problem is finding the regions of attraction for the two equilibria. By exploiting convexity properties of affine functions on polytopes, it can be shown that $\mathcal{X}_3$ and $\mathcal{X}_7$ are invariants for dynamics 3 and 7, respectively. From this, we can immediately conclude that $\mathcal{X}_3$ and $\mathcal{X}_7$ are regions of attraction for the two equilibria. Therefore, our problem reduces to finding maximal regions satisfying LTL formulas $\phi_1 = \Diamond 3$ and $\phi_2 = \Diamond 7$. In other words, we want to find maximal sets of initial conditions guaranteeing that all trajectories of the system eventually reach regions $\mathcal{X}_3$ or $\mathcal{X}_7$, respectively.

The initial finite quotient included 9 states and its computation required under 1 sec. on a 3.4GHz machine with 1GB of memory. A limit $\epsilon = 2$ was imposed on the size of regions that can undergo refinement as described in Sec. VI. The refinement procedure for both specifications required under 20 sec. and terminated without returning a formula equivalent quotient but satisfying regions were identified for both ϕ_1 and ϕ_2 and are shown in Fig. 3-B. It is important to note that nothing can be guaranteed about trajectories originating in the region shown in white in Fig. 3-B and satisfying regions are given as unions of open polytopes, so boundaries are not considered part of the satisfying set of initial conditions.

VIII. Conclusion

We described a computational framework for the identification of initial sets from which all trajectories of a discrete-time piecewise affine (PWA) system satisfy a specification expressed as a Linear Temporal Logic (LTL) formula over linear predicates. Our approach is based on the iterative construction and refinement of abstractions. We showed that existing methods for the refinement of such abstractions might be too restrictive and proposed conditions guaranteeing the equivalence of an infinite system and its finite abstraction with respect to a specific temporal logic formula. We developed methods for the refinement of finite abstractions aiming at the construction of formula equivalent systems and demonstrated that our approach can be applied to the analysis of small genetic networks.

References

[1] A. Aziz, T. Shiple, V. Singhal, R. Brayton, and A. Sangiovanni-Vincentelli, "Formula-dependent equivalence for compositional CTL model checking," *Formal Methods in System Design*, vol. 21, pp. 193–224, 2002.

[2] J. Barnat, L. Brim, and P. Ročkai, "DiVinE 2.0: High-Performance Model Checking," in *High Performance Computational Systems Biology (HiBi)*. IEEE Computer Society Press, 2009, pp. 31–32.

[3] G. Batt, D. Ropers, H. de Jong, J. Geiselmann, R. Mateescu, M. Page, and D. Schneider, "Validation of qualitative models of genetic regulatory networks by model checking : Analysis of the nutritional stress response in *Escherichia coli*," *Bioinformatics*, vol. 21, no. Suppl.1, pp. i19–i28, 2005.

[4] A. Bouajjani, J.-C. Fernandez, and N. Halbwachs, "Minimal model generation," in *Computer-Aided Verification (CAV)*, vol. 531, 1990, pp. 197–203.

[5] 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, 2001.

[6] A. Cimatti, E. Clarke, E. Giunchiglia, F. Giunchiglia, M. Pistore, M. Roveri, R. Sebastiani, and A. Tacchella, "NuSMV Version 2: An OpenSource Tool for Symbolic Model Checking," in *Computer-Aided Verification (CAV)*, vol. 2404, 2002.

[7] E. Clarke, A. Fehnker, Z. Han, B. Krogh, J. Ouaknine, O. Stursberg, and M. Theobald, "Abstraction and counterexample-guided refinement in model checking of hybrid systems," *International Journal of Foundations of Computer Science*, vol. 14, no. 4, pp. 583–604, 2003.

[8] E. M. Clarke, D. Peled, and O. Grumberg, *Model checking*. MIT Press, 1999.

[9] J. Davoren, V. Coulthard, N. Markey, and T. Moor, "Non-deterministic temporal logics for general flow systems," in *Hybrid Systems: Computation and Control*, 2004, pp. 280–295.

[10] G. E. Fainekos, H. Kress-Gazit, and G. J. Pappas, "Hybrid controllers for path planning: a temporal logic approach," in *IEEE Conf. on Decision and Control*, 2005.

[11] T. Gardner, C. Cantor, and J. Collins, "Construction of a genetic toggle switch in Escherichia coli." *Nature*, vol. 403, no. 6767, pp. 339–342, 2000.

[12] P. Gastin and D. Oddoux, "LTL with past and two-way very-weak alternating automata," in *Proc. MFCS'03*, vol. 2747, 2003, pp. 439–448.

[13] W. P. M. H. Heemels, B. D. Schutter, and A. Bemporad, "Equivalence of hybrid dynamical models," *Automatica*, vol. 37, no. 7, pp. 1085–1091, 2001.

[14] A. L. Juloski, W. Heemels, G. Ferrari-Trecate, R. Vidal, S. Paoletti, and J. Niessen, "Comparison of four procedures for the identification of hybrid systems," in *Hybrid Systems: Computation and Control*, 2005, vol. 3414, pp. 354–369.

[15] M. Kvasnica, P. Grieder, and M. Baotić, "Multi-Parametric Toolbox (MPT)," 2004. [Online]. Available: http://control.ee.ethz.ch/∼mpt/

[16] M.Kloetzer and C. Belta, "Dealing with non-determinism in symbolic control," in *Hybrid Systems: Computation and Control*, 2008, pp. 287–300.

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

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

[19] M. Y. Vardi and P. Wolper, "An automata-theoretic approach to automatic program verification," in *Proc. Logic in Computer Science*, 1986.

[20] B. Yordanov, C. Belta, and G. Batt, "Model checking discrete time piesewise affine systems: application to gene networks," in *European Control Conference*, 2007.

[21] B. Yordanov, J. Tůmová, C. Belta, I. Černá, and J. Barnat, "Formal analysis of piecewise affine systems through formula-guided refinement," Boston University, Tech. Rep. CISE 2010-IR-0013, 2010.