A Formal Verification Approach to the Design of Synthetic Gene Networks

Boyan Yordanov and Calin Belta

Abstract— **The design of genetic networks with specific functions is one of the major goals of synthetic biology. However, constructing biological devices that work "as required" remains challenging, while the cost of uncovering flawed designs experimentally is large. To address this issue, we propose a fully automated framework that allows the correctness of synthetic gene networks to be formally verified *in silico* from rich, high level functional specifications.**

Given a device, we automatically construct a mathematical model from experimental data characterizing the parts it is composed of. The specific model structure guarantees that all experimental observations are captured and allows us to construct finite abstractions through polyhedral operations. The correctness of the model with respect to temporal logic specifications can then be verified automatically using methods inspired by model checking.

Overall, our procedure is conservative but it can filter through a large number of potential device designs and select few that satisfy the specification, to be implemented and tested further experimentally. As illustration, our methods are applied to the design of a simple synthetic gene network.

I. INTRODUCTION

Synthetic biology is an emerging field that focuses on the rational design of biological systems. A number of *biological devices* - gene networks engineered for a specific function - have been constructed (see [21] for a review) but success stories have largely been the result of extensive experimentation. As the field matures, a systematic approach that enables the implementation of complicated designs into functionally correct devices with less experimental work is needed. One approach enabled by biological standards [16] and (online) libraries [1] involves the modular design and construction of devices from *biological parts* - genetic sequences known to function as promoters, ribosome binding sites, coding sequences, etc. Using bio-design automation (BDA) platforms such as `Clotho` [11], parts can be retrieved from online libraries and devices can be designed, checked against rules of correct assembly and implemented automatically. Even so, BDA platforms asses potential devices in terms of their assembly feasibility but not based on their correctness with respect to specifications of required function.

Designing biological devices that work "as required" remains challenging and is usually approached through modeling to minimize costly experimentation (see [9] for a review of modeling formalisms). A realistic model is needed to guide design efforts but such models are often hard to analyze. In addition, estimating model parameters may require extensive experimental data which is rarely available, although characterizations resulting in biological part data sheets [7] are currently ongoing. Besides selecting a realistic yet analytically tractable model, specifying the required device behavior in a formalism that is both general and allows for automatic analysis procedures is a separate challenge. In this paper, we propose a fully automated framework for *in silico* verification of synthetic gene networks from rich, high level specifications expressed as temporal logic formulas.

Temporal logics [8] are customarily used for specifying the correctness of digital circuits and computer programs. Due to their expressivity and resemblance to natural language they have gained popularity in other areas including the specification and analysis of qualitative behavior of genetic networks [2], [4], [3]. There also exist off-the-shelf algorithms for verifying the correctness of a finite state system for a temporal logic specification (model-checking [8]). However, such finite models are usually too simple to capture the dynamics of genetic networks with the detail necessary for design applications.

In our previous work [24] we used piecewise affine (PWA) systems as models of gene networks [23]. Such systems are globally complex and can approximate nonlinear dynamics with arbitrary accuracy [17], which makes them realistic models. They are also locally simple, which allowed us to analyze them formally from temporal logic specifications through a procedure based on the construction and refinement of finite abstractions through polyhedral operations [24] and model-checking [8]. In this paper, we use a class of models that is inspired by PWA systems but is more general. To account for the variability due to experimental conditions and the uncertainty inherent in biological systems we allow model parameters to vary in some ranges. We develop a procedure for the automatic construction of such models from part characterization data with the guarantee that all experimental observations can be reproduced by the identified model. We also extend our methods from [24] and integrate them with our model identification procedure, which leads to a fully automatic framework for specifying and verifying the correctness of genetic networks constructed from parts. Our approach can be used both to verify individual device designs or to automatically explore the space of potential device designs that can be constructed from characterized parts, available from libraries.

