Model Checking Genetic Regulatory Networks with Parameter Uncertainty

Grégory Batt[1], Calin Belta[1], and Ron Weiss[2]
[1] Center for Information and Systems Engineering and Center for BioDynamics, Boston University, Brookline, MA, USA
[2] Department of Electrical Engineering and Department of Molecular Biology, Princeton University, Princeton, NJ, USA
batt@bu.edu, cbelta@bu.edu, rweiss@princeton.edu

Abstract. The lack of precise numerical information for the values of biological parameters severely limits the development and analysis of models of genetic regulatory networks. To deal with this problem, we propose a method for the analysis of genetic regulatory networks with parameter uncertainty. We consider models based on piecewise-multiaffine differential equations, dynamical properties expressed in temporal logic, and intervals for the values of uncertain parameters. The problem is then either to guarantee that the system satisfies the expected properties for every possible parameter value - the corresponding parameter set is then called valid - or to find valid subsets of a given parameter set. The proposed method uses discrete abstractions and model checking, and allows for efficient search of the parameter space. This approach has been implemented in a tool for robust verification of gene networks (RoVerGeNe) and applied to the tuning of a synthetic network build in *E. coli*.

1 Introduction

Numerous cellular processes are controlled at the molecular level by networks of interactions between genes, proteins and small molecules, called *genetic regulatory networks*. Understanding how the cellular behavior emerges from these networks of interactions is a central problem in systems and synthetic biology [1, 2]. Arguably, the most widely-used modeling frameworks for the analysis of the dynamics of these networks are based on differential equations [3]. With few exceptions [4], it is generally assumed that the numerical values of state variables and model parameters are precisely known. However, given the current limitations of experimental measurement techniques, and the fact that parameter values themselves vary with the ever-fluctuating extra- and intracellular environments, the results obtained by these techniques may be of limited validity.

In this work, we present a method for the analysis of genetic regulatory networks with *parameter uncertainty*. We consider gene network models based on piecewise-multiaffine (PMA) differential equations, dynamical properties expressed in temporal logic, and intervals for the values of uncertain parameters. The problem is then either to *guarantee* that the system satisfies the expected properties for *every* possible parameter value - the corresponding parameter set is then called *valid* - or to *find* valid subsets of a given parameter set.

In the proposed approach, we use a partition of the *state space* induced by the piecewise nature of the models and specific properties of multiaffine functions [5] to define an equivalence relation on parameters. *Discrete abstractions* [6] are used to transpose the problem defined on (infinite) continuous state and parameter spaces into a problem defined on (finite) discrete spaces. Algorithmic analysis by *model-checking* [7] is then possible. Conservative approximations are used that guarantee that the parameter sets returned by the procedure are valid. However, not all valid parameter sets are guaranteed to be found. This approach has been implemented in a tool for Robust Verification of Gene Networks (RoVerGeNe) and applied to the analysis of the tuning of a synthetic gene network, build in the bacterium *E. coli*. This case study demonstrate the practical applicability and biological relevance of the proposed approach.

This paper is organized as follows. Section 2 introduces preliminary notions. PMA models are presented in Section 3, and the proposed approach is detailed in Section 4. The application to the tuning of a network is presented in Section 5. The final section discusses the results in the context of related work.

2 Preliminaries

All the notions and notations presented here are described at length in [8]. We consider Kripke structures $T = (S, \rightarrow, \Pi, \models)$ defined over sets of atomic propositions Π, and simply called *transition systems* [7]. S is a (finite or infinite) set of states, $\rightarrow \subseteq S \times S$, a total transition relation, and $\models \subseteq S \times \Pi$, a satisfaction relation. An *execution* of T is an infinite sequence $e = (s_0, s_1, s_2, \ldots)$ such that for every $i \geq 0$, $s_i \in S$ and $(s_i, s_{i+1}) \in \rightarrow$. We use the syntax and the semantics of *Linear Temporal Logic* (LTL) formulas defined over executions of Kripke structures given in [7]. We refer to [6] for the usual notions of *simulation* between transition systems and of *quotient transition systems*. T_1 simulates T_2 is denoted $T_2 \preceq T_1$, and we recall that simulation relations weakly preserve LTL [7].

Polytopes are bounded intersections of finitely-many open or closed half-spaces. A polytope P is *hyperrectangular* if $P = P_1 \times \ldots \times P_n$ with $P_i = \{x_i \in \mathbb{R} \mid x = (x_1, \ldots, x_n) \in P\}$, $i \in \{1, \ldots, n\}$. The definitions of the closure, vertices, faces and facets of a polytope are recalled in [8]. $\overline{P}$ and $\mathcal{V}_P$ denote respectively the closure and the set of vertices of a polytope P. A function $f : \mathbb{R}^n \rightarrow \mathbb{R}^m$ is multiaffine if it is a polynomial with the property that the degree of f in any of its variable is at most 1. Theorem 1 is proven in [5].

Theorem 1 *Let $f : \mathbb{R}^n \rightarrow \mathbb{R}^m$ be a multiaffine function and P be a hyperrectangular polytope in $\mathbb{R}^n$, $n, m \in \mathbb{N}$. Then, for every $x \in P$, $f(x)$ is a convex combination of the values of f at the vertices of P.*

3 Uncertain PMA models of genetic regulatory networks

3.1 PMA systems and specifications

In this section, we present a formalism for modeling gene networks. The notations and terminology are adapted from [9]. We consider a gene network consisting of

n genes. The state of the network is given by the vector $x = (x_1, \ldots, x_n)$, where x_i is the concentration of the protein encoded by gene i. The state space $\mathcal{X}$ is a hyperrectangular subset of $\mathbb{R}^n$: $\mathcal{X} = \prod_{i=1}^{n}[0, max_{x_i}]$, where max_{x_i} denotes a maximal concentration of the protein encoded by gene i. Some parameters may be *uncertain*: $p = (p_1, \ldots, p_m)$ is the vector of uncertain parameters, with values in the parameter space $\mathcal{P} = \prod_{j=1}^{m}[min_{p_j}, max_{p_j}]$, where min_{p_j} and max_{p_j} denote a minimal and a maximal value for p_j.

