2008 American Control Conference
Westin Seattle Hotel, Seattle, Washington, USA
June 11-13, 2008

Formal Analysis of Piecewise Affine Systems under Parameter Uncertainty with Application to Gene Networks

Boyan Yordanov and Calin Belta

Abstract— **In this paper, we consider discrete-time continuous-space Piecewise Affine (PWA) systems with uncertain parameters, and study temporal logic properties of their trajectories. Specifically, given a PWA system with polyhedral parameter uncertainties and a Linear Temporal Logic (LTL) formula over linear predicates in its state variables, we attempt to find the largest region of initial states from which all trajectories of the system satisfy the formula. Our method is based on the iterative computation and model checking of finite transition systems simulating the original PWA system. We illustrate our method by computing the basins of attraction for the two equilibria of a PWA model of a two-gene network.**

Index Terms— **gene networks, piecewise affine systems, parameter uncertainty, abstraction, formal analysis.**

I. INTRODUCTION

Temporal logics and model checking [10] 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 [11], control of linear systems from temporal logic specifications [21], [17], task specification and controller synthesis in mobile robotics [19], [12] and specification and analysis of qualitative behavior of genetic circuits [2], [3], [4].

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. For such applications, states of the system usually represent concentrations of species (mRNA, proteins and other molecules). Additionally, for most gene networks, dynamics are affine, with the exception of gene regulation function (GRF), which captures the effect of concentrations of various transcription factors on the activity of genes they control. Recent experimental techniques, based of fluorescent reporter genes [13], allow for the collection of a large amount of input-output data relating transcription factor concentrations and gene activity. There also exist computationally attractive techniques for the identification of PWA models from such data, which include Bayesian methods, bounded-error procedures, clustering-based methods, Mixed-Integer Programming, and algebraic geometric methods (see [16] for a review). By combining experimental and computational techniques, PWA models of gene networks can be efficiently obtained.

PWA systems are also quite general, since they can approximate nonlinear dynamics with arbitrary accuracy, and are proven to be equivalent with several other classes of hybrid systems [15]. Even so, a PWA system with fixed parameters might not provide a good model. 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 can capture the rich behavior of systems under a range of conditions, a PWA system 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 gene networks are naturally expressed in Linear Temporal Logic (LTL) [10] formulas over linear predicates in the state variables. Examples include remaining within certain allowed concentration ranges (invariance), getting to certain target concentrations (reachability) or avoiding certain dangerous concentrations (safety). We are interested in analyzing PWA models with uncertain parameters using properties described as LTL formulas. Specifically, given a system in a bounded polytope $\mathcal{X}$ in $\mathbb{R}^N$ and an LTL formula ϕ, we attempt to find the largest region $\mathcal{X}_\phi \subseteq \mathcal{X}$ such that all trajectories of the system originating in $\mathcal{X}_\phi$ satisfy the property expressed by ϕ, no matter what (allowed) parameters the system takes on. However, PWA systems have an infinite number of states, and model checking cannot be used directly. We use the polytopes from the definition of the PWA system and the linear predicates in the formula to define equivalence classes, and model check the produced quotients. Our method is iterative, and based on the notions of transition system, simulation, and bisimulation [10].

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 [1] for a review), and is closely related to [20], [21], [17]. Unlike counterexample guided refinement [9], which eliminates spurious runs in the abstraction, our approach relays on iterative refinement. Although the main problem is similar to the focus of our previous work [22], the methods are significantly different, as a result of the added parameters uncertainty. One of the main contributions of this paper is to

B. Yordanov is with the Department of Biomedical Engineering, Boston University, Boston, MA 02215, USA yordanov@bu.edu

C. Belta is with the Center for Information and Systems Engineering, Boston University, Brookline, MA 02446, USA cbelta@bu.edu

This work was partially supported by NSF 0432070 and NSF CAREER 0447721 at Boston University.

show that, even in this case, all the steps of our solution are computable for PWA systems, and can be reduced to operations on polyhedral sets, which can be performed efficiently [18]. Additionally, by using an in-house model checker [17] rather than off-the-shelf tool, we manage to integrate polyhedral operations and model checking efficiently. The embedding of discrete-time systems into transition systems is inspired from [20], [21]. However, while the focus there is on characterizing the existence of bisimulation quotients, in this work we focus on computation and refinement of simulation and bisimulation quotients, while including the model checking process.

