Parameter Synthesis for Piecewise Affine Systems from Temporal Logic Specifications

Boyan Yordanov and Calin Belta*

Boston University, Boston MA
yordanov@bu.edu, cbelta@bu.edu

Abstract. In this paper, we consider discrete-time continuous-space Piecewise Affine (PWA) systems with parameter uncertainties, and study temporal logic properties of their trajectories. Specifically, given a PWA system with polytopal parameter uncertainties, and a Linear Temporal Logic (LTL) formula over linear predicates in the states of the system, we attempt to find subsets of parameters guaranteeing the satisfaction of the formula by all trajectories of the system. We illustrate our method by applying it to a PWA model of a two-gene network.

Keywords: Piecewise Affine Systems, Formal Verification.

1 Introduction

Temporal logics and model checking [1] are customarily used for specifying and verifying the correctness of digital circuits and computer programs. However, due to their resemblance to natural language, expressivity, and existence of off-the-shelf algorithms for model checking, temporal logics have the potential to impact several other areas. Examples include analysis of systems with continuous dynamics [2], control of linear systems from temporal logic specifications [3,4], task specification and controller synthesis in mobile robotics [5,6] and specification and analysis of qualitative behavior of genetic circuits [7,8,9].

In this paper we focus on piecewise affine systems (PWA) that evolve along different affine dynamics (in discrete time) in different polytopal regions of the (continuous) state space. PWA systems are widely used as models in many areas, including systems and synthetic biology, where they are particularly fitting for describing gene circuits [8]. PWA systems are also attractive because of the existence of tools for model identification [10]. Additionally, PWA systems are quite general, since they can approximate nonlinear dynamics with arbitrary accuracy [11], and are proven to be equivalent with several other classes of hybrid systems [12]. Even so, a PWA system with fixed parameters might not provide a good model of a real system. This is especially true for gene networks, where processes depend on various, hard to control external factors such as temperature and concentrations of chemicals not part of the system. To develop a model that

* This work is partially supported by NSF CAREER 0447721 and NSF 0410514 at Boston University.

M. Egerstedt and B. Mishra (Eds.): HSCC 2008, LNCS 4981, pp. 542–555, 2008.

can capture the rich behavior of systems under a range of conditions, a PWA model with uncertain parameters can be used. For such models, the dynamics in each region of the state space can take on parameters from a polytopal range.

A rich spectrum of properties of dynamical systems are naturally expressed in Linear Temporal Logic (LTL) [1] formulas over linear predicates in the state variables. Examples include remaining within certain regions (invariance), getting to certain target regions (reachability), or avoiding dangerous regions (safety). In this paper, we consider a parameter synthesis problem: given a PWA system with polytopal parameter uncertainties, and a Linear Temporal Logic (LTL) formula over linear predicates in the states of the system, we attempt to find subsets of parameters guaranteeing the satisfaction of the formula by all trajectories of the system. Our approach is based on the construction of finite simulation quotients, model checking, and use of counterexamples for determining ranges of allowed parameters.

From a theoretical and computational point of view, this work can be seen in the context of literature focused on the construction of finite quotients of infinite systems (see [13] for a review), and is closely related to [14,3,4] and our previous work on formal analysis of PWA systems with fixed parameters [15]. Unlike counterexample guided refinement [16], where violating trajectories of the quotient are checked against the concrete model and, if spurious, removed by refinement, we use counterexamples to remove a set of (possibly spurious) violating transitions from the quotient. From an application point of view, this paper relates to [8,17,18,19,9], where temporal logics are used to specify properties of biomolecular networks. These works analyze whether the trajectories of a system satisfy a temporal logic formula. In contrast, we search parameter sets guaranteeing the satisfaction of a formula. We implemented our method as a software tool for parameter synthesis in PWA systems, which is freely downloadable from `http://iasi.bu.edu/~cbelta/software.htm`.

2 Preliminaries

2.1 Transition Systems, Simulations, and Bisimulations

Definition 1. *A transition system is a tuple* $T = (Q, Q_0, \rightarrow, \Pi, \models)$*, where* Q *is a set of states,* $Q_0 \subseteq Q$ *is the set of initial states,* $\rightarrow \subseteq Q \times Q$ *is a transition relation,* Π *is a finite set of atomic propositions, and* $\models \subseteq Q \times \Pi$ *is a satisfaction relation.*

A transition $(q, q') \in \rightarrow$ is also denoted by $q \rightarrow q'$. The transition system T is called *finite* if its set of states Q is finite, and infinite otherwise. The transition system T is called *non-blocking* if, for every state $q \in Q$, there exists $q' \in Q$ such that $(q, q') \in \rightarrow$ (*i.e.,* the relation $\rightarrow$ is total). The transition system T is called *deterministic* if, for all $q \in Q$, there exists at most one $q' \in Q$ such that $(q, q') \in \rightarrow$ (the case $q = q'$ is included in the definitions above).