The dynamics of the network is given by the differential equations

$$\dot{x}_i = f_i(x, p) = \sum\nolimits_{j \in P_i} \kappa_i^j \, r_i^j(x) - \sum\nolimits_{j \in D_i} \gamma_i^j \, r_i^j(x) \, x_i, \quad i \in \{1, \ldots, n\}, \qquad (1)$$

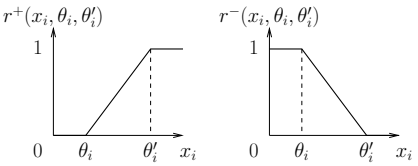
where P_i and D_i are sets of indices, $\kappa_i^j > 0$ and $\gamma_i^j > 0$ are (*possibly uncertain*) *production* and *degradation rate parameters*, and $r_i^j : \mathcal{X} \to [0, 1]$ are continuous, PMA functions, called *regulation functions* (see [8] for their precise syntax). PMA functions arise from products of ramp functions r^+ and r^- (Figure 1(a)) used for representing complex gene regulations or protein degradations (Figure 5(b), Eq. 4). With the additional assumption that r_i^j does not depend on x_i for $j \in D_i$,[3] it holds that $f = (f_1, \ldots, f_n) : \mathcal{X} \times \mathcal{P} \to \mathbb{R}^n$ is a (non-smooth) *continuous* function of x and p, a *piecewise-multiaffine* function of x and a *piecewise-affine* function of p. Note that production and degradation rate parameters may be uncertain, but regulation functions (with their threshold parameters) should be known. Finally, Equation (1) is easily extended to account for constant inputs u by considering u as a new variable satisfying $\dot{u} = 0$.

A number of dynamical properties of gene networks can be specified in temporal logic by LTL formulas over atomic propositions of type $x_i < \lambda$ or $x_i > \lambda$, where $\lambda \in \mathbb{R}_{\geq 0}$ is a constant. We denote by Π the set of all such atomic propositions. A *PMA system* Σ is then defined by a piecewise-multiaffine function f defined as above and a set of atomic propositions Π: $\Sigma = (f, \Pi)$.

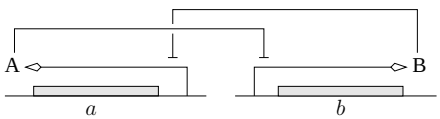
Consider the cross-inhibition network represented in Figure 1(b). This system can be represented by the PMA differential equations given in Figure 1(c). For example, the first equation states that protein A synthesis is inhibited by protein B (r^- function) and that its degradation is not regulated. Parameter values are given in Figure 1(d). Synthesis parameters are unknown: $(\kappa_a, \kappa_b) \in \mathcal{P} = [0, 40] \times [0, 20]$. For illustrating our purpose, we also consider $p_1 \in \mathcal{P}$ with $p_1 = (36, 17)$. This network is known to be *bistable*: it has two stable equilibrium states, corresponding to protein A and B concentrations being respectively high and low, or low and high. This property can be expressed in LTL by the property ϕ_1 (Figure 1(e)). For example, the first part of the property expresses that if the concentrations of protein A and B are respectively low ($x_a < \theta_a^1$) and high ($x_b > \theta_b^2$), then the system will always (G) remain in such a state. We refer the reader to [10] for a discussion of the use of invariants to express stability in biology.

PMA models of gene networks were proposed in [11] (see [12] for a related, piecewise-continuous formalism). The models considered here are also related to

[3] This assumption requires that a protein does not regulate its own degradation. In practice, this assumption is generally satisfied.



(a)



(b)

$$\dot{x}_a = \kappa_a r^-(x_b, \theta_b^1, \theta_b^2) - \gamma_a\, x_a$$
$$\dot{x}_b = \kappa_b r^-(x_a, \theta_a^1, \theta_a^2) - \gamma_b\, x_b$$

(c)

$$\phi_1 = \begin{array}{l}(x_a < \theta_a^1 \wedge x_b > \theta_b^2 \rightarrow \mathrm{G}\ (x_a < \theta_a^1 \wedge x_b > \theta_b^2)) \\ \wedge (x_b < \theta_b^1 \wedge x_a > \theta_a^2 \rightarrow \mathrm{G}\ (x_b < \theta_b^1 \wedge x_a > \theta_a^2))\end{array}$$

(e)

	κ_i	γ_i	θ_i^1	θ_i^2
x_a	$[0, 40]$	2	8	12
x_b	$[0, 20]$	1	8	12

(d)

Fig. 1. (a) Ramp functions r^+ and r^-. θ_i and θ'_i are threshold parameters. (b) Gene network comprising two genes, a and b, coding for two repressor proteins, A and B. Each protein represses the expression of the other gene, forming a cross-inhibition network. (c) PMA model of the network in (b). Because of its simplicity, this model is actually piecewise-affine. (d) Known and uncertain parameter values. (e) Bistability property expressed in LTL.

the piecewise-affine (PA) models proposed in [13] (see also [9]). However, contrary to the step functions used in PA models, ramp functions capture the graded response of gene expression to continuous changes in effector concentrations.

3.2 Embedding transition systems

The specific form of the PMA functions f suggests a division of the state space $\mathcal{X}$ into hyperrectangular regions (Figure 2(a) for our example). For every $i \in \{1, \ldots, n\}$, let $\Lambda_i = \{\lambda_i^j\}_{j \in \{1,\ldots,l_i\}}$ be the ordered set of all threshold constants in f, and of all atomic proposition constants in Π, associated with gene i, together with 0 and max_{x_i}. The cardinality of Λ_i is l_i. Then, we define $\mathcal{R}$ as the following set of n-dimensional hyperrectangular polytopes $R \subseteq \mathcal{X}$, simply called *rectangles*:

$$\mathcal{R} = \{R_c \mid c = (c_1, \ldots, c_n) \text{ and } \forall i \in \{1, \ldots, n\} : c_i \in \{1, \ldots, l_i - 1\}\},$$

where

$$R_c = \{x \in \mathcal{X} \mid \forall i \in \{1, \ldots, n\} : \lambda_i^{c_i} < x_i < \lambda_i^{c_i+1}\}.$$

c is the *coordinate* of the rectangle R_c. The union of all rectangles in $\mathcal{X}$ is denoted by $\mathcal{X}_\mathcal{R}$: $\mathcal{X}_\mathcal{R} = \cup_{R \in \mathcal{R}} R$. Note that $\mathcal{X}_\mathcal{R} \neq \mathcal{X}$. Notably, threshold hyperplanes are not included in $\mathcal{X}_\mathcal{R}$. Two rectangles R and R', are said *adjacent*, denoted $R \leftrightarrows R'$, if they share a facet. $coord : \mathcal{R} \rightarrow \prod_{i=1}^{n} \{1, \ldots, l_i - 1\}$ maps every rectangle $R \in \mathcal{R}$ to its coordinate, and $rect : \mathcal{X}_\mathcal{R} \rightarrow \mathcal{R}$ maps every point x in $\mathcal{X}_\mathcal{R}$ to the rectangle R such that $x \in R$. For the cross-inhibition network, the set $\mathcal{R} = \{R_{11}, \ldots, R_{33}\}$ of all rectangles is represented in Figure 2(b). R_{11} and R_{21} are adjacent, whereas R_{11} and R_{22} are not.

Formally, the semantics of a PMA system Σ is defined by means of an embedding transition system.