From an application point of view, this paper relates to [3], [5], [7], [4], where temporal logics are used to specify properties of biomolecular networks. These works aim at checking whether a system satisfies dynamical properties for given (sets of) initial conditions. In contrast, we search for the largest set of initial conditions for which the given properties are satisfied, yielding more informative results instead of simple Yes/No answers.

II. PRELIMINARIES

A. Transition Systems, Simulations, and Bisimulations

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

A transition $(q, q') \in \rightarrow$ is also denoted by $q \rightarrow q'$. The transition system T is *finite* if its set of states Q is finite, *non-blocking* if, for every state $q \in Q$, there exists $q' \in Q$ such that $(q, q') \in \rightarrow$ and *deterministic* if, for all $q \in Q$, there exists at most one $q' \in Q$ such that $(q, q') \in \rightarrow$.

For an arbitrary state $q \in Q$, we define $\Pi_q = \{\pi \in \Pi \,|\, q \vDash \pi\}$, $\Pi_q \in 2^{\Pi}$ as the set of all atomic propositions satisfied at q. A *trajectory* or *run* of T starting from q 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 $q \in Q$ is called the *language* of T originating at q and is denoted by $L_T(q)$. A subset X of the state set Q ($X \subseteq Q$) is called a *region* of T. The set of all words generated by all runs of T originating at all states of X is called the language of T originating at X and is denoted by $L_T(X)$.

For an arbitrary region X, we define the set of states $Pre(X)$ that reach X in one step as

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

Similarly, we define the set of states $Post(X)$ that can be reached from X in one step as

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

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* $T/_\sim = (Q/_\sim, \rightarrow_\sim, \Pi, \vDash_\sim)$. $Q/_\sim$ is the quotient space (the set of all 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

$$L_T(X) \subseteq L_{T/_\sim}(X), \tag{3}$$

for any $X \in Q/_\sim$ (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 $T = (Q, \rightarrow, \Pi, \vDash)$ 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 $T/_\sim \backsimeq T$. An immediate consequence of bisimulation is language equivalence, *i.e.,* $L_T(X) = L_{T/_\sim}(X)$, for all $X \in Q/_\sim$.

B. Linear Temporal Logic and Model Checking

To specify temporal logic properties for trajectories of PWA systems, in this paper we use Linear Temporal Logic [10]. 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 the power set 2^{Π} of 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 $T = (Q, \rightarrow, \Pi, \vDash)$ 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*. 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 ($T \vDash \phi$) from X if and only if $T(X) \vDash \phi$.

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

$$T/_\sim(X) \vDash \phi \Rightarrow T(X) \vDash \phi. \quad (4)$$

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

$$T/_\sim(X) \vDash \phi \Leftrightarrow T(X) \vDash \phi \quad (5)$$

Properties (4) and (5) (which follow immediately from (3)) allow one to model check finite quotients and extend the results to the (possibly infinite) original transition system.

Definition 3: A region $X \subseteq Q$ of a transition system T is the *largest* region that satisfies a formula ϕ if and only if $q \in X \Leftrightarrow T(q) \vDash \phi$. In other words, if X is the largest satisfying region for ϕ, then there is at least one trajectory violating ϕ that originates from each state outside of X.

III. PROBLEM FORMULATION

Let $\mathcal{X}, \mathcal{X}_l, l \in L$ 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 $\bigcup_{l \in L} \bar{\mathcal{X}}_l = \bar{\mathcal{X}}$, where $\bar{\mathcal{X}}$ is the closure of $\mathcal{X}$.

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

$$x_{k+1} = A_l x_k + b_l,\ x_k \in \mathcal{X}_l,\ l \in L,\ k = 0, 1, 2, \ldots \quad (6)$$

where parameters A_l and b_l are uncertain, but known to belong to polytopal uncertainty sets $\mathcal{P}_l^A \subseteq \mathbb{R}^{N \times N}$ and $\mathcal{P}_l^b \subseteq \mathbb{R}^N$, respectively.

$\mathcal{X}$ is assumed to be an invariant for the trajectories of (6) under all values of the parameters. We are interested in studying properties of trajectories of system (6) specified in terms of a set of linear predicates of the form

$$\Pi = \{\pi_i \,|\, \pi_i :\ c_i^T x + d_i < 0,\ i = 1, \ldots, K\}, \quad (7)$$

where $x, c_i \in \mathbb{R}^N$ and $d_i \in \mathbb{R}$. Informally, the semantics of system (6) can be understood in the following sense: a trajectory $x_0 x_1 x_2 \ldots$ produces a word $w_0 w_1 w_2 \ldots$, 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 formulas ϕ over Π. A formal definition is given in Section V through an embedding transition system.