For an arbitrary state $q \in Q$, we define $\Pi_q = \{\pi \in \Pi \mid q \models \pi\}$, $\Pi_q \in 2^{\Pi}$ as the set of all atomic propositions satisfied at q. A *trajectory* or *run* of T

starting from $q \in Q_0$ is an infinite sequence $r = r(1)r(2)r(3)\ldots$ with the property that $r(1) = q$, $r(i) \in Q$, and $(r(i), r(i+1)) \in \rightarrow$, for all $i \geq 1$. A trajectory $r = r(1)r(2)r(3)\ldots$ defines a *word* $w = w(1)w(2)w(3)\ldots$, where $w(i) = \Pi_{r(i)}$. The set of all words generated by the set of all trajectories starting at $q \in Q_0$ is called the *language* of T originating at q and is denoted by $L_T(q)$. The language of the transition system T is defined as $L_T(Q_0)$ or L_T for simplicity.

A subset X of the state set Q ($X \subseteq Q$) is called a *region* of T. For an arbitrary region X, we define the set of states $Post(X)$ that can be reached from X in one step as

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

and the set of states that can reach X in one step as

$$Pre(X) = \{q' \in Q \mid \exists q \in X,\ q' \rightarrow q\} \tag{2}$$

An equivalence relation $\sim \subseteq Q \times Q$ over the state space of T is *proposition preserving* if for all $q_1, q_2 \in Q$ and all $\pi \in \Pi$, if $q_1 \sim q_2$ and $q_1 \vDash \pi$, then $q_2 \vDash \pi$. Among the several proposition preservation equivalence relations that can be defined, *propositional equivalence* defined as $q_1 \sim q_2$ if and only if $\Pi_{q_1} = \Pi_{q_2}$ is of special interest. A proposition preserving equivalence relation naturally induces a *quotient transition system* $T/_\sim = (Q/_\sim, Q_0/_\sim, \rightarrow_\sim, \Pi, \vDash_\sim)$. $Q/_\sim$ is the quotient space (the set of all equivalence classes) and $Q_0/_\sim = \{X \in Q/_\sim \mid \exists q \in X$ such that $q \in Q_0\}$ is the set of all initial equivalence classes. 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 $q_1 \in X_1$ and $q_2 \in X_2$ such that $q_1 \rightarrow q_2$. The satisfaction relation is defined as follows: for $X \in Q/_\sim$, we have $X \vDash_\sim \pi$ if and only if there exist $q \in X$ such that $q \vDash \pi$. It is easy to see that $L_T(X) \subseteq L_{T/_\sim}(X)$, for any $X \in Q_0/_\sim$ (with a slight abuse of notation, we use the same symbol X to denote both a state of $T/_\sim$ and the corresponding region of equivalent states of T). The quotient transition system $T/_\sim$ is said to *simulate* the original system T, which is written as $T/_\sim \geq T$.

Definition 2. *A proposition preserving equivalence relation $\sim$ is a bisimulation of a transition system $T = (Q, Q_0, \rightarrow, \Pi, \vDash)$ if, for all states $p, q \in Q$, if $p \sim q$ and $p \rightarrow p'$, then there exist $q' \in Q$ such that $q \rightarrow q'$ and $p' \sim q'$.*

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 by $T/_\sim \cong T$. An immediate consequence of bisimulation is language equivalence, *i.e.,* $L_T(X) = L_{T/_\sim}(X)$, for all $X \in Q_0/_\sim$.

2.2 Linear Temporal Logic and Model Checking

To specify temporal logic properties for trajectories of PWA systems, in this paper we use Linear Temporal Logic [1]. Informally, the LTL formulas are recursively defined over a set of atomic propositions Π, by using the standard Boolean operators (*e.g.,* $\neg$ (negation), $\vee$ (disjunction), $\wedge$ (conjunction)) and temporal

operators, which include $\mathcal{U}$ ("until"), $\Box$ ("always"), $\Diamond$ ("eventually"). LTL formulas are interpreted over infinite words in 2^{Π}, as are those generated by the transition system T from Definition 1. If ϕ_1 and ϕ_2 are two LTL formulas over Π, formula $\phi_1 \mathcal{U} \phi_2$ intuitively means that (over some word) ϕ_2 will eventually become true and ϕ_1 is true until this happens. For an LTL formula ϕ, formula $\Diamond\phi$ means that ϕ becomes eventually true, whereas $\Box\phi$ indicates that ϕ is true at all positions of a word. More expressiveness can be achieved by combining the mentioned operators. For example, $\Diamond\Box\phi$ means that ϕ will eventually become true and then remain true forever, while $\Box\Diamond\phi$ means that ϕ is true infinitely often.

Given a finite transition system $T = (Q, Q_0 \rightarrow, \Pi, \vDash)$ and a formula ϕ over Π, checking whether the words of T starting from a region X satisfy ϕ (written as $T(X) \vDash \phi$) is called *model checking* [1]. If we denote by L_ϕ the set of all words (language) satisfying ϕ, then model checking means deciding the language inclusion $L_T(X) \subseteq L_\phi$. We also say that a transition system satisfies a formula ($T \vDash \phi$) if and only if $T(Q_0) \vDash \phi$.

If $T/_\sim$ is a quotient of T, then for any equivalence class $X \in Q_0/_\sim$ and formula ϕ, we have:

$$T/_\sim(X) \vDash \phi \Rightarrow T(X) \vDash \phi. \tag{3}$$

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

$$T/_\sim(X) \vDash \phi \Leftrightarrow T(X) \vDash \phi \tag{4}$$

Properties (3) and (4) allow one to model check finite quotients and extend the results to the (possibly infinite) original transition system.