Definition 1 (Embedding transition system) *Let $p \in \mathcal{P}$. The embedding transition system associated with the PMA system $\Sigma = (f, \Pi)$ is $T_{\mathcal{X}}(p) = (\mathcal{X}_{\mathcal{R}}, \rightarrow_{\mathcal{X},p}, \Pi, \models_{\mathcal{X}})$ defined such that:*

- *$\rightarrow_{\mathcal{X},p} \subseteq \mathcal{X}_{\mathcal{R}} \times \mathcal{X}_{\mathcal{R}}$ is the transition relation defined by $(x, x') \in \rightarrow_{\mathcal{X},p}$ iff there exists a solution ξ of (1) and $\tau \in \mathbb{R}_{>0}$ such that $\xi(0) = x$, $\xi(\tau) = x'$, $\forall t \in [0, \tau]$, $\xi(t) \in \overline{rect}(x) \cup \overline{rect}(x')$, and either $rect(x) = rect(x')$ or $rect(x) \fallingdotseq rect(x')$,*
- *$\models_{\mathcal{X}} \subseteq \mathcal{X}_{\mathcal{R}} \times \Pi$ is the satisfaction relation defined by $(x, \pi) \in \models_{\mathcal{X}}$ iff $x = (x_1, \ldots, x_n)$ satisfies the proposition π (of type $x_i < \lambda$ or $x_i > \lambda$) with the usual semantics.*

Remark. Not all solution trajectories of (1) are represented by executions of the embedding. First, due to our restricted notion of adjacency ($\fallingdotseq$), solution trajectories of (1) that go from a rectangle to another by passing through a face of low ($<n$–1) dimension are not represented in the embedding. Second, the dynamics of the system in $\mathcal{X} \setminus \mathcal{X}_{\mathcal{R}}$ (including the threshold hyperplanes) is not described by the embedding. However, since the vector field is continuous everywhere, trajectories originating in full-dimensional rectangles can not "disappear" in a facet by sliding along the supporting hyperplane. Consequently, the embedding describes *almost* all solution trajectories of (1), which is satisfying for all practical purposes.

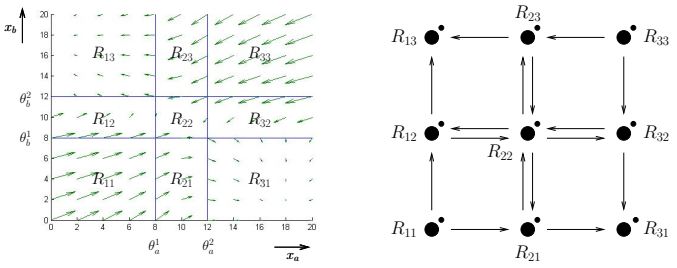


Fig. 2. (a) Continuous dynamics in the state space of the cross-inhibition network for parameter $p_1 = (\kappa_a, \kappa_b) = (36, 17)$. (b) Discrete abstraction of the dynamics in (a). Dots denote self transitions.

A PMA system Σ satisfies an LTL formula ϕ for a given parameter $p \in \mathcal{P}$ if $T_{\mathcal{X}}(p) \models \phi$, that is, if every execution of $T_{\mathcal{X}}(p)$ satisfies ϕ. Then, valid parameter sets are defined as follows.

Definition 2 *Let Σ be a PMA system and ϕ an LTL formula. A parameter set $P \subseteq \mathcal{P}$ is* valid *for ϕ iff Σ satisfies ϕ for almost all $p \in P$.*

Again, the use of *almost all* is motivated by the fact that this criteria is sufficient for all practical purposes. Finally, we consider the following problems.

Problem *Let Σ be a PMA system, $\mathcal{P}$ an hyperrectangular parameter space, and ϕ an LTL formula.*
1. Robustness analysis*: Check whether $\mathcal{P}$ is valid for ϕ.*
2. Synthesis*: Find a set $P \subseteq \mathcal{P}$ such that P is valid for ϕ.*

4 Analysis of PMA system with parameter uncertainty

4.1 Discrete abstraction

We use discrete abstractions [6] to obtain finite transition systems preserving dynamical properties of $T_{\mathcal{X}}(p)$ and amenable to algorithmic verification [7]. Let $\sim_{\mathcal{R}} \subseteq \mathcal{X}_{\mathcal{R}} \times \mathcal{X}_{\mathcal{R}}$ be the (proposition-preserving) equivalence relation defined by the surjective map *rect*: $x \sim_{\mathcal{R}} x'$ iff $rect(x) = rect(x')$. $\mathcal{R}$ is the set of equivalence classes. Then, the discrete abstraction of $T_{\mathcal{X}}(p)$ is the *quotient* of $T_{\mathcal{X}}(p)$ given the equivalence relation $\sim_{\mathcal{R}}$.

Definition 3 *Let $p \in \mathcal{P}$. The discrete abstraction of $T_{\mathcal{X}}(p)$ is $T_{\mathcal{R}}(p) = (\mathcal{R}, \rightarrow_{\mathcal{R},p}, \Pi, \models_{\mathcal{R}})$, the quotient of $T_{\mathcal{X}}(p)$ given the equivalence relation $\sim_{\mathcal{R}}$.*

For our example network, $T_{\mathcal{R}}(p_1)$ is represented in Figure 2(b). By definition of quotient transition systems, $T_{\mathcal{R}}(p)$ *simulates* $T_{\mathcal{X}}(p)$.

$$\text{For every } p \in \mathcal{P},\ T_{\mathcal{X}}(p) \preceq T_{\mathcal{R}}(p). \tag{2}$$

In words, the discrete transition system $T_{\mathcal{R}}(p)$ is a *conservative approximation* of the continuous dynamics of the PMA system described by $T_{\mathcal{X}}(p)$. Because simulation relations *weakly preserve* LTL, we have for any LTL formula ϕ: if $T_{\mathcal{R}}(p) \models \phi$ then $T_{\mathcal{X}}(p) \models \phi$. The converse does not necessarily hold.

By exploiting specific properties of multiaffine functions defined over hyperrectangular polytopes [5], we provide the following characterization.