Specifically, we consider the following problem:

Problem 1: Given a discrete-time continuous-space piecewise affine system with parameter uncertainty (6) and an LTL formula ϕ over a set of linear predicates (7), find a set $X_\phi \subseteq \mathcal{X}$ such that all trajectories of (6) originating in X_ϕ satisfy formula ϕ for all allowed parameters.

We propose a solution of Problem 1, involving the construction of equivalence classes, induced by the polytopal partition of the state space and linear predicates, and subsequent iterative refinement and model checking of the finite quotients. This approach is conservative, in the sense that obtaining a largest region (Definition 3) is not guaranteed.

IV. ITERATIVE CONSTRUCTION AND VERIFICATION OF SIMULATION QUOTIENTS

A characterization of bisimulation (Definition 2) can be given using the Pre operator from Equation (1) and leads to the iterative procedure for the construction of the coarsest bisimulation $\sim$ given in [6]. In general, the bisimulation algorithm does not terminate but despite that, the satisfaction of an LTL formula ϕ by T might be decided. Indeed, the equivalence relation produced at each step can be used to construct finite simulation quotients $T/_\sim$, which can then be model checked against an LTL formula (see Equation (4)). A similar idea was used in [8] for the universal fragment ACTL of CTL.

In our previous work [22], we combined the bisimulation procedure with model checking for an algorithm that relays on the computation of $Pre()$ and attempts to find the largest region $X_\phi \subseteq Q$ (Definition 3) that satisfies an LTL formula ϕ. The initial partition, given by propositional equivalence, was iteratively refined and the produced finite quotient transition systems were model checked against both ϕ and $\neg\phi$, partitioning the set of states Q into two subsets: X_ϕ (the set of all states satisfying the formula) and $X_{\neg\phi}$ (the set of all states satisfying the negation of the formula).

If $Pre()$ is not available but $Post()$ (Definition 2) is, the set of transitions in $T/_\sim$ can still be obtained:

$$(X, X') \in \rightarrow_\sim \text{ if and only if } Post(X) \cap X' \neq \emptyset \quad (8)$$

Although the construction of the quotient $T/_\sim$ can be performed in the absence of $Pre()$, refinement cannot be done as an implementation of the bisimulation procedure. Instead, a splitting strategy, which does not depend on the topology of the quotient can be used. The subregions, resulting from such refinement might not provide as much detail about the system as bisimulation refinement, since they don't consider the dynamics, but given enough iterations, regions X_ϕ and $X_{\neg\phi}$ might still be expanded.

A procedure for refinement and model checking of a quotient at each iteration in the absence of $Pre()$ is summarized as Algorithm 1. An in-house model checker, optimized for the particular case where each state of the system is considered as initial, is implemented [17]. Two Büchi automata (one for the formula and one for its negation) are constructed from the LTL formula (using the LTL2BA package [14]) before iterating through the algorithm. A product automaton of the quotient with each Büchi automaton is constructed at the beginning of each iteration. Then, within an iteration, only the initial states of the two product automata are changed, which is much faster than their initial construction.

If there are no satisfying runs (defined as trajectories visiting a final state infinitely often) for the negation of the formula originating in a particular state of the product automaton then the formula is satisfied from that state (similarly, lack of satisfying runs for the formula signifies the satisfaction of its negation).

Algorithm 1 will terminate either when a partition of the state space is obtained in X_ϕ and $X_{\neg\phi}$ or when a

Algorithm 1 Determine the largest X such that $T(X) \vDash \phi$ when $Pre()$ is not computable