In terms of analysis, our method is related to tools such as the `Genetic Network Analyzer (GNA)` [10] and `RoVerGeNe` [5], which study biological systems using tem-

This work was supported by NSF grant CNS-0834260. B. Yordanov (`yordanov@microsoft.com`) was with the Department of Mechanical Engineering, Boston University, Boston MA and is now with Microsoft Research, Cambridge UK. C. Belta (`cbelta@bu.edu`) is with the Department of Mechanical Engineering, Boston University, Boston MA

poral logic specifications but usually focus on the analysis of separate devices for which a model is available. Instead, our procedure can explore different devices constructed from a set of parts, while models are derived automatically from part characterization data. In that respect, our approach is related to methods such as [18] that focus on the generation and analysis of analog and mixed-signal circuit models from simulation traces. Unlike other gene network modeling approaches, we do not enforce sigmoidal (Hill) regulation functions but construct models which capture all experimental observations and resemble uncertain parameter PWA systems. This relates our procedure to other PWA identification methods (see [15] for a review). Such tools address the threshold reconstruction problem more rigorously but only identify fixed parameter models and require device experimental data which is not usually available during design. This motivates the development of our procedure for constructing device models from part characterization data.

Compared to other tools such as `GEC` [20] and `GenoCAD` [6] that allow the computational study of biological devices constructed from parts, our approach differs in three major aspects (see [25] for details). First, we only assume that protein degradation rates are known (they are often available from literature or can be predicted computationally [14]) but construct device models automatically from the available experimental part characterization data, which makes our approach easier to apply in practice. Second, instead of relying on numerical simulation for model analysis, we construct models which can be analyzed formally (using model-checking based methods, developed previously in [24] and extended in this paper) but are also rich enough to capture all experimental observations. Finally, we formalize high level specifications in linear temporal logic which is both rich (*i.e.* it captures many properties of interest) and "user-friendly" (*i.e.* it resembles natural language).

Throughout the rest of the paper we use the following notation. Given a set S we use $|S|$ and 2^S to denote the cardinality and the powerset (the set of subsets) of S, respectively. For a set $S \subset \mathbb{R}^N$ and a scalar $\lambda \in \mathbb{R}$, we use λS to denote the set of elements from S multiplied by λ. Given sets S and S' we denote their Minkowski (set) sum by $S+S'$. Given polytope X, we denote the set of vertices of X by $\mathcal{V}(X)$ and their convex hull as $X = hull(\{v \in \mathcal{V}(X)\})$.

II. PROBLEM FORMULATION

In this section we formulate the problem of verifying the correctness of a gene network (biological device) from high level specifications. We start by discussing our simplified view of the biochemistry involved in gene expression, the parts we consider as basic building blocks of all devices and the experimental data that we assume is available.

We consider only two basic types of biological parts - sequences of DNA that either function as *promoters* or code for proteins (we refer to such sequences as *genes*). This is the minimal set of parts required to define the interactions in gene networks but the methods we subsequently develop can be extended easily for more detailed formulations. We assume that each gene codes for a single protein which degrades at a certain rate and whose concentrations can be measured directly in experiments. We treat protein production as a single step process, which is sufficient to capture transcriptional regulation (see Fig. 1).

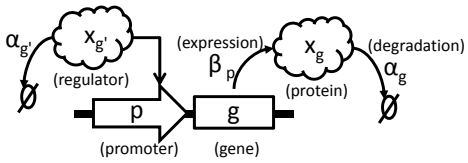
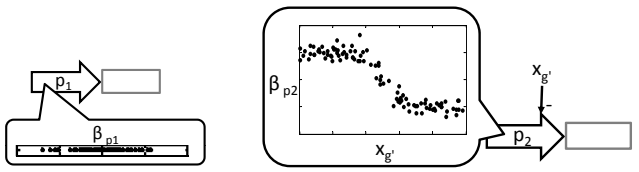


Fig. 1. In the simplified expression mechanism we consider, a gene g is expressed from promoter p at rate β_p to produce protein, whose concentration is denoted as x_g. The protein degrades at rate α_g. The promoter might be regulated by protein $x_{g'}$.

For a protein to be produced, its corresponding gene must be expressed, which requires placing it after a promoter (we assume that other sequences required for correct expression such as a ribosome binding site are already contained within a gene). The simplest device we consider contains a single promoter and expresses a single gene to produce a single protein (Fig. 1). By placing multiple genes on the same promoter and including additional promoters, more complicated devices can be assembled. We assume that, in a device, a gene is expressed from a single promoter - such assembly constraints are handled by platforms such as `Clotho` [11].