3 Problem Formulation

Let $\mathcal{X}, \mathcal{X}_l,\ l \in L$ be a set of open polytopes in $\mathbb{R}^N$ and $\mathcal{P}_l$ be a set of open polytopes in $\mathbb{R}^{(N^2+N)}$, 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 $\bigcup_{l \in L} cl(\mathcal{X}_l) = cl(\mathcal{X})$, where $cl(\mathcal{X})$ is the closure of $\mathcal{X}$.

A discrete-time continuous-space piecewise affine (PWA) system Σ with polytopal parameter uncertainty is defined as:

$$\Sigma\ :\ x_{k+1} = A(p)x_k + b(p),\ x_0 \in \mathcal{X}_{in},\ x_k \in \mathcal{X}_l,\ p \in \mathcal{P}_l,\ l \in L,\ k = 0, 1, , \ldots \tag{5}$$

where $\mathcal{X}_{in} \subseteq \mathcal{X}$ is a set of initial conditions and $\mathcal{P}_l$ is the allowed set of parameters in region $l \in L$. The linear functions $A : \mathbb{R}^{(N^2+N)} \rightarrow \mathbb{R}^{N \times N}$ and $b : \mathbb{R}^{(N^2+N)} \rightarrow \mathbb{R}^{N \times 1}$ simply take the first N^2 and the last N components of $p \in \mathbb{R}^{(N^2+N)}$ and form a $N \times N$ matrix and $N \times 1$ vector, respectively.

$\mathcal{X}$ is assumed to be an invariant for the trajectories of Σ under all values of the parameters. We are interested in studying properties of trajectories of system (5) specified in terms of a set of linear predicates of the form

$$\Pi = \{\pi_i \,|\, \pi_i :\ a_i^T x + b_i < 0,\ i = 1, \ldots, K\}, \tag{6}$$

where $x, a_i \in \mathbb{R}^N$ and $b_i \in \mathbb{R}, i = 1, \ldots, K$. Without loss of generality, we assume that the set of initial states $\mathcal{X}_{in}$ from the definition of the PWA system (5) is a union of polytopes from the set of polytopes determined by the regions $\mathcal{X}_l$ and the linear predicates from Π (if this is not the case, more linear predicates can be added to Π in Equation (6)).

Informally, the semantics of system (5) can be understood in the following sense: a trajectory $x_0x_1x_2\ldots$ of the system can be obtained by selecting an initial condition $x_0 \in \mathcal{X}_{in}$, finding an $l \in L$, such that $x_0 \in \mathcal{X}_l$, selecting a parameter $p \in \mathcal{P}_l$, applying the affine map of Equation (5) and repeating this procedure for each subsequent step. A trajectory produces a word $w_0w_1w_2\ldots$, where each $w_i \in 2^{\Pi}$ lists the propositions from Π which are satisfied by x_i. Then, such words can be checked against satisfaction of LTL formula ϕ over Π. A formal definition is given in Section 4 through an embedding transition system. We consider the following problem:

Problem 1. Given a PWA system (Equation (5)) and an LTL formula ϕ over a set of linear predicates Π (Equation (6)), find sets of parameters such that all the trajectories of the system satisfy the formula, under all identified parameters.

In other words, we are interested in excluding parameters from the allowed sets for each region, for which the formula is not satisfied. As it will become clear later, for each region $l \in L$, the solution will be in the form of a union of disjoint open subpolytopes of the allowed polytope $\mathcal{P}_l$.

To provide a solution to Problem 1, we first embed the PWA system (5) into an infinite transition system T_e. By using the equivalence classes induced by the predicates from (6), we then construct a finite overapproximation quotient transition system $\overline{T_e/\sim}$ whose language includes the language of T_e (see Section 4). We then use model checking to cut transitions from $\overline{T_e/\sim}$ (see Section 5.1) and, correspondingly, sets of parameters from (5) (see Section 5.2), until all its trajectories satisfy the formula. Alternatively, in Section 6, we propose a method for the direct construction of a bisimulation quotient. In both approaches, our method is conservative, as it will become clear later.

Remark 1. There are several simplifying assumptions that we make in the formulation of Problem 1. First, we assume that the polytope $\mathcal{X}$ is an invariant for all trajectories of (5). However, this assumption is not restrictive, since $\mathcal{X}$ can be assumed large enough to contain all possible state values in a particular process. Second, we assume that the predicates in Equation (6) are given over strict inequalities, and only the reachability of open full dimensional polytopes is captured in the semantics of the embedding and of the quotients[1]. However, this seems to be enough for practical purposes, since only sets of measure zero are disregarded, and it is unreasonable to assume that equality constraints can be detected in a real world application.

[1] Throughout the rest of the paper, unless clearly specified, we refer to an "open polytope" simply as "polytope."