Proposition 1 *Let $p \in \mathcal{P}$. $T_{\mathcal{R}}(p) = (\mathcal{R}, \rightarrow_{\mathcal{R},p}, \Pi, \models_{\mathcal{R}})$, where*

- *$\rightarrow_{\mathcal{R},p} \subseteq \mathcal{R} \times \mathcal{R}$ is such that $(R, R') \in \rightarrow_{\mathcal{R},p}$ iff $R = R'$, or $R \Rightarrow R'$ and there exists $v \in \mathcal{V}_R \cap \mathcal{V}_{R'}$ such that*

$$f_i(v,p)(c'_i - c_i) > 0,$$

  with $c = coord(R)$, $c' = coord(R')$ and $i \in \{1, \ldots, n\}$ such that $c_i \neq c'_i$.
- *$\models_{\mathcal{R}} \subseteq \mathcal{R} \times \Pi$ is such that $(R, \pi) \in \models_{\mathcal{R}}$ iff for every $x \in R$, $(x, \pi) \in \models_{\mathcal{X}}$.*

Proof. Let $R, R' \in \mathcal{R}$. By Definition 1 and 3, it is clear that if neither $R = R'$ nor $R \Rightarrow R'$, there can not exist a transition from R to R'. If $R = R'$, then since it exists a solution of (1) that remains in R on $[0, \tau]$ for some $\tau > 0$, there exists a (self) transition from R to R' (Definition 1 and 3). The last case is when $R \Rightarrow R'$. Then, let $c = coord(R)$, $c' = coord(R')$ and $i \in \{1, \ldots, n\}$ such that $c_i \neq c'_i$ and let F be the facet shared by R and R'. We assume without loss of generality that $c'_i - c_i = 1$, the other case ($= -1$) being symmetrical.
$\Rightarrow$ (by contradiction): Suppose that for every $v \in \mathcal{V}_R \cap \mathcal{V}_{R'} = \mathcal{V}_F$, $f_i(v,p) \leq 0$. Using Theorem 1, it holds that for every $x \in F$, $f_i(x,p) \leq 0$. Consequently, no solution can enter R' from R and $(R, R') \notin \rightarrow_{\mathcal{R},p}$.
$\Leftarrow$: Assume that there exists $v \in \mathcal{V}_F$ such that $f_i(v,p) > 0$. By continuity of f, there exists a ball $B_{v,\epsilon}$ of center v and radius ϵ such that $\forall x \in B_{v,\epsilon}$, $f_i(x,p) > 0$. In particular, there exist $x_f \in F$, $x_f \neq v$, such that $f_i(x_f, p) > 0$. Then, there exists a solution entering R' from R without leaving $\overline{R} \cup \overline{R'}$, and by Definition 1 and 3, $(R, R') \in \rightarrow_{\mathcal{R},p}$.

The characterization of $\models_{\mathcal{R}}$ follows immediately from the fact that the equivalence relation $\sim_{\mathcal{R}}$ preserves the atomic propositions in Π. □

Informally, Proposition 1 simply states that there is a transition between two adjacent rectangles if and only if there exists at least one common vertex at which the direction of the vector field ($f_i(v,p)$) is in agreement with the relative position of the two rectangles ($c'_i - c_i$). Similar rules have been proposed in [14]. Consider the two rectangles R_{11} and R_{21} in Figure 2(a). They share two vertices: $v_1 = (\theta_a^1, 0)$ and $v_2 = (\theta_a^1, \theta_b^1)$. From Proposition 1, there is a transition from R_{11} to R_{21}, because $f_a(v_1, p_1) > 0$, and there is no transition from R_{21} to R_{11}, because neither $f_a(v_1, p_1) < 0$ nor $f_a(v_2, p_1) < 0$ (check with Figure 2(b)).

For known parameters, Proposition 1 provides a means to compute the relation $\rightarrow_{\mathcal{R},p}$ by evaluating f at all the vertices. The computation of the set of states $\mathcal{R}$ and of the relation $\models_{\mathcal{R}}$ are trivial. So $T_{\mathcal{R}}(p)$ can be computed and one can use model checking for testing whether $T_{\mathcal{R}}(p) \models \phi$. If the abstract system $T_{\mathcal{R}}(p)$ satisfies ϕ, then so does the original system $T_{\mathcal{X}}(p)$ (Property (2)), and p is valid for ϕ. Conversely, if $T_{\mathcal{R}}(p)$ does not satisfy ϕ, no conclusion on the validity of p can be obtained. If some parameters are unknown, we will use Proposition 1 to define an equivalence relation on parameters.

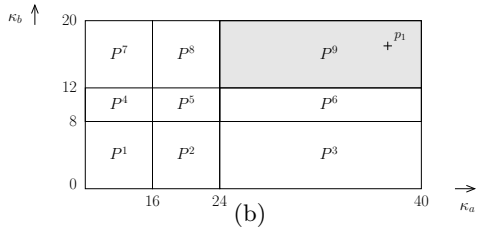
4.2 Parameter equivalence classes

Consider a vertex $v \in \mathcal{V}_R$, $R \in \mathcal{R}$. Because f is a piecewise-affine and continuous function of p, $f_i(v,p)$ is an affine expression in p: $f_i(v,p) = a^T p + b$, with $a \in \mathbb{R}^m$ and $b \in \mathbb{R}$. Let Ψ be the set of all such non-constant ($a \neq 0$) affine expressions:

$$\Psi = \{f_i(v,p) = a_{i,v}^T\, p + b_{i,v} \mid i \in \{1,\ldots,n\}, v \in \mathcal{V}_R, R \in \mathcal{R} \text{ and } a_{i,v} \neq 0\}.$$

After having removed repeated elements, we denote by n_Ψ the cardinality of Ψ and order the elements in Ψ: $\Psi = \{\psi_1, \ldots, \psi_{n_\Psi}\}$. For our example network, with uncertain parameters κ_a and κ_b, out of the 32 affine expressions only 4 different non-constant expressions exist: $n_\Psi = 4$ (Figure 3(a)).

$\Psi = \{\psi_1, \psi_2, \psi_3, \psi_4\}$, with
$\psi_1(p) = \kappa_a - 24$,
$\psi_2(p) = \kappa_b - 12$,
$\psi_3(p) = \kappa_b - 8$,
$\psi_4(p) = \kappa_a - 16$.
(a)



(b)

Fig. 3. (a) Set of affine expressions for the cross-inhibition network with unknown parameters κ_a and κ_b. (b) Parameter space in the dimensions of κ_a and κ_b. $p_1 = (36, 17)$ is represented. The shaded region is the set of all valid parameters for property ϕ_1.

The affine predicates $\psi_i(p) = 0$, $\psi_i \in \Psi$, divide the *parameter space* into polyhedral regions (Figure 3(b))[4]. These regions can be represented by a Boolean

[4] Note that, in general, the partition of the parameter space is not hyperrectangular

encoding. Let $\mathcal{B}^l$ be the set of Boolean numbers of length l: $\mathcal{B}^l = \{0,1\}^l$. We denote by ϵ the Boolean of length 0. Then, to every Boolean $b \in \mathcal{B}^l$, $l \in \{0,\ldots,n_\Psi\}$, we associate the parameter set P_b such that $P_\epsilon = \mathcal{P}$ and, if $b \neq \epsilon$,

$$P_b = \{p \in \mathcal{P} \mid \forall i \in \{1,\ldots,l\} : \psi_i(p) < 0, \text{ if } b_i = 0, \text{ and } \psi_i(p) > 0, \text{ if } b_i = 1\}.$$

The sets P_b are subsets of $\mathcal{P}$ obtained by adding constraints of type $\psi_i(p) < 0$ or $\psi_i(p) > 0$, with $\psi_i \in \Psi$. If b is a prefix of b', then $P_{b'} \subseteq P_b$. The hierarchy between the sets P_b induced by the set-inclusion partial-order is represented in Figure 4 for the cross-inhibition network (see [15, 16] for similar ideas in the context of predicate abstraction).