We differentiate between *constitutive* and *regulated* promoters. A protein is always produced if its gene is expressed from a constitutive promoter (*i.e.* the promoter is always "on"), while expression from a regulated promoter varies, depending on the concentrations of proteins or chemicals (inducers), called *regulators*. In general, a promoter can be regulated by several regulators but, for simplicity of presentation in this paper we consider only the case of a single regulator per promoter, although our methods can also be extended for the more general case.

We consider only devices built from *characterized parts* - genes and promoters for which experimental data indicative of their performance is available. A gene is characterized by the degradation rate (or equivalently the half-life) of the protein it codes for, which we assume is a fixed and known value. Protein degradation rates are often available from literature or can be predicted computationally [14]. A promoter is characterized by a rate of expression, which we assume is the same for all genes expressed from it. However, because of variability in experimental conditions and the inherent uncertainty of biological systems, we assume that the rate of expression from a promoter varies in a certain range. For a constitutive promoter, the characterization data is simply a set of experimentally measured expression rates (Fig. 2(a)), while for a regulated promoter, we assume that experimental measurements of the expression rate at different concentrations of the regulator are available (Fig. 2(b)). Measuring expression rates directly can be challenging and such data is usually obtained by simultaneously measuring the concentration of a regulator and a gene expressed from the regulated promoter [22]. In Sec. III we provide a procedure for converting such measurements to the expression rates



(a) Constitutive promoter (b) Regulated promoter

Fig. 2. (a) Experimental observations in the form of a range of values of the rate of expression β_{p1} characterize the constitutive promoter p_1. (b) The regulated promoter p_2 is characterized by experimental observations of the rate of expression β_{p2} at different concentrations of the regulator (in this case, repressor) $x_{g'}$.

data shown in Fig. 2, which we assume is available for all characterized promoters.

Given a device, we are interested in studying the dynamics of the concentrations of proteins expressed from its genes. Let G denote the set of genes where $N = |G|$ is the device size. We use x_g to denote the concentration of the protein expressed from gene $g \in G$, which is bounded in a physiologically relevant range $x_g^{min} \leq x_g \leq x_g^{max}$. The hyper-rectangle

$$\mathcal{X} = [x_{g_1}^{min}, x_{g_1}^{max}] \times \ldots \times [x_{g_N}^{min}, x_{g_N}^{max}] \quad (1)$$

is the feasible state space of the device, where each $x \in \mathcal{X}$ is a vector of the concentrations of all proteins $x_g, g \in G$. Given an initial state $x(0) \in \mathcal{X}$, the concentrations of species from G evolve over time and produce an infinite *trajectory* $x(0), x(1), \ldots$, where $x(k) \in \mathcal{X}$ is the state at step k.

We define a set of atomic propositions Π as a set of linear inequalities

$$\Pi = \{\pi_i, i = 1, \ldots, K\}, \pi_i = \{x \in \mathcal{X} \mid c_i^T x + d_i \leq 0\}. \quad (2)$$

In other words, each atomic proposition π_i partitions the feasible space $\mathcal{X}$ into a satisfying and violating subset for π_i. Given a state $x \in \mathcal{X}$ we write $x \vDash \pi_i$ if and only if $c_i^T x + d_i \leq 0$ (*i.e* x satisfies π_i). A trajectory $x(0), x(1), \ldots$ produces an infinite word $w(0), w(1), \ldots$ where $w(k) = \{\pi \in \Pi \mid x(k) \vDash \pi\}$ is the set of propositions satisfied at step k.

To specify temporal logic properties of trajectories of the system we use Linear Temporal Logic [8]. Informally, LTL formulas over Π are inductively defined by using the standard Boolean operators (*e.g.,* $\neg$ (negation), $\vee$ (disjunction), $\wedge$ (conjunction)) and temporal operators, which include $\bigcirc$ ("next"), U ("until"), $\Box$ ("always"), and $\Diamond$ ("eventually"). LTL formulas are interpreted over infinite words, as those generated by the system. For example, the word $w(0), w(1), \ldots$ where $w(0) = \{\pi_1, \pi_2\}, w(1) = \{\pi_1, \pi_2, \pi_3\}$, and $w(2), w(3), \ldots = \{\pi_1, \pi_4\}$ satisfies formulas $\Box\pi_1, \Diamond\pi_3, \Diamond\Box(\pi_1 \wedge \pi_4)$, and $\pi_2 \mathsf{U} \pi_4$ but violates $\Box\pi_2$ and $\Diamond\pi_5$. We say that a trajectory $x(0), x(1), \ldots$ satisfies an LTL formula ϕ if and only if the corresponding word $w(0), w(1), \ldots$ satisfies ϕ. The device satisfies ϕ from a given region $X \subseteq \mathcal{X}$ if and only if all trajectories originating in X satisfy the formula.