4 Construction of Finite Quotients

To formally define the satisfaction of a formula ϕ over Π by the PWA system (5), we embed it into a transition system:

Definition 3. *An embedding transition system for (5) and the set of predicates Π can be defined as $T_e = (Q_e, Q_{0e}, \rightarrow_e, \Pi_e, \models_e)$, where*

- $Q_e = \bigcup_{l \in L} \mathcal{X}_l$,
- $Q_{0e} = \mathcal{X}_{in}$,
- $(x, x') \in \rightarrow_e$ *if and only if there exist* $l \in L$, $x \in \mathcal{X}_l$ *and* $p \in \mathcal{P}_l$ *such that* $x' = A(p)x + b(p)$,
- $\Pi_e = L \bigcup \Pi$,
- $\models_e$ *is defined as follows: if* $\pi = l \in L$, *then* $x \models_e \pi$ *if and only if* $x \in \mathcal{X}_l$; *if* $\pi = \pi_i \in \Pi$, *then* $x \models_e \pi$ *if and only if* $a_i^T x + b_i < 0$,

Definition 4. *Given a subset $X \subseteq Q_{0e}$, we say that all trajectories of system (5) originating in X satisfy formula ϕ if and only if $T_e(X)$ satisfies ϕ (as defined in Section 2.1).*

The embedding transition system T_e has infinitely many states and cannot be model checked directly. Given a polytopal, proposition preserving equivalence relation $\sim$ on Q_e, one can try to construct (and model check) the quotient transition system $T_e/_\sim = \{Q_e/_\sim, Q_{0e}/_\sim, \rightarrow_{e\sim}, \Pi_e, \models_e\}$ (see Section 2.1). The construction of the states $Q_e/_\sim$ and initial states $Q_{0e}/_\sim$ amounts to checking the non-emptiness of polyhedral sets and intersections of polyhedral sets, respectively. Satisfaction of each state is induced directly from the equivalence relation. If the $Post()$ (or $Pre()$) operator can be computed, transitions in the quotient can be assigned as follows: $(X_i, X_j) \in \rightarrow_{e\sim}$ if and only if $Post(X_i) \cap X_j \neq \emptyset$ (or $Pre(X_j) \cap X_i \neq \emptyset$).

In our previous work [15], we focused on PWA systems with fixed parameters (*i.e.,* $\mathcal{P}_l$ in Equation (5) were singletons), and showed that for the propositional equivalence relation $\sim$, the quotient $T_e/_\sim$ can be efficiently constructed, based on the computation of the $Pre()$ operator, which was a polyhedral set. Moreover, we showed that all the steps in the "bisimulation algorithm" for the iterative construction of simulation quotients leading to the coarsest bisimulation quotient [20,21] (if one exists) are implementable.

Under parameter uncertainty, $T_e/_\sim$ cannot always be constructed, since in general, there are no algorithms capable of exact computations of $Pre()$ or $Post()$ operators. In fact, it can be proven that when parameters are allowed to vary in polyhedral sets, both operators might return a non-convex set even when applied to a polyhedral set [22].

If we denote by $Post_e()$ the "$Post()$" operator of the embedding transition system T_e, then from Equation (1) and Definition 3, for an arbitrary polytope $X \subseteq \mathcal{X}_l, l \in L$ we have

$$Post_e(X) = \{x' \in \mathbb{R}^N \mid \exists p \in \mathcal{P}_l,\ \exists x \in X \text{ such that } x' = A(p)x + b(p)\} \quad (7)$$

Proposition 1. *A polyhedral overapproximation of $Post_e(X)$ is given by*

$$\overline{Post_e(X)} = Conv\{A(w)v + b(w), w \in \mathcal{V}(\mathcal{P}_l), v \in \mathcal{V}(X)\} \tag{8}$$

where $\mathcal{V}(X)$ and $\mathcal{V}(\mathcal{P}_l)$ denote the sets of vertices of $\mathcal{X}$ and $\mathcal{P}_l$, respectively.

Proof. See `http://iasi.bu.edu/~yordanov/papers/HSCC2008full.pdf`.

Similarly to [23], we use the smallest convex set containing $Post_e()$ as an overapproximation. Although a precise distance between the real set and its overapproximation has not been determined, it has been established through extensive simulation that in general, the volume of $Post_e()$ is not significantly increased by the approximation.

By using $\overline{Post_e(X)}$ instead of the regular $Post_e()$ operator, transitions in an overapproximation quotient can be efficiently obtained as described for the general case.

Formally, if $\sim$ is the propositional equivalence relation, the overapproximation quotient can be given as $\overline{T_e/\sim} = \{Q_e/\sim, Q_{0e}/\sim, \overline{\rightarrow_{e\sim}}, \Pi_e, \models_e\}$, where $\rightarrow_{e\sim} \subseteq \overline{\rightarrow_{e\sim}}$. This implies that

$$L_{T_e} \subseteq L_{T_e/\sim} \subseteq L_{\overline{T_e/\sim}} \tag{9}$$

and therefore the overapproximation quotient simulates the exact quotient and the embedding system. As a result, model checking can be performed on the overapproximation quotient and satisfaction of a formula can be extended to the embedding.

5 Counterexample Guided Parameter Synthesis

In Section 4 we showed that an overapproximation quotient $\overline{T_e/\sim}$ can be constructed, and all operations involved are computable. In this section, we propose to use LTL model checking to "cut" transitions from $\overline{T_e/\sim}$ until we obtain a transition system $\overline{T_e/\sim}^{\phi}$ satisfying the formula. Then we go back to the initial system (5) and remove parameter values such that the language of the new embedding transition system is included in the language of $\overline{T_e/\sim}^{\phi}$, which guarantees the satisfaction of the formula by the PWA system (5).