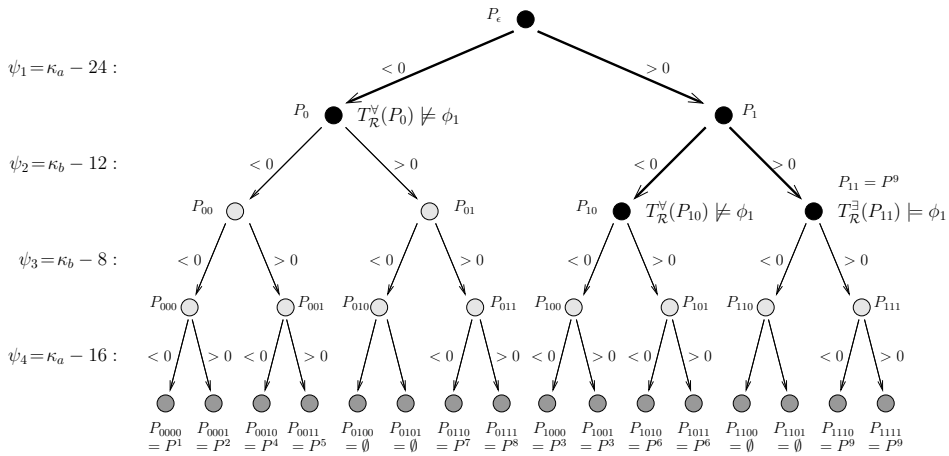


Fig. 4. Hierarchy between the parameter sets P_b, represented as a binary tree. Arrows indicate set inclusion: $P \rightarrow P'$ means $P' \subseteq P$. Leaves (dark gray) correspond to parameter equivalence classes. $P^1,\ldots,P^9$ refer to regions in Figure 3. The fragment of the tree actually computed during hierarchical parameter space exploration for the analysis of property ϕ_1 is emphasized. Model checking results used for backtracking are shown at the nodes where the recursive search stops.

We say that two parameters p and p' are *equivalent* if their associated discrete transition systems $T_\mathcal{R}(p)$ and $T_\mathcal{R}(p')$ are isomorphic. A similar definition is used in [17, 9]. Naturally, a PMA system satisfies the same LTL properties for two equivalent parameters.

Definition 4 *Let $\sim_\mathcal{P} \subseteq \mathcal{P} \times \mathcal{P}$ be the equivalence relation defined by $p \sim_\mathcal{P} p'$ iff $T_\mathcal{R}(p) = T_\mathcal{R}(p')$.*

Proposition 2 *Let $b_\Psi \in \mathcal{B}^{n_\Psi}$. For every $p, p' \in P_{b_\Psi}$, $p \sim_\mathcal{P} p'$.*

Proof. Let $b_\Psi \in \mathcal{B}^{n_\Psi}$ and $p, p' \in P_{b_\Psi}$. Then, $\forall i \in \{1,\ldots,n\}$, $R \in \mathcal{R}$ and $v \in \mathcal{V}_R$, $f_i(v,p) \# 0$ iff $f_i(v,p') \# 0$, with $\# \in \{<,>\}$. So, by Proposition 1, $T_\mathcal{R}(p) = T_\mathcal{R}(p')$ and $p \sim_\mathcal{P} p'$.

The above proposition states that the set of all predicates $\psi_i(p) = 0$, $\psi_i \in \Psi$, divide the parameter space in equivalence classes. Consequently, with $b_\Psi \in \mathcal{B}^{n_\Psi}$, if for some $p \in P_{b_\Psi}$, $T_\mathcal{R}(p) \models \phi$, then using Propositions 2 and Property (2), it holds that for all $p \in P_{b_\Psi}$, $T_\mathcal{X}(p) \models \phi$: P_{b_Ψ} is a valid parameter set. Since we can compute $T_\mathcal{R}(p)$ for any given p (Proposition 1), solutions to Problem 1 and 2 can be obtained by testing for every equivalence class $P_{b_\Psi} \subseteq \mathcal{P}$ whether $T_\mathcal{R}(p) \models \phi$ for some (randomly chosen) $p \in P_{b_\Psi}$. Note however that if $T_\mathcal{R}(p) \not\models \phi$, no conclusion can be obtained on P_{b_Ψ}. On our example network, only two equivalence classes, P_{1110} and P_{1111}, both corresponding to P^9, are found to be valid for the bistability property ϕ_1 (Figure 4 and 3). However, this naive approach is impractical since the number of equivalence classes (*i.e.* the leaves of the tree in Figure 4) increases exponentially with the number of affine predicates, the latter increasing exponentially with the number of variables and uncertain parameters. A more efficient approach is proposed in the next section.

4.3 Hierarchical parameter space exploration

Our goal is to describe the behavior of the network for *sets* of parameters $P \subseteq \mathcal{P}$. To do so, we introduce two transition systems, $T^\exists_\mathcal{R}(P)$ and $T^\forall_\mathcal{R}(P)$.

Definition 5 *Let* $P \subseteq \mathcal{P}$. *Then* $T^\exists_\mathcal{R}(P) = (\mathcal{R}, \rightarrow^\exists_{\mathcal{R},P}, \Pi, \models_\mathcal{R})$ *and* $T^\forall_\mathcal{R}(P) = (\mathcal{R}, \rightarrow^\forall_{\mathcal{R},P}, \Pi, \models_\mathcal{R})$, *where*
– $(R, R') \in \rightarrow^\exists_{\mathcal{R},P}$ *iff* $\exists p \in P$ *such that* $(R, R') \in \rightarrow_{\mathcal{R},p}$ *in* $T_\mathcal{R}(p)$, *and*
– $(R, R') \in \rightarrow^\forall_{\mathcal{R},P}$ *iff* $\forall p \in P$, $(R, R') \in \rightarrow_{\mathcal{R},p}$ *in* $T_\mathcal{R}(p)$.