Initialize $\sim$ with propositional equivalence;
Initialize $X_\phi := \emptyset$, $X_{\neg\phi} := \emptyset$;
Construct Büchi B_ϕ and $B_{\neg\phi}$;
while number of iterations is less than limit **do**
  Construct $T/_\sim$;
  Construct Product Automata $P_\phi = B_\phi \times T/_\sim$ and $P_{\neg\phi} = B_{\neg\phi} \times T/_\sim$;
  for every $X_t \in Q/_\sim, X_t \notin X_\phi, X_t \notin X_{\neg\phi}$ **do**
    adjust initial state in P_ϕ and $P_{\neg\phi}$ for X_t;
    if there does not exist a satisfying run in $P_{\neg\phi}$ **then**
      $X_\phi := X_\phi \cup X_t$;
    else if there does not exist a satisfying run in P_ϕ **then**
      $X_{\neg\phi} := X_{\neg\phi} \cup X_t$;
    end if
  end for
  if $X_\phi \cup X_{\neg\phi} = Q$ **then**
    return $X_\phi, X_{\neg\phi}$;
  end if
  for every $X \in Q/_\sim, X \notin X_\phi, X \notin X_{\neg\phi}$ **do**
    split X;
  end for
end while
return $X_\phi, X_{\neg\phi}$;

preset iteration limit is reached. Once an equivalence class is labeled during the execution, it is memorized and no longer considered for refinement or model checking. We do not need to refine a state that has been found to satisfy the formula or its negation, since all or no trajectories, respectively, originating at that state satisfy the formula. We do not need to model check that state any more, because additional refining anywhere else in the system would not change the satisfaction of a formula by the state. This optimization limits the explosion of states that have to be considered as refinement progresses and computation can be targeted to smaller regions where more accurate partitioning is needed. Additionally, since equivalence classes are refined in parallel, *i.e.*, all classes that need refinement are split once at every iteration, if the execution of Algorithm 1 is terminated because a preset number of iterations is exceeded, the refinement would not be concentrated in only one region.

Remark 1: Algorithm 1 can be applied directly, even when the $Post()$ operation is not exactly computable but an overapproximation, $\overline{Post}()$ can be obtained. In such cases, an overapproximation $\bar{T}/_\sim$ of the quotient is constructed that has the same states as $T/_\sim$ but may contain additional (spurious) transitions. Since language inclusion (Equation 3) is not violated, $\bar{T}/_\sim$ can still be used in Algorithm 1. The overall method, however, becomes more conservative.

V. FINITE QUOTIENTS OF PWA SYSTEMS

To formally define the satisfaction of a formula ϕ over Π by system (6), we embed it into a transition system:

Definition 4: An embedding transition system for (6) and the set of predicates Π can be defined as $T_{emb} = (Q_{emb}, \rightarrow_{emb}, \Pi_{emb}, \vDash_{emb})$, where

- $Q_{emb} = \bigcup_{l \in L} \mathcal{X}_l$,
- $(x, x') \in \rightarrow_{emb}$ for $x \in \mathcal{X}_l$ if and only if there exist $l \in L$, $A_l \in \mathcal{P}_l^A$ and $b_l \in \mathcal{P}_l^b$ such that $x' = A_l x + b_l$,
- $\Pi_{emb} = L \bigcup \Pi$,
- $\vDash_{emb}$ is defined as follows: if $\pi = l \in L$, then $x \vDash_{emb} \pi$ if and only if $x \in \mathcal{X}_l$; if $\pi = \pi_i \in \Pi$, then $x \vDash_{emb} \pi$ if and only if $c_i^T x + d_i < 0$,

Given a subset $X \subseteq Q_{emb}$, we say that all trajectories of system (6) originating in X satisfy formula ϕ if and only if $T_{emb}(X)$ satisfies ϕ, which was defined in Section II-B.

The embedding transition system T_{emb} has infinitely many states and cannot be model checked directly. However, starting from propositional equivalence, finite quotients can be iteratively constructed. If $Pre()$ is computable, then the algorithms described in [22] can provide a solution to Problem 1. Alternatively, if $Pre()$ is not computable but $Post()$ can be computed exactly, Algorithm 1 can be used instead. Finally, if $Post()$ cannot be computed exactly but an overapproximation $\overline{Post}()$ can be obtained, then Problem 1 can be solved conservatively by constructing the overapproximation $\bar{T}/_\sim$ within Algorithm 1 (Remark 1).

The computation of the (initial) propositional equivalence relation $\sim$ amounts to checking the non-emptiness of the open polytopes given by the intersection of each $\mathcal{X}_l$ with all subsets of Π (recall that $\mathcal{X}_l$ are pairwise disjoint). The equivalence classes formed by all such nonempty sets will be the states of the first quotient $Q_{emb}/_\sim$ and all operations with those states are polyhedral operations.