We are now ready to formulate the main problem we consider in this paper:

Problem 1: Given a device constructed from characterized parts and a specification expressed as an LTL formula over a set of linear inequalities in the concentrations of proteins, determine if the device satisfies the specification.

Our approach to Problem 1 consists of two main steps. Given a device, we first use the characterization data available for its parts to automatically construct a mathematical model by applying the procedure we develop in Sec. III. The particular model structure we enforce allows us to guarantee that all experimental observations can be reproduced by the identified model. As a second step, we also exploit this structure to analyze the model from the temporal logic specification using a method inspired by model-checking, which we described in [24] but review and extend in Sec. IV. Our analysis procedure results in the computation of a satisfying (respectively, violating) region - a subset of the system's state space from which all trajectories are guaranteed to satisfy (respectively, violate) the specification. A device design is considered "good" if analysis reveals a large satisfying region and an empty or small violating region, while a design is "bad" whenever a substantial violating region is found. Given a library of characterized parts, our overall procedure can serve to evaluate a large number of possible device designs in order to select few for further experimental testing.

III. MODEL CONSTRUCTION

In this section, we describe our procedure for the automatic construction of device models from part characterization data. As it will become clear later, the resulting models capture all experimental observations and take the form of uncertain parameter systems with different dynamics in different regions of the state space.

In Sec. II we considered a simplified mechanism of gene expression (Fig. 1). A gene g was expressed from promoter p at rate β_p to make protein whose concentration was denoted by x_g and which degraded at rate α_g. We can express the dynamics of protein concentration as

$$x_g(k+1) = \alpha_g x_g(k) + \beta_p. \quad (3)$$

In the problem formulation of Sec. II, we assumed that, for each gene (protein) g, α_g has a fixed value which is known for characterized parts, but β_p is allowed to vary in some range, which is unknown and must be computed from the promoter characterization data (Fig. 2).

We first consider the computation of a range $B_p^c \subset \mathbb{R}$ for a constitutive promoter p, such that $\beta_p \in B_p^c$ in Eqn. (3). Then, we consider a regulated promoter where $\beta_p \in B_p(x_{g'})$ (*i.e.* the range of allowed rates $B_p(x_{g'}) \subset \mathbb{R}$ is, in general, a function of the regulator concentration $x_{g'}$). For both, we first discuss the case when experimental observations of expression rates are directly available and later extend our procedure to compute such measurements indirectly from more realistic experimental data. We conclude the section by discussing the construction of models for general devices composed of a number of characterized parts. In developing our model identification procedure, we seek to compute a range of expression rates that is tight but contains all experimental measurements. This leads to the construction of models that can reproduce all observed behavior but are not

overly general, which would make their subsequent analysis in Sec. IV too conservative. In the following, we denote measured expression rates and protein concentrations from promoter p and gene g by $\tilde{\beta}_p$ and $\tilde{x}_g$, respectively.

A. Constitutive promoter

If promoter p is constitutive, expression rate β_p does not depend on the concentrations of other species in the system (there are no regulators) but varies in range B_p^c. If a data set $D_p^c = \{\tilde{\beta}_p(1), \ldots \tilde{\beta}_p(n)\}$ of experimentally measured expression rates is available (Fig. 2(a)), this range is simply $B_p^c = [min(D_p^c), max(D_p^c)]$, which captures all experimentally observed rates and extrapolates under the assumption that any rate between the minimal and maximal observed one is also possible for the system.