5.1 Construction of Satisfying Quotients

Using our LTL model checker described in [4], we start by searching for a shortest run[2] of $\overline{T_e/\sim}$ satisfying the negation of the formula $\neg\phi$. If such a run exists, then we eliminate it by removing one of its transitions. Then we reiterate the process until we obtain the transition system $\overline{T_e/\sim}^{\phi}$ satisfying ϕ.

[2] A standard representation of an infinite run includes finite prefix and suffix, where the suffix is repeated an infinite number of times. The length of a run is defined as the sum of the lengths of the prefix and suffix.

Since, in general, several different transitions are taken during the generation of a counterexample, removing any one of them will remove the counterexample from the language of the quotient. It is impossible to determine which transition's removal will lead to a solution (or to the best solution when more than one exists). Therefore, in this paper, we exhaustively generate all solutions. This process can be seen as generating a tree, having the initial finite quotient as its root. Each child node in the tree represents a quotient that has the same set of states as the parent, but only a subset of its transitions. The children for each node are generated by removing one different transition, appearing in the shortest counterexample, from the parent.

When transitions are removed, a state of the quotient might become blocking, resulting in the appearance of finite words in its language. Since the semantics of LTL are defined only over infinite ω-words, we make all blocking states unreachable by removing all their incoming transitions. It is also possible that one or more of the initial states become blocking, in which case we ignore the corresponding quotient (further removal of transitions will not lead to a solution).

A leaf node in the tree represents a quotient for which computation stopped, since no additional counterexamples can be generated. The quotients represented by such nodes satisfy the LTL formula, since their languages are nonempty (all initial states are non-blocking), do not contain finite words (no blocking states are reachable), and have an empty set of counterexamples.

5.2 Parameter Synthesis

The finite quotient $T_e/\sim$ is constructed so that it captures all possible transitions of the embedding T_e. By Definition 3, transitions are included in the embedding if and only if appropriate parameters for such a transition are allowed. Therefore, we can relate the transitions present in the finite quotient to sets of allowed parameters for the PWA system.

Definition 5. *Given two polytopes X and Y in $\mathbb{R}^N$, the set of parameters $P^{X \not\to Y}$, for which the image of X does not have an intersection with Y, is defined as:*

$$P^{X \not\to Y} = \{p \in \mathbb{R}^{(N^2+N)} \mid A(p)x + b(p) \notin Y \text{ for all } x \in X\} \tag{10}$$

Proposition 2. *Let X and Y be polytopes in $\mathbb{R}^N$ given in V-representation as $X = Conv\{v_1, \ldots v_m\}$ and H-representation as $Y = \{x \in \mathbb{R}^N \mid c_i^T x + d_i < 0,\ i = 1, \ldots, n\}$, respectively. Then,*

$$\underline{P^{X \not\to Y}} = \bigcup_{i=1}^{n} \{p \in \mathbb{R}^{(N^2+N)} \mid c_i^T(A(p)v_j + b(p)) + d_i > 0, \text{ for all } j = 1, \ldots, m\}$$

*is an underapproximation of $P^{X \not\to Y}$ (*i.e., $\underline{P^{X \not\to Y}} \subseteq P^{X \not\to Y}$*)*

Proof. See `http://iasi.bu.edu/~yordanov/papers/HSCC2008full.pdf`.

In other words, a conservative underapproximation $\underline{P^{X \not\to Y}}$ of $P^{X \not\to Y}$ can be obtained as the union of polyhedral sets from the V-representation of X and the H-representation of Y.

We use the underapproximation from Proposition 2 to find sets of parameters for each region $l \in L$, such that, for each node of the tree described in Section 5.1, the corresponding PWA system is simulated by the quotient transition system at that node. Specifically, for two polytopes $X \subseteq \mathcal{X}_l$ and Y, if the parameters in region $l \in L$ are restricted to the set $\mathcal{P}_l \cap \underline{P^{X \not\to Y}}$, then, by Proposition 2, the transition $x \to_e y$ will not appear in the embedding T_e, for any $x \in X$ and $y \in Y$. This means that, in the corresponding quotient, the transition $X \to_{e\sim} Y$ will not exist. As already stated, we cannot compute $T_e/_\sim$. However, by restricting the parameters as described above, we can ensure that, at every node of the tree constructed in Section 5.1, the PWA system with restricted parameters is simulated by the quotient transition system at that node. As previously stated, the leaf nodes of the computation tree contain quotients satisfying the formula and their corresponding PWA systems provide a solution to Problem 1.

Because of the overapproximation used in the construction of the quotient, a spurious transition might appear in place of a deleted one ($(X, Y) \in \overline{\to_e \sim}$ but $(X, Y) \notin \to_e \sim$). We prevent this by enforcing that a deleted transition never reappears in the quotient. Additionally, the structure of PWA systems allows different polytopes (determined by the set of linear predicates) to share the same sets of parameters, and therefore, it is possible that other transitions are removed from the quotient besides the target one. To account for this, we reconstruct the quotient every time parameters are cut. If, during the removal of parameters, a set $\mathcal{P}_l$ becomes empty, then we embed all polytopes from region $l \in L$ as blocking states, and make them unreachable.

Given only the purely discrete problem of modifying a quotient to satisfy a formula by taking a subset of its transitions, our approach is guaranteed to terminate, finding a solution when one exists, as it is exhaustive and follows a tree of size limited by the total number of transitions in the initial quotient. In the combined problem of transition and parameter removal, computation will still terminate but a potential solution might be missed due to the approximations. If a solution is found, however, it is guaranteed to be correct.