With this representation of equivalence classes in the quotient transition system, and given a region $\mathcal{X} \subseteq \mathcal{X}_l, l \in L$ from Equation (2) and Definition 4, we have

$$Post(\mathcal{X}) = \{x' \in \mathbb{R}^N \mid x' = Ax + b, \text{ for all } A \in \mathcal{P}_l^A, b \in \mathcal{P}_l^b, x \in \mathcal{X}\} \quad (9)$$

Under parameter uncertainty, the $Post()$ of a convex region is not necessarily convex. In Proposition 1, we show that if $\mathcal{X}$ is a polytope included in $\mathcal{X}_l$ for some $l \in L$, the convex hull of the affine transformations of all vertices of $\mathcal{X}$, with all vertices of $\mathcal{P}_l^A$ and $\mathcal{P}_l^b$ as parameters, provides a polyhedral overapproximation of $Post(\mathcal{X})$.

Proposition 1: A polyhedral overapproximation of $Post(X)$ in Equation (9) can be computed as

$$\overline{Post}(\mathcal{X}) = Conv\{A_l^{v_m} x^{v_k} + b_l^{v_s}, \text{ for all } m \leq M, k \leq K, s \leq S\} \quad (10)$$

where $\mathcal{X} = Conv\{x^{v_1}, \ldots, x^{v_K}\}, \mathcal{X} \subseteq \mathcal{X}_l$, $\mathcal{P}_l^A = Conv\{A_l^{v_1}, \ldots, A_l^{v_M}\}$ and $\mathcal{P}_l^b = Conv\{b_l^{v_1}, \ldots, b_l^{v_S}\}$

Proof: http://iasi.bu.edu/~yordanov/papers/ACC08.pdf ■

The computation of $\overline{Post}()$ described above returns a polyhedral set and, therefore, an overapproximation of the quotient transition system $\bar{T}/_\sim$ can be constructed (see Remark 1). All the remaining computation involved in the

execution of Algorithm 1 consists only of polyhedral set operations and model checking.

The regions X_ϕ and $X_{\neg\phi}$ returned by Algorithm 1 correspond to regions of initial conditions of the system with the property that that all runs originating at X_ϕ and no runs originating at $X_{\neg\phi}$ satisfy the formula no matter what parameters are selected at every step. If $X_\phi \bigcup X_{\neg\phi} = \mathcal{X}$ (here we abuse the notation by ignoring the facets of the polytopes), then X_ϕ is the largest region of initial conditions for the system satisfying the formula.

A software tool incorporating Algorithm 1 (with a quadtree splitting strategy) and bisimulation refinement (Algorithm 3 of [22] for the deterministic, fixed parameter case) was implemented in MATLAB and is freely downloadable at `http://iasi.bu.edu/~cbelta/software.htm`.

Remark 2: There are several simplifying assumptions that we make in the formulation and solution of Problem 1. First, we assume that the polytope $\mathcal{X}$ is an invariant for all trajectories of (6). 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 (7) 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. 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.

VI. ANALYSIS OF A TWO-GENE NETWORK

As in our previous work [22], we illustrate the proposed method by analyzing the genetic network shown in Fig. 1.

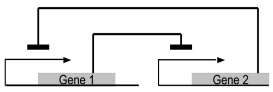

Fig. 1. A genetic switch consisting of two mutual repressors. High levels of one of the products shut down the expression of the other gene.

To analyze the system, we developed 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, at high repressor concentrations (above threshold 2) expression is only basal and the response between the two thresholds is graded. Since there are two repressors, two ramp functions are used and, therefore, the system has a total of nine rectangular invariants (denoted as $\mathcal{X}_1, \ldots, \mathcal{X}_9$ with $L = \{1, 2, \ldots, 9\}$ as the set of labels).

The initial model was developed with fixed parameters for each region (not shown due to space constraints but available at `http://iasi.bu.edu/~yordanov/papers/ACC08.pdf`) and hyper-rectangular parameter uncertainty was introduced by allowing each component of the parameters A_l and b_l for region $l \in L$ to vary in a range of size specified as a percentage of the fixed parameter value and centered around it (parameter components equal to 0 were also allowed to vary in a small range).