In general, expression rates cannot be measured directly and must be computed from protein concentration measurements [22]. If gene g is expressed from constitutive promoter p, given a finite trajectory fragment $\tilde{x}_g(0), \tilde{x}_g(1), \ldots, \tilde{x}_g(n+1)$ observed in experiments[1], from Eqn. (3) it follows that the expression rate $\tilde{\beta}_p(k)$ observed at step $k = 0, \ldots, n$ is

$$\tilde{\beta}_p(k) = \tilde{x}_g(k+1) - \alpha_g \tilde{x}_g(k). \tag{4}$$

B. Regulated promoter

For a regulated promoter p, the rate of expression β_p varies in a range $B_p(x_{g'})$, which is a function of the regulator concentration $x_{g'}$. Range $B_p(x_{g'})$ is unknown and must be computed from the available promoter characterization data $(\tilde{\beta}_p, \tilde{x}_{g'}) \in D_p$ (*i.e.* D_p is a set of expression rates measured at different repressor concentrations as in Fig. 2(b)). In the following, we focus on the construction of the set

$$\bar{B}_p = \{(\beta_p, x_{g'}) \mid x_{g'} \in [x_{g'}^{min}, x_{g'}^{max}], \beta_p \in B_p(x_{g'})\}. \tag{5}$$

This allows us to compute $B_p(x_{g'})$ at arbitrary concentrations $x_{g'}$ as the slice of $\bar{B}_p$ at $x_{g'}$ (*i.e.* $B_p(x_{g'}) = \{\beta_p \mid (\beta_p, x_{g'}) \in \bar{B}_p\}$). By constructing the tightest $\bar{B}_p$ that contains all experimental measurements (*i.e.* $D_p \subset \bar{B}_p$), we guarantee that the model we identify can reproduce all observed behavior but our subsequent analysis in Sec. IV is not overly conservative.

We introduce a set of thresholds $\theta_{g'}^i$ (computed as described in Sec. III-C), such that $x_{g'}^{min} \leq \theta_{g'}^i \leq x_{g'}^{max}$ for all $i = 1, \ldots, n_{g'}$ and $\theta_{g'}^i < \theta_{g'}^{i+1}$ for all $i = 1, \ldots, n_{g'} - 1$. For each regulator concentration region (*i.e.* when $\theta_{g'}^i < x_{g'} < \theta_{g'}^{i+1}$), we construct a trapezoid $\bar{B}_p^i$ that has the two thresholds as its bases and contains all expression rates observed in that region. Then, we have $\bar{B}_p = \bigcup_{i=1}^{n_{g'}-1} \bar{B}_p^i$ and, given regulator concentration $x_{g'}$ such that $x_{g'} = \lambda\theta_{g'}^i + (1-\lambda)\theta_{g'}^{i+1}$ for some $i = 1, \ldots, n_{g'} - 1$ and $\lambda \in [0, 1]$,

$$B_p(x_{g'}) = \lambda B_p(\theta_{g'}^i) + (1-\lambda) B_p(\theta_{g'}^{i+1}). \tag{6}$$

Additional details and other strategies for the computation of $\bar{B}_p^i$ and $\bar{B}_p$ are described in [25].

[1]When protein concentration measurements from individual cells are not available the method can still be applied but additional assumptions or computation are necessary as described in [25]

As for constitutive promoters, when expression rates are not available directly, they can be computed from protein concentration measurements. Given genes g,g' and a promoter p, such that g is expressed from p and g' regulates p, and a trajectory fragment $\tilde{x}(0), \tilde{x}(1), \ldots, \tilde{x}(n+1)$ where $\tilde{x}(k) = (\tilde{x}_g(k), \tilde{x}_{g'}(k))$ is a vector of regulator and protein concentrations, we have

$$\begin{aligned} D_p = \{(\tilde{\beta}_p(k), \tilde{x}_{g'}(k)) \mid \tilde{x}(k) = (\tilde{x}_g(k), \tilde{x}_{g'}(k)), \\ \tilde{\beta}_p(k) = \tilde{x}_g(k+1) - \alpha_g \tilde{x}_g(k), k = 1, \ldots, n\}. \end{aligned} \tag{7}$$