Going back to the tree construction from Section 5.1, in general, our algorithm will produce more than one solution (each leaf corresponds to a satisfying transition system). Selecting the "best" solution is a non-trivial problem, and might depend on the application. For example, it is possible to introduce additional constraints (such as requiring that a particular transition is present in the solution) or compare total number of transitions of the solutions, since more reachable states from the initial one with more transitions result in a richer language. In the case study presented at the end of this paper, we chose the latter.

Our solution to Problem 1 is summarized in Algorithm 1.. In order to prevent unnecessary computation, we first check the system from each initial state. If there exists an initial state from which the negation of the formula is satisfied,

then there are no satisfying trajectories originating there, so a solution will not be found by refining the transitions (and corresponding parameters). As stated earlier, the algorithm is guaranteed to terminate, as its execution follows at tree of finite size.

Algorithm 1. Obtain subsets of parameters for a PWA system Σ_{in} such that an LTL formula ϕ is satisfied.

$T_{sat} = \emptyset$;
Construct $\overline{T_e/\sim}$ from the initial PWA system Σ_{in};
for each $X \in Q_{0e}/\sim$ **do**
  if $\overline{T_e/\sim}(X) \vDash \neg\phi$ **then**
    return $\emptyset$;
  end if
end for
$T_{all} = \{(\Sigma_{in}, \overline{T_e/\sim})\}$;
while $T_{all} \neq \emptyset$ **do**
  for each pair $(\Sigma, T) \in T_{all}$ **do**
    $T_{all} = T_{all} \setminus (\Sigma, T)$;
    generate the shortest counter-example c for T and formula ϕ;
    if $c = \emptyset$ and $L_T \neq \emptyset$ **then**
      add (Σ, T) to T_{sat};
    else
      for each transition $X \to Y$ of counterexample c **do**
        Find $\mathcal{X}_l$ such that $X \subseteq \mathcal{X}_l$;
        Construct Σ' from Σ by setting $\mathcal{P}'_l = \mathcal{P}_l \cap \underline{P^{X \not\to Y}}$;
        Reconstruct quotient T' from Σ' and ensure no previously removed transitions reappear;
        Make blocking states of T' unreachable
        if initial states in T' are non-blocking **then**
          add (Σ', T') to $\{T_{all}\}$
        end if
      end for
    end if
  end for
end while
return $\{T_{sat}\}$;

Both the number of states and transitions in the embedding $\overline{T_e/\sim}$ contribute to the complexity of Algorithm 1.. A high dimensional system with many regions of different dynamics and propositions would be embedded with a high number of states. This, together with the complexity of the LTL formula affects the time required to perform model checking on the system. The number of transitions in the original embedding, on the other hand, depends on the dynamics of the system and determines how many times model checking must be performed, since the execution of the algorithm follows a finite tree described in Section 5.1. As a result, Algorithm 1. can perform well even on high dimensional systems, as long as the total number of transitions is low.

6 Construction of Bisimulation Quotients

In this section we show that if the parameters of the PWA system (Equation (5)) are restricted to appropriate subsets, an exact finite bisimulation quotient can be constructed without extensive iterative computation. Subsequently, satisfiability of an LTL formula by the original PWA system can be proven if model checking this quotient with the negation of the formula produces a counterexample. Of course, by limiting the sets of parameters, certain transitions might disappear from the system and, therefore, the richness of its language might be diminished.

Definition 6. *Given two polytopes X and Y in $\mathbb{R}^N$, the set of parameters for which the image of X is completely included in Y is defined as:*

$$P^{X \to Y} = \{p \in \mathbb{R}^{(N^2+N)} \mid A(p)x + b(p) \in Y \text{ for all } x \in X\} \tag{11}$$

Proposition 3. *Let X and Y be polytopes in $\mathbb{R}^N$ given in the V-representation as $X = Conv\{v_1, \dots v_m\}$ and in the H-representation as $Y = \{x \in \mathbb{R}^N \mid c_i^T x + d_i < 0,\ i = 1, \dots, n\}$, respectively. Then*

$$P^{X \to Y} = \{p \in \mathbb{R}^{(N^2+N)} \mid c_i^T(A(p)v_j + b(p)) + d_i < 0,\ i = 1, \dots, n,\ j = 1, \dots, m\}$$

Proof. See `http://iasi.bu.edu/~yordanov/papers/HSCC2008full.pdf`.

In other words, the polyhedral set of parameters $P^{X \to Y}$ can be computed immediately from the V-representation of X and the H-representation of Y.

Proposition 4. *If in each location $l \in L$, the parameters of the PWA system (5), are restricted to $\mathcal{P}_l \cap (\bigcup_{i \in L} P^{\mathcal{X}_l \to \mathcal{X}_i})$, then the propositional equivalence quotient $T_e/\sim$ is a bisimulation quotient, and it is computable.*

Proof. The proof for bisimulation follows immediately from Definitions 2, 3 and Proposition 3. On the computation of the quotient, the equivalence classes are computed as above, and a transition $(X, Y) \in \to_{e\sim}$ exists if and only if $X \subseteq \mathcal{X}_l$ and $\mathcal{P}_l \cap P^{X \to Y} \neq \emptyset$.