In words, $T^\exists_\mathcal{R}(P)$ contains all the transitions present in at least one transition system $T_\mathcal{R}(p)$ and $T^\forall_\mathcal{R}(P)$ contains only the transitions present in all the transition systems $T_\mathcal{R}(p)$. For every $p \in P$, $T^\exists_\mathcal{R}(P)$ simulates $T_\mathcal{R}(p)$, which simulates $T^\forall_\mathcal{R}(P)$. This follows immediately from the definition of simulation between transition systems, using the fact that $\rightarrow^\forall_{\mathcal{R},P} \subseteq \rightarrow_{\mathcal{R},p} \subseteq \rightarrow^\exists_{\mathcal{R},P}$. Informally, $T^\exists_\mathcal{R}(P)$ and $T^\forall_\mathcal{R}(P)$ can be respectively considered as over- and under-approximations of the possible behaviors of $T_\mathcal{R}(p)$, when p varies.

Proposition 3 *For every* $p \in P$, $T^\forall_\mathcal{R}(P) \preceq T_\mathcal{R}(p) \preceq T^\exists_\mathcal{R}(P)$.

Using Proposition 3 and Property (2), it holds that for *any* $P \in \mathcal{P}$, if $T^\exists_\mathcal{R}(P) \models \phi$ then $\forall p \in P$, $T_\mathcal{X}(p) \models \phi$: P is a valid parameter set. Alternatively, using Proposition 3, it also holds that if $T^\forall_\mathcal{R}(P) \not\models \phi$, then $\forall p \in P$, $T_\mathcal{R}(p) \not\models \phi$: no valid parameter can be found in P using our approach, either because P contains no valid parameter, or because the discrete abstraction is overly conservative. Otherwise ($T^\exists_\mathcal{R}(P) \not\models \phi$ and $T^\forall_\mathcal{R}(P) \models \phi$), it is worth inspecting subsets of P, that may contain valid parameter sets. Accordingly, we propose an algorithm, detailed in [8], that explores $\mathcal{P}$ in a hierarchical manner by considering parameter sets P_b associated with Booleans of increasing length, starting from P_ϵ. This amounts to explore recursively the tree represented in Figure 4 for our example, using Proposition 3 as explained above to stop the search as soon as possible

(either because $T^{\exists}_{\mathcal{R}}(P_b) \models \phi$, or because $T^{\forall}_{\mathcal{R}}(P_b) \not\models \phi$). For the leaves (*i.e.* the equivalence classes), $T^{\exists}_{\mathcal{R}}(P_b) = T^{\forall}_{\mathcal{R}}(P_b)$ and the search necessarily terminates. Note that although $T^{\forall}_{\mathcal{R}}(P)$ does *not* provide information on the original system $T_{\mathcal{X}}(p)$ (no relation exist between $T^{\forall}_{\mathcal{R}}(P)$ and $T_{\mathcal{X}}(p)$), it makes it possible to identify large regions of the parameter space in which no valid parameter set can be found. Consequently, it plays a key role when exploring large parameter spaces where only small regions are valid sets. The fragment of the tree actually computed for the analysis of property ϕ_1 is represented in Figure 4. The same result is obtained as previously (P^9 is a valid parameter set), but in much fewer tests.

We have not yet explained how $T^{\forall}_{\mathcal{R}}(P)$ and $T^{\exists}_{\mathcal{R}}(P)$ can be computed.

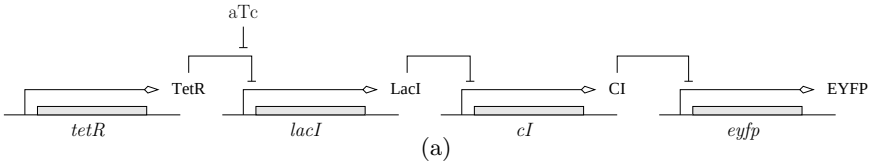
Proposition 4 (Computation of $T^{\exists}_{\mathcal{R}}(P)$ and $T^{\forall}_{\mathcal{R}}(P)$) *Let $P \subseteq \mathcal{P}$.*
– $(R, R') \in \rightarrow^{\exists}_{\mathcal{R},P}$ *iff either* $R = R'$, *or* $R \nleftrightarrow R'$ *and* $P \cap g(R, R') \neq \emptyset$,
– $(R, R') \in \rightarrow^{\forall}_{\mathcal{R},P}$ *iff either* $R = R'$, *or* $R \nleftrightarrow R'$ *and* $P \subseteq g(R, R')$,
where $g(R, R') = \{p \in \mathcal{P} \mid \exists v \in \mathcal{V}_R \cap \mathcal{V}_{R'}$ *such that* $f_i(v, p)(c'_i - c_i) > 0\}$, *with* $c = coord(R)$, $c' = coord(R')$ *and* $i \in \{1, \ldots, n\}$ *such that* $c_i \neq c'_i$.

Proof. Let $P \subseteq \mathcal{P}$ and $R, R' \in \mathcal{R}$ be such that $R \nleftrightarrow R'$ (the other cases being trivial). From Proposition 1, it is easy to see that $g(R, R')$ is the set of parameters $p \in \mathcal{P}$ for which there is a transition from R to R' in $T_{\mathcal{R}}(p)$. Then, the result follows from the definition of the transition relations $\rightarrow^{\exists}_{\mathcal{R},P}$ and $\rightarrow^{\forall}_{\mathcal{R},P}$ (Definition 5).

Given that $f_i(v, p)$ is an affine expression in p, the sets $g(R, R')$ correspond to unions of polytopes in $\mathcal{P}$. Consequently, for polyhedral sets P, the computation of the transition systems $T^{\exists}_{\mathcal{R}}(P)$ and $T^{\forall}_{\mathcal{R}}(P)$ using Proposition 4 simply amounts to compute intersections and inclusions of unions of polytopes, which are standard polyhedral operations efficiently implemented in toolboxes. This method has been implemented in a freely-available tool for Robust Verification of Gene Networks (RoVerGeNe, see `http://iasi.bu.edu/~batt/rovergene/rovergene.htm`). It is written in Matlab on top of several other tools (MPT, MatlabBGL, NuSMV). Because the efficiency of the computations may significantly depend on the order in which the affine predicates $\psi_i(p) = 0$, $\psi_i \in \Psi$, are considered during the search, we implemented a simple heuristic that orders first the predicates splitting the parameter space the more evenly (*i.e.* yielding two polytopes of similar volumes). Additionally, RoVerGeNe supports an extension of the method presented here, dealing with problems specifically encountered when verifying liveness properties, and described in [8, 18].

5 Tuning of a transcriptional cascade

The method presented in the previous section is applied to the analysis of the steady-state input/output (I/O) behavior of a synthetic transcriptional cascade build and analyzed in [19] (Figure 5(a)). We have developed a PMA model of this system, represented in Figure 5(b). Parameter values were estimated based on experimental data available in [19].