C. Device models

To summarize the construction of models using the procedures we discussed so far, we consider a device consisting of a set of genes G and promoters P (see the problem formulation in Sec. II). For notational simplicity, we assume that for $i = 1, \ldots, N$, gene $g_i \in G$ is expressed from promoter $p_i \in P$, which is either constitutive or regulated by the protein produced by gene $g_i' \in G$. We assume that, for each gene $g \in G$, we have at least two thresholds (*i.e.* $n_g \geq 2$) where $\theta_g^1 = x_g^{min}$ and $\theta_g^{n_g} = x_g^{max}$ (*i.e.* the boundaries of the state space $\mathcal{X}$ introduced in Sec. II are thresholds). Computing the set of thresholds is not the focus of this paper but related methods are available [12]. Here, we implement a sampling procedure where, out of a number of randomly generated thresholds, we select the subset of a given size that minimizes the volume of $\bar{B}_p$.

For a state $x \in \mathcal{X}_l$ where $x = (x_{g_1}, \ldots, x_{g_N})$, the dynamics of each component x_g are given by Eqn. (3) where

$$\beta_{p_i} \in \begin{cases} B_p^c & \text{if } p \text{ is constitutive or} \\ B_p(x_{g_i'}) & \text{if } p \text{ is regulated} \end{cases} \tag{8}$$

It is important to note that the identified model can reproduce all experimental data used for part characterization. Consider a trajectory fragment used in Eqn. (4) or (7) to respectively characterize a constitutive or regulated promoter. We can guarantee that the expression rate from the promoter, required to reach the concentration of the expressed protein observed at step $k+1$ starting from the concentration observed at step k, is always in the allowed range. In Sec. IV we will show that the model structure allows the computation of finite abstractions through polyhedral operations, enabling the application of formal analysis techniques.

IV. FORMAL ANALYSIS

In Sec. III we developed a procedure for the automatic construction of device models from part characterization data. All experimental measurements were captured in the resulting models by allowing expression rates to vary in certain ranges. In this section we show that, despite this uncertainty, finite quotients of the identified models can be constructed using polyhedral operations, which enables analysis through methods inspired by model checking. With the exception of Prop. 1, the material presented in this section is largely a review of our results from [24].

The state space $\mathcal{X}$ from Eqn. (1) is partitioned by the thresholds $\theta_g^i, i = 1, \ldots, n_g$ of all genes $g \in G$ into a number

of hyper-rectangular regions. We partition $\mathcal{X}$ further using all linear inequalities $\pi \in \Pi$ (Eqn. (2)) and ignore the measure-zero set consisting of all boundaries[2]. This results in a set of open polytopes $\mathcal{X}_l, l \in L$ such that, for all $l_1, l_2 \in L$, $\mathcal{X}_{l_1} \cap \mathcal{X}_{l_2} = \emptyset$ and $\cup_{l \in L} cl(\mathcal{X}_l) = \mathcal{X}$, where $cl()$ denotes the closure of a set. We denote the set $\cup_{l \in L} \mathcal{X}_l$ as $\bar{\mathcal{X}}$. Note that all states from a given region satisfy the same atomic propositions (*i.e.* for all $x_1, x_2 \in \mathcal{X}_l$ for some $l \in L$ and all $\pi \in \Pi$, $x_1 \vDash \pi$ if and only if $x_2 \vDash \pi$).

We define two states as equivalent if and only if they belong to the same region $\mathcal{X}_l$ for some $l \in L$. The finite, proposition preserving quotient induced by this equivalence relation is the transition system $T = (Q, \rightarrow, \Pi, \vDash)$ where $Q = L$ is the finite set of states, Π is the set of atomic propositions from Eqn. (2), and $\vDash \subseteq Q \times \Pi$ is the satisfaction relation[3] where, given $l \in L$ and $\pi \in \Pi$, $l \vDash \pi$ if and only if, for all $x \in \mathcal{X}_l$, $x \vDash \pi$. The transition relation $\rightarrow \subseteq Q \times Q$ is defined as $(l_1, l_2) \in \rightarrow$ if and only if there exists a transition from a state in region $\mathcal{X}_{l_1}$ to a state in $\mathcal{X}_{l_2}$. From this definition, it follows that T simulates the infinite system identified through our procedure from Sec. III (in other words, T can produce any word that the infinite system can produce [19]). This allows us to guarantee that if an arbitrary LTL formula ϕ is satisfied by T at state $l \in L$, then all trajectories of the system originating in region $\mathcal{X}_l$ satisfy the formula. Note that when T does not satisfy ϕ from state l we cannot say anything about the satisfaction of ϕ from region $\mathcal{X}_l$, which makes the overall method conservative.