7 Analysis of a Genetic Toggle Switch

We illustrate the proposed method by analyzing the genetic network shown in Figure 1 (A). The system is described by a two dimensional discrete time PWA model, using ramp functions to represent gene regulation. A ramp function is defined by two threshold values, which induce three regions of different dynamics. At low concentrations of repressor (below threshold 1) the regulated gene is fully expressed, while at high repressor concentrations (above threshold 2) expression is only basal. For concentrations between the two thresholds expression is graded. Since there are two repressors, two ramp functions are used and, therefore, the system has a total of nine rectangular invariants. We use $L = \{1, 2, \dots, 9\}$ as a

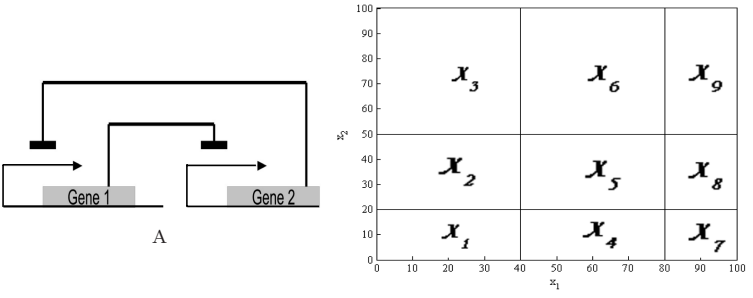

Fig. 1. (A) A genetic switch consisting of two mutual repressors. High levels of one of the products shut down the expression of the other gene. (B) Invariants of the system determined by ramp functions, describing gene regulation.

set of labels and $\{\mathcal{X}_1, \ldots, \mathcal{X}_9\}$ and $\{\mathcal{P}_1, \ldots, \mathcal{P}_9\}$ to denote the invariants (Figure 1 (B)) and parameters of the system, respectively.

We are interested in analyzing the behavior of the system when it is initialized with low concentrations of both genes. To specify this, we introduce two propositions $\mathit{\Pi} = \{\pi_1, \pi_2\}$, where $\pi_1 = \{x_1 - 10 < 0\}$ and $\pi_2 = \{x_2 - 10 < 0\}$ and we set $\mathcal{X}_{in}$ as the region where both predicates are satisfied ($\mathcal{X}_{in} = X_1$ in Figure 2 (A)). We assume hyper-rectangular parameter sets and, by using Proposition 3, we restrict the parameters for each region $l \in L$ to subsets of $P^{\mathcal{X}_l \to \mathcal{X}}$, ensuring that $\mathcal{X}$ is an invariant. The parameter ranges of the system for all regions are available at `http://iasi.bu.edu/~yordanov/papers/HSCC2008full.pdf`.

First, we apply the method outlined in Section 6 in order to modify the parameters of the system and obtain a bisimulation quotient directly. The parameter ranges, computed by the algorithm, are available at `http://iasi.bu.edu/~yordanov/papers/HSCC2008full.pdf` and the resulting bisimulation quotient is shown in Figure 2 (A). As expected, some transitions of the system are lost when parameters are restricted to smaller sets, but a lot of its behavior is captured by the quotient. Due to the language equivalence with the initial PWA system, inherent to the bisimulation quotient, it could provide an useful tool for the analysis of the system.

Next, we apply the approach of Section 5.2 and find subsets of the parameters for each region of the system, such that the property "eventually gene 2 is expressed in high concentrations, while gene 1 is expressed only basally" is always satisfied. For this, we use the same initial PWA model as before. During the execution of Algorithm 1. a number of transitions are removed from the quotient by removing appropriate sets of parameters of the system. The quotient corresponding to a solution, obtained as a leaf node in the computation tree (see Section 5.2) is shown in Figure 2 (B). Regions of parameters for the PWA system obtained as a solution to problem 1 are available at `http://iasi.bu.edu/~yordanov/papers/HSCC2008full.pdf`.

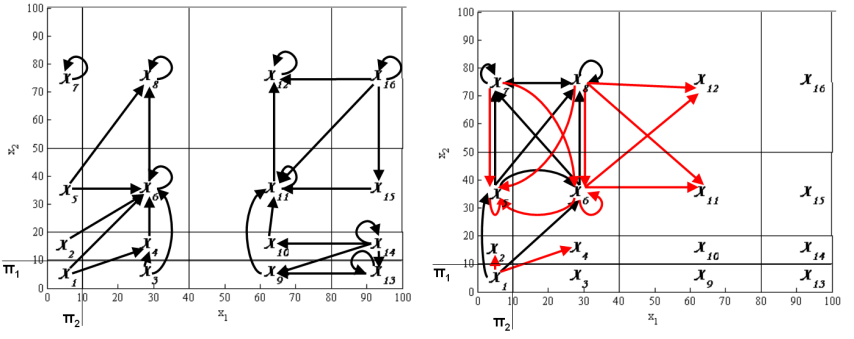

Fig. 2. (A) Graphical representations of the bisimulation quotient obtained from the PWA model. (B) Satisfying simulation quotient. Only transitions for reachable states are shown. Transitions shown in red were eliminated by the algorithm.