Under the fixed parameters, dynamics 3 and 7 have unique, asymptotically stable equilibria inside rectangles $\mathcal{X}_3$ and $\mathcal{X}_7$. 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 and exploring how those regions change when parameter uncertainty is introduced. For this, additional propositions are not necessary, *i.e.,* $\Pi = \emptyset$ and therefore, in Definition 4, $\Pi_{emb} = L$. By exploiting convexity properties of affine functions on polytopes, it can be easily proved that under the fixed parameters, $\mathcal{X}_3$ and $\mathcal{X}_7$ are invariants for dynamics 3 and 7, respectively (this is also obvious from the lack of outgoing transitions from those regions other than a self loop in Fig. 2 (A)). 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 $\phi_1 =$ "$\Diamond\Box\mathcal{X}_3$" and $\phi_2 =$ "$\Diamond\Box\mathcal{X}_7$" *i.e.* we want to find maximal sets of initial conditions, from which trajectories will eventually reach regions $\mathcal{X}_3$ or $\mathcal{X}_7$ and stay there forever.

The overapproximation first quotient $\bar{T}/_\sim$ (or exact $T/_\sim$ under fixed parameters) of the embedding T_{emb} (Definition 4) determined by propositional equivalence has nine states under all uncertainties and transitions shown in Fig. 2. As expected, more transitions are introduced in the finite quotient as parameters are allowed to vary in larger ranges (although some of the additional transitions might be due to the underlaying conservatism). We note that at 5% uncertainty we cannot guarantee that $\mathcal{X}_7$ is an invariant for trajectories of the system (Fig. 2 (C)) while at 10% uncertainty, there is no guarantee that $\mathcal{X}_3$ is an invariant either (Fig. 2 (D)).

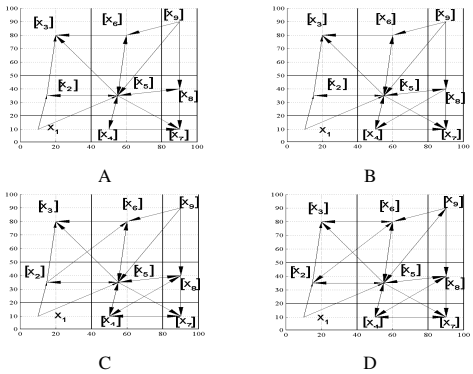

Fig. 2. A graphical representation of the first quotient transition system under fixed parameters (A), and an overapproximation of the first quotient for 1% (B), 5% (C), and 10% (D) uncertainty (square brackets around the state label represent self loops).

Because of the rectangular initial partition of the state space, 2^N-trees were used as an efficient splitting strategy. After 4 iterations, our method identifies only an attracting region for the equilibria at $\mathcal{X}_3$ for 5% uncertainty (Fig. 3 (A,B)). Biologically, this signifies that the stability of the two modes of the system changes as variation in the parameters is

introduced and, at 5% uncertainty we cannot guarantee that once gene 1 is activated it will remain activated (performing the analysis with different formulas might reveal other possible types of behavior).

Results from the computation with various levels of uncertainty are compared with the ones obtained using Algorithm 3 from [22] under fixed parameters (Fig. 3). As expected, increasing the level of uncertainty in the parameters decreases the size of the identified regions of state space (but a region identified at higher uncertainty is always a subset of the one identified at lower uncertainty). Because the refinement scheme used in [22] is different, less polytopes are partitioned per iteration of the algorithm, and therefore more iterations are required to achieve similar level of detail (as indicated by the total number of polytopes after performing the computation).

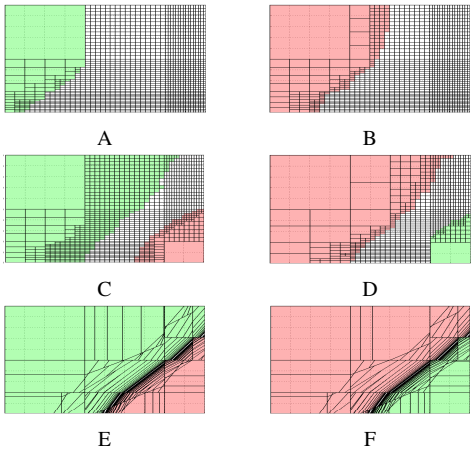

Fig. 3. Computation for ϕ_1 (A,C,E) and ϕ_2 (B,D,F) after 4 iterations under 5% (A,B) and 1% (C,D) uncertainty and 15 iterations under fixed parameters (E,F) (Algorithm 3 from [22]). Regions satisfying the formula are shown in green, while regions satisfying the negation are shown in red.