In [24] we developed an analysis procedure based on the construction, model checking and refinement of simulation quotients such as T. Our algorithm used model checking to partition the set of states L into set $L^\phi \subseteq L$ from which T satisfied an LTL formula ϕ and $L^{\neg\phi} \subseteq L$ from which T satisfied the negation $\neg\phi$. This allowed us to guarantee that all trajectories originating in the *satisfying region* $\mathcal{X}^\phi = \bigcup_{l \in L^\phi} \mathcal{X}_l$ and none of the trajectories originating in the *violating region* $\mathcal{X}^{\neg\phi} = \bigcup_{l \in L^{\neg\phi}} \mathcal{X}_l$ satisfied ϕ. Both satisfying and violating trajectories originated in region $\bar{\mathcal{X}} \setminus (\mathcal{X}^\phi \cup \mathcal{X}^{\neg\phi})$ and our algorithm implemented an iterative refinement procedure to try and separate them, in which case $\mathcal{X}^\phi$ and $\mathcal{X}^{\neg\phi}$ can be expanded.

To apply our method from [24] (implemented as the software tool `FaPAS`) we need to be able to construct T, which reduces to the computation of its transitions $\rightarrow$. For all $l \in L$, we denote the set of states reachable from $\mathcal{X}_l$ in one step as $Post(\mathcal{X}_l)$. Transitions of T can be computed as

$$(l_1, l_2) \in \rightarrow \text{ if and only if } Post(\mathcal{X}_{l_1}) \cap \mathcal{X}_{l_2} \neq \emptyset. \quad (9)$$

To show that T can be constructed, we show that $Post(\mathcal{X}_{l_1}) \cap \mathcal{X}_{l_2}$ is computable for all $l_1, l_2 \in L$. We use the notation introduced in Sec. III where each promoter, gene and regulator is denoted by $p_i \in P$ and $g_i, g_i' \in G$, $i = 1, \ldots, N$, respectively. Given a state $x \in \mathcal{X}_l$ for some $l \in L$ such that $x = (x_{g_1}, \ldots, x_{g_N})$, the overall system dynamics are given by

$$x(k+1) \in Ax(k) + B(x(k)), \quad (10)$$

where A is the diagonal matrix of degradation rates $A = diag(\alpha_{g_1}, \ldots, \alpha_{g_N})$ and

$$B(x) = B_1(x_{g_1'}) \times \ldots \times B_N(x_{g_N'}) \text{ where} \quad (11)$$

$$B_i(x_{g_i'}) = \begin{cases} B_{p_i}^c & \text{if } p_i \text{ is constitutive or} \\ B_{p_i}(x_{g_i'}) & \text{if } p_i \text{ is regulated} \end{cases}$$

Proposition 1: For all $l \in L$, $Post(\mathcal{X}_l)$ is convex and computable as[4]

$$Post(\mathcal{X}_l) = hull(\{Av + B(v) \mid v \in \mathcal{V}(\mathcal{X}_l)\}).$$

Following from Prop. 1 (with a proof available in [25]), the intersection $Post(\mathcal{X}_{l_1}) \cap \mathcal{X}_{l_2}$ is convex and computable for all $l_1, l_2 \in L$. Then, transitions can be computed using Eqn. (9) which completes the construction of T and, therefore, the satisfying and violation regions of the system identified in Sec. III can be computed. The relative volumes of those regions can be used to determine if a device satisfies the specification, which provides a solution to Problem 1. The same approach can also be used to compare different designs constructed from a set of parts based on their satisfaction of a common specification. To illustrate this, in Sec. V we use our method to design a synthetic gene network.