8 Conclusion

We showed that an iterative procedure can be used to efficiently obtain subsets of parameters for a PWA system, such that an LTL formula is satisfied. Our method relied on the computation of finite overapproximation simulation quotients and generation of counterexamples. Additionally, we described an approach for the synthesis of parameters, such that a bisimulation quotient can be constructed for the system without extensive computation. We applied our methods to a PWA model of a genetic switch and in the future plan to focus on models of gene networks constructed from experimental data.

References

1. Clarke, E.M., Peled, D., Grumberg, O.: Model checking. MIT Press, Cambridge (1999)
2. Davoren, J., Coulthard, V., Markey, N., Moor, T.: Non-deterministic temporal logics for general flow systems. In: Alur, R., Pappas, G.J. (eds.) HSCC 2004. LNCS, vol. 2993, pp. 280–295. Springer, Heidelberg (2004)
3. Tabuada, P., Pappas, G.: Model checking LTL over controllable linear systems is decidable. In: Maler, O., Pnueli, A. (eds.) HSCC 2003. LNCS, vol. 2623, Springer, Heidelberg (2003)
4. Kloetzer, M., Belta, C.: A fully automated framework for control of linear systems from LTL specifications. In: Hespanha, J.P., Tiwari, A. (eds.) HSCC 2006. LNCS, vol. 3927, pp. 333–347. Springer, Heidelberg (2006)
5. Loizou, S.G., Kyriakopoulos, K.J.: Automatic synthesis of multiagent motion tasks based on LTL specifications. In: 43rd IEEE Conference on Decision and Control (2004)
6. Fainekos, G.E., Kress-Gazit, H., Pappas, G.J.: Hybrid controllers for path planning: a temporal logic approach. In: Proceedings of the 2005 IEEE Conference on Decision and Control (2005)

7. Antoniotti, M., Park, F., Policriti, A., Ugel, N., Mishra, B.: Foundations of a query and simulation system for the modeling of biochemical and biological processes. In: Proceedings of the Pacific Symposium on Biocomputing, pp. 116–127 (2003)
8. Batt, G., Ropers, D., de Jong, H., Geiselmann, J., Mateescu, R., Page, M., Schneider, D.: Validation of qualitative models of genetic regulatory networks by model checking: Analysis of the nutritional stress response in *Escherichia coli*. Bioinformatics 21(Suppl. 1), i19–i28 (2005)
9. Batt, G., Belta, C., Weiss, R.: Model checking genetic regulatory networks with parameter uncertainty. In: Bemporad, A., Bicchi, A., Buttazzo, G. (eds.) HSCC 2007. LNCS, vol. 4416, pp. 61–75. Springer, Heidelberg (2007)
10. Juloski, A.L., Heemels, W., Ferrari-Trecate, G., Vidal, R., Paoletti, S., Niessen, J.: Comparison of four procedures for the identification of hybrid systems. In: Morari, M., Thiele, L. (eds.) HSCC 2005. LNCS, vol. 3414, pp. 354–369. Springer, Heidelberg (2005)
11. Lin, J.N., Unbehauen, R.: Canonical piecewise-linear approximaions. IEEE Transactions on Circuits and Systems - I: Fundamental Theory and Applications 39(8), 697–699 (1992)
12. Heemels, W.P.M.H., Schutter, B.D., Bemporad, A.: Equivalence of hybrid dynamical models. Automatica 37(7), 1085–1091 (2001)
13. Alur, R., Henzinger, T.A., Lafferriere, G., Pappas, G.J.: Discrete abstractions of hybrid systems. Proceedings of the IEEE 88, 971–984 (2000)
14. Pappas, G.J.: Bisimilar linear systems. Automatica 39(12), 2035–2047 (2003)
15. Yordanov, B., Batt, G., Belta, C.: Model checking discrete-time piecewise affine systems: application to gene networks. In: European Control Conference (2007)
16. Clarke, E., Fehnker, A., Han, Z., Krogh, B., Ouaknine, J., Stursberg, O., Theobald, M.: Abstraction and counterexample-guided refinement in model checking of hybrid systems. International Journal of Foundations of Computer Science 14(4), 583–604 (2003)
17. Bernot, G., Comet, J.P., Richard, A., Guespin, J.: Application of formal methods to biological regulatory networks: Extending Thomas' asynchronous logical approach with temporal logic. Journal of Theoretical Biology 229(3), 339–347 (2004)
18. Chabrier-Rivier, N., Chiaverini, M., Danos, V., Fages, F., Schächter, V.: Modeling and querying biomolecular interaction networks. Theoretical Comp. Science 325(1), 25–44 (2004)
19. Eker, S., Knapp, M., Laderoute, K., Lincoln, P., Talcott, C.: Pathway logic: Executable models of biological networks. Electronic Notes in Theoretical Computer Science 71 (2002)
20. Bouajjani, A., Fernandez, J.C., Halbwachs, N.: Minimal model generation. In: Clarke, E., Kurshan, R.P. (eds.) CAV 1990. LNCS, vol. 531, pp. 197–203. Springer, Heidelberg (1991)
21. Kanellakis, P.C., Smolka, S.A.: CCS expressions, finite-state processes, and three problems of equivalence. Inform. Computat. 86, 43–68 (1990)
22. Habets, L.: Personal communication. Eindhoven University of Technology (2007)
23. Barmish, B.R., Sankaran, J.: The propagation of parametric uncertainty via polytopes. IEEE Transactions on automatic control 24, 346–349 (1979)