(a)

$$\dot{x}_{tetR} = \kappa_{tetR} - \gamma_{tetR}\, x_{tetR}, \tag{3}$$

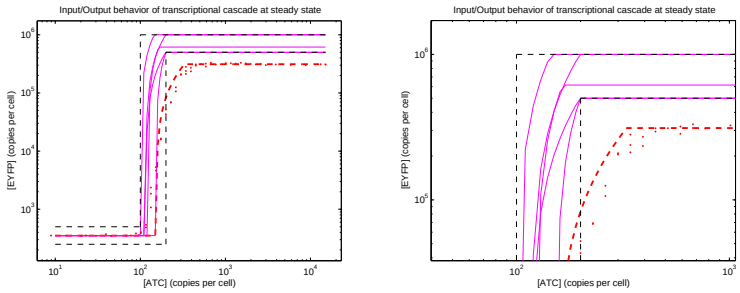
$$\dot{x}_{lacI} = \kappa^0_{lacI} + \kappa_{lacI}(1 - r^+(x_{tetR}, \theta^1_{tetR}, \theta^2_{tetR})\, r^-(u_{aTc}, \theta^1_{aTc}, \theta^2_{aTc})) - \gamma_{lacI}\, x_{lacI}, \tag{4}$$

$$\dot{x}_{cI} = \kappa^0_{cI} + \kappa_{cI}\; r^-(x_{lacI}, \theta^1_{lacI}, \theta^2_{lacI}) - \gamma_{cI}\, x_{cI}, \tag{5}$$

$$\dot{x}_{eyfp} = \kappa^0_{eyfp} + \kappa_{eyfp}\; r^-(x_{cI}, \theta^1_{cI}, \theta^2_{cI}) - \gamma_{eyfp}\, x_{eyfp}, \tag{6}$$

$(\theta^1_{aTc}, \theta^2_{aTc}) = (80, 4000)$; $(\kappa_{tetR}, \gamma_{tetR}, \theta^1_{tetR}, \theta^2_{tetR}) = (260, 0.013, 4500, 5500)$; $(\kappa^0_{lacI}, \kappa_{lacI}, \gamma_{lacI}, \theta^1_{lacI}, \theta^2_{lacI}) = (2.40, 875.6, 0.013, 500, 4500)$; $(\kappa^0_{cI}, \kappa_{cI}, \gamma_{cI}, \theta^1_{cI}, \theta^2_{cI}) = (3.90, 386, 0.013, 600, 23000)$; $(\kappa^0_{eyfp}, \kappa_{eyfp}, \gamma_{eyfp}, \theta^1_{eyfp}, \theta^2_{eyfp}) = (4.58, 4048, 0.013)$

(b)



(c) (d)

Fig. 5. (a) Synthetic transcriptional cascade made of four genes. *tetR* inhibits *lacI*, *lacI* inhibits *cI*, and *cI* inhibits *eyfp*. The input aTc relieves the inhibition of *lacI* by TetR. The fluorescence of the protein EYFP is the output. (b) PMA model. Equation (4) states that *lacI* is repressed when the protein TetR is present and aTc absent. (c) I/O response of the cascade at steady state (zoomed in (d)). Measured (red dots), predicted (red line) and expected (region delimited by dashed lines) behaviors of the actual network. Predicted (magenta) behaviors for different parameters in the set P_1.

The cascade is ultrasensitive: the steady-state I/O behavior is such that the output (EYFP) undergoes a dramatic change for a moderate change of the input (aTc) in a transition region. The cascade is expected to present at least a 1000-fold increase of the output value for a two-fold increase of the input value. Using FGp ("eventually, p will be always true") to express that property p holds at equilibrium, the specifications in Figure 5(c) can be translated in LTL as follows.

$$\begin{aligned}
\phi_2 = \quad & u_{aTc} < 100 \rightarrow \mathrm{FG}(x_{eyfp} > 2.5\,10^2 \wedge x_{eyfp} < 5\,10^2) \\
\wedge \;\; & 100 < u_{aTc} < 200 \rightarrow \mathrm{FG}(x_{eyfp} > 2.5\,10^2 \wedge x_{eyfp} < 10^6) \\
\wedge \quad & u_{aTc} > 200 \rightarrow \mathrm{FG}(x_{eyfp} > 5\,10^5 \wedge x_{eyfp} < 10^6).
\end{aligned}$$

The actual network does not meet its specifications. So, we tried to tune it by finding valid parameter sets for property ϕ_2 (Problem 2). Using RoVerGeNe, we found a valid set, P_1, by tuning three production rate parameters:

$$P_1 : 1832.43 < \kappa_{lacI} < 3350.62,\ 393.46 < \kappa_{cI} \text{ and } 6495.42 < \kappa_{eyfp} < 12995.42$$

In order to evaluate the significance of these constraints, we computed by numerical simulation the steady-state I/O behavior of the system for different parameters in P_1, notably using extreme values (Figure 5(c)). This clearly reveals that relevant constraints on the parameters have been identified by our method.

With a partition of the state space having 1500 rectangles, 18 affine predicates on parameters were found, defining $> 200\,000$ equivalence classes. The computation lasted < 2 hours (PC, 3.4 GHz processor, 1 Gb RAM) and only 350 different parameter sets were analyzed. This computational time can be considered as very reasonable, given the difficulty of the problem: we systematically explore a 3-dimensional parameter space, testing a non-trivial dynamical property for any initial condition in a 5-dimensional (1 input and 4 state variables) state-space. As explained elsewhere [8, 18], we have also been able to assess the robustness of the network with 11 uncertain parameters.

6 Discussion

We have presented a method for the analysis of genetic regulatory networks with parameter uncertainty. Given a PMA model, a property expressed in LTL over rectangular predicates and a polyhedral parameter set, the proposed approach can be used to test whether the property is satisfied for every parameter in the parameter set -the set is then called valid-, or to find valid subsets of the given parameter set. To do so, we use a discrete abstraction $T_{\mathcal{R}}(p)$ of an embedding continuous transition system $T_{\mathcal{X}}(p)$ to define an equivalence relation on parameters p, in the sense that two equivalent parameters are associated to the same discrete abstraction. Then we define discrete transition systems, $T_{\mathcal{R}}^{\exists}(P)$ and $T_{\mathcal{R}}^{\forall}(P)$, that over- and under-approximate $T_{\mathcal{R}}(p)$ with parameter p in a set P, and show how they can be used to search the parameter space efficiently. The proposed approach is conservative: if a parameter set is found, it is guaranteed to be valid. However, not all valid parameter sets are guaranteed to be found. The method is implemented in a publicly-available tool called RoVerGeNe, and its practical applicability and biological relevance is demonstrated on the tuning of a synthetic network build in *E. coli*. Network tuning is a central problem in synthetic biology, since most initial attempts at constructing gene networks do not result in a system exhibiting the desired behavior [2].