Even if computation is performed for many iterations, under parameter uncertainty it is possible that a subset of the state space is never included in the identified regions- a property resulting from nondeterminism introduced in the embedding transitions system. Even though complete partitioning of the state space might not be possible, performing more iterations of the procedures provides further refinement and, therefore, greater detail of the identified regions (initial iterations attempt to capture large regions, while subsequent ones expand X_ϕ and $X_{\neg\phi}$ less but provide greater resolution on their boundaries).

VII. CONCLUSION

We showed that finite simulation quotients of PWA systems can be efficiently constructed and model checked against an LTL formula to analyze properties of their trajectories, even under polyhedral parameter uncertainty. Additionally, we analyzed a PWA model of a genetic switch to determine the effect of parameter uncertainty on the regions of attraction for the two equilibria of the system. In the future we will focus on models of gene networks constructed from experimental data and identification of subsets of parameters, for which a property is satisfied (parameter synthesis).

REFERENCES

[1] R. Alur, T. A. Henzinger, G. Lafferriere, and G. J. Pappas, "Discrete abstractions of hybrid systems," *Proceedings of the IEEE*, vol. 88, pp. 971–984, 2000.

[2] M. Antoniotti, F. Park, A. Policriti, N. Ugel, and B. Mishra, "Foundations of a query and simulation system for the modeling of biochemical and biological processes," ser. Proceedings of the Pacific Symposium on Biocomputing, R. Altman, A. Dunker, L. Hunter, and T. Klein, Eds., 2003, pp. 116–127.

[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] G. Batt, C. Belta, and R. Weiss, "Model checking genetic regulatory networks with parameter uncertainty," in *HSCC*, ser. Lecture Notes in Computer Science, vol. 4416, 2007, pp. 61–75.

[5] G. Bernot, J.-P. Comet, A. Richard, and J. Guespin, "Application of formal methods to biological regulatory networks: Extending Thomas' asynchronous logical approach with temporal logic," *Journal of Theoretical Biology*, vol. 229, no. 3, pp. 339–347, 2004.

[6] A. Bouajjani, J.-C. Fernandez, and N. Halbwachs, "Minimal model generation," in *CAV 90: Computer-Aided Verification*, ser. Lecture Notes in Computer Science, R. P. Kurshan and E. M. Clarke, Eds., vol. 531. Springer, 1990, pp. 197–203.

[7] N. Chabrier-Rivier, M. Chiaverini, V. Danos, F. Fages, and V. Schächter, "Modeling and querying biomolecular interaction networks," *Theoretical Comp. Science*, vol. 325, no. 1, pp. 25–44, 2004.

[8] A. Chutinan and B. H. Krogh, "Verification of infinite-state dynamic systems using approximate quotient transition systems," *IEEE Transactions on automatic control*, vol. 46, no. 9, pp. 1401–1410, 2001.

[9] 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.

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

[11] J. Davoren, V. Coulthard, N. Markey, and T. Moor, "Non-deterministic temporal logics for general flow systems," in *HSCC: 7th International Workshop*, 2004, pp. 280–295.

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

[13] T. Gardner, Personal communication, Boston University, 2006.

[14] P. Gastin and D. Oddoux, "LTL with past and two-way very-weak alternating automata," in *Proceedings of MFCS'03*, B. Rovan and P. Vojtáš, Eds., vol. 2747. Springer, 2003, pp. 439–448.

[15] 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.

[16] 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," *Lecture Notes in Computer Science*, vol. 3414/2005, pp. 354–369, 1993.

[17] M. Kloetzer and C. Belta, "A fully automated framework for control of linear systems from LTL specifications," in *HSCC: 9th International Workshop*, ser. Lecture Notes in Computer Science, J. Hespanha and A. Tiwari, Eds. Springer, 2006, vol. 3927, pp. 333 – 347.

[18] M. Kvasnica, P. Grieder, and M. Baotić, "Multi-Parametric Toolbox (MPT)."

[19] S. G. Loizou and K. J. Kyriakopoulos, "Automatic synthesis of multiagent motion tasks based on LTL specifications," in *43rd IEEE Conference on Decision and Control*, 2004.

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

[21] P. Tabuada and G. Pappas, "Model checking LTL over controllable linear systems is decidable," ser. Lecture Notes in Computer Science, O. Maler and A. Pnueli, Eds. Springer, 2003, vol. 2623.

[22] B. Yordanov, G. Batt, , and C. Belta, "Model checking discrete-time piecewise affine systems: application to gene networks," in *European Control Conference (ECC)*, 2007.