Other approaches have been proposed for the verification of continuous or hybrid systems with parameter uncertainties. In most approaches, unknown parameters are represented as symbolic constants, and symbolic operations are used to manipulate sets of states and compute (approximations of) sets of predecessors or successors[17, 20–23]. A major limitation is that the computational techniques supporting these symbolic operations currently apply only to systems having rather simple continuous dynamics, such as timed automaton [20,

21], linear hybrid automaton [22], piecewise-affine systems [17], or affine hybrid automaton [23]. Alternatively, numerical approaches have been proposed in which parameter uncertainties are captured by means of differential inclusions (*e.g.* $\dot{x} \in \text{hull}(\{f(x,p) \mid p \in P\})$) [24]. For large parameter sets, these approaches can be very conservative. In this paper, we propose an approach which is successively symbolic (parameter constraint synthesis) and numerical (transition systems computation). The results of the first step are used to refine the parameter set considered in the second step, in order to limit (though not eliminate) overconservatism, while preserving efficiency.

In the field of systems biology, several approaches use formal verification to analyze uncertain models, often with a focus on parameter identification. In [25, 26], solution trajectories are computed by numerical simulation for parameter values chosen in specified intervals. Model checking is used to select trajectories satisfying the expected properties. This approach applies to very general classes of models, but can not provide guaranties for dense sets of parameters. Alternatively, exhaustive search or symbolic computations have been used to obtain constraints on parameters of discrete models having finite parameter spaces [26, 27], or of piecewise-affine models having dense parameter spaces [23]. However, these models do not capture complex genetic regulations with graded responses, as in the transcriptional cascade example.

Motivated by applications in synthetic biology, we view two directions for further work. A first improvement would be to deal also with uncertain *threshold* parameters. A second desirable extension would be to allow for the verification of the frequently-encountered properties involving timing constraints.

Acknowledgements We thank Boyan Yordanov for contributions to the development of earlier versions of the model and financial support by NSF 0432070.

References

1. Kitano, H.: Systems biology: A brief overview. Science **295**(5560) (2002) 1662–1664
2. Andrianantoandro, E., Basu, S., Karig, D., Weiss, R.: Synthetic biology: New engineering rules for an emerging discipline. Mol. Syst. Biol. (2006)
3. Szallasi, Z., Stelling, J., Periwal, V., eds.: System Modeling in Cellular Biology: From Concepts to Nuts and Bolts. MIT Press (2006)
4. de Jong, H., Ropers, D.: Qualitative approaches to the analysis of genetic regulatory networks. In [3] (2006) 125–148
5. Belta, C., Habets, L.C.G.J.M.: Controlling a class of nonlinear systems on rectangles. IEEE Trans. Aut. Control (2006) In press.
6. Alur, R., Henzinger, T.A., Lafferriere, G., Pappas, G.J.: Discrete abstractions of hybrid systems. Proc. IEEE **88**(7) (2000) 971–984
7. Clarke, E.M., Grumberg, O., Peled, D.A.: Model Checking. MIT Press (1999)
8. Batt, G., Belta, C.: Model checking genetic regulatory networks with applications to synthetic biology. CISE Tech. Rep. 2006-IR-0030, Boston University (2006)
9. de Jong, H., Gouzé, J.-L., Hernandez, C., Page, M., Sari, T., Geiselmann, J.: Qualitative simulation of genetic regulatory networks using piecewise-linear models. Bull. Math. Biol. **66**(2) (2004) 301–340

10. Abate, A., Tiwari, A.: Box invariance of hybrid and switched systems. In: Proc. ADHS'06. (2006)
11. Belta, C., Habets, L.C.G.J.M., Kumar, V.: Control of multi-affine systems on rectangles with applications to hybrid biomolecular networks. In: Proc. CDC'02. (2002)
12. Mestl, T., Plahte, E., Omholt, S.: A mathematical framework for describing and analysing gene regulatory networks. J. Theor. Biol. **176** (1995) 291–300
13. Glass, L., Kauffman, S.: The logical analysis of continuous non-linear biochemical control networks. J. Theor. Biol. **39**(1) (1973) 103–129
14. Kloetzer, M., Belta, C.: Reachability analysis of multi-affine systems. In Hespanha, J., Tiwari, A., eds.: Proc. HSCC'06. LNCS 3927, Springer (2006) 348–362
15. Alur, R., Dang, T., Ivancic, F.: Progress on reachability analysis of hybrid systems using predicate abstraction. In Pnueli, A., Maler, O., eds.: Proc. HSCC'03. LNCS 2623, Springer (2003) 4–19
16. Koutsoukos, X., Antsaklis, P.J.: Safety and reachability of piecewise linear hybrid dynamical systems based on discrete abstractions. J. Discrete Event Dynamic Systems **13**(3) (2003) 203–243
17. 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 *E. coli*. Bioinformatics **21**(Suppl.1) (2005) i19–i28
18. Batt, G., Belta, C., Weiss, R.: Model checking liveness properties of genetic regulatory networks. Submitted. http://iasi.bu.edu/∼batt/ (2006)
19. Hooshangi, S., Thiberge, S., Weiss, R.: Ultrasensitivity and noise propagation in a synthetic transcriptional cascade. Proc. Natl. Acad. Sci. USA **102**(10) (2005) 3581–3586
20. Annichini, A., Asarin, E., Bouajjani, A.: Symbolic techniques for parametric reasoning about counter and clock systems. In Emerson, E., Sistla, A., eds.: Proc. CAV'00. LNCS 1855, Springer (2000) 419–434
21. Wang, F.: Symbolic parametric safety analysis of linear hybrid systems with BDD-like data-structures. IEEE Trans. Softw. Eng. **31**(1) (2005) 38–51
22. Henzinger, T., Ho, P.-H., Wong-Toi, H.: HYTECH: A model checker for hybrid systems. Software Tools Technology Transfer **1**(1-2) (1997) 110–122
23. Ghosh, R., Tomlin, C.J: Symbolic reachable set computation of piecewise affine hybrid automata and its application to biological modelling: Delta-Notch protein signalling. IEE Proc. Syst. Biol. **1**(1) (2004) 170–183
24. Lin, H., Antsaklis, P.J.: Robust regulation of polytopic uncertain linear hybrid systems with networked control system applications. In Antsaklis, P., Liu, D., eds.: Stability and Control of Dynamical Systems with Applications. Birkhauser (2003)
25. Antoniotti, M., Piazza, C., Policriti, A., Simeoni, M., Mishra, B.: Taming the complexity of biochemical models through bisimulation and collapsing: Theory and practice. Theor. Comput. Sci. **325**(1) (2004) 45–67
26. Calzone, L., Chabrier-Rivier, N., Fages, F., Soliman, S.: Machine learning biochemical networks from temporal logic properties. Trans. Comput. Syst. Biol. (2006) In press.
27. 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. J. Theor. Biol. **229**(3) (2004) 339–347