V. DESIGN OF GENE NETWORKS

To illustrate our method, we apply it to the design of a bistable gene network inspired by the "genetic toggle switch" [13], which has the topology shown in Fig. 3. We start by constructing a library of parts, which includes three genes (denoted by $g_1, \ldots g_3$) and three promoters (denoted by $p_1, \ldots p_3$). We assume that the proteins produced from all three genes are stable and their degradation rates are negligible compared to the dilution due to cell growth, which leads to a protein half-life of 30 min - an estimate of the generation time of bacteria. All promoters are regulated and the protein produced by gene g_i represses promoter p_i.

To characterize the promoters in the library, we need to obtain experimental data of their expression rates at different repressor concentrations as described in Sec. II and III. We follow the strategy from [22] where a characterization device similar to the one from Fig. 1 is used. It consists of an arbitrary reporter protein that is expressed from the regulated promoter to be characterized. The regulator protein is initialized at high concentration but is not expressed[5] and, as a result, repressor concentration decreases over time due to degradation. By measuring both repressor and reporter concentrations simultaneously we can compute the promoter characterization data as in Eqn. (7). Through numerical

[2]It is unreasonable to assume that equality constraints can be detected in practice and, in general, trajectories of the system do not disappear in such measure zero sets.

[3]We abuse the notation and use symbol $\vDash$ to denote the satisfaction of a proposition by a state in the original infinite system and its abstraction T.

[4]Although set $B(v)$ might be different for different regions that share vertex $v \in \mathcal{V}(\mathcal{X}_l)$, from the index $l \in L$ it is always clear which $B(v)$ is used for the computation.

[5]Experimentally, this is accomplished by expressing the regulator from an externally controlled promoter, which is switched off.

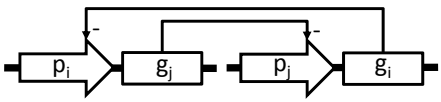


Fig. 3. Toggle switch. Two repressor proteins are expressed from two regulated promoters and mutually repress each other.

simulation of stochastic differential equations we generate a number of trajectories for each characterization device *in lieu* of single cell experimental measurements and use this information for model construction as described in Sec. III (plots of some characterization device trajectories and the constructed expression rate ranges are available in [25]).

We consider all "toggle switch" devices with topology as in Fig. 3 that can be constructed from the set of available parts. For device 1, gene g_2 is expressed from promoter p_1 and gene g_1 is expressed from promoter p_2. For device 2, g_3 is expressed from p_1 and g_1 is expressed from p_3. For device 3, g_2 is expressed from p_3 and g_3 is expressed from p_2. For each device, we consider specifications $\phi_1 = \diamondsuit\square\pi_1$ and $\phi_2 = \diamondsuit\square\pi_2$ where $\pi_1 := x_{g_i} \geq 2x_{g_j}$ and $\pi_2 := 2x_{g_i} \leq x_{g_j}$. In other words, specification ϕ_1 indicates that eventually and for all future times the concentration of protein x_{g_i} is at least twice greater than that of protein x_{g_j}, while ϕ_2 indicates the opposite. We seek a bistable device satisfying both specifications from different initial conditions.

Analysis using the procedure described in Sec. IV leads to the computation of the relative volumes of the satisfying and violating regions for all three devices for each of the two specifications (results are summarized in Table I). Analysis indicates that only device 3 is bistable, which is confirmed through simulations of the stochastic differential equation models of all three devices (results are available in [25]).

VI. CONCLUSION

In this paper, we presented an automated procedure for the design of functionally correct synthetic gene networks from parts. We formalized high level specifications of required device behavior as temporal logic formulas over linear inequalities in protein concentrations. We developed a procedure for the construction of device models from experimental data characterizing the different parts the devices were composed of. The identified models were related to PWA systems but allowed expression rates from promoters to vary in certain ranges and could capture all experimental observations. This model structure also allowed us to construct finite quotients through polyhedral operations. Such quotients could then be analyzed using methods inspired by model checking to compute a range of initial conditions from which all trajectories of the device model were guaranteed to satisfy (or violate) the specification. The relative sizes of those regions provided information about the correctness of a device design with respect to the specification. Our procedure could test individual, user-specified device designs or automatically search for correct devices by exploring the design space of devices constructed from a set of parts. Future research directions involve decreasing the conservatism of the method by quantifying the "likelihood" of different parameters and applying it to real experimental studies.

TABLE I

spec.	$\phi_1 = \diamondsuit\square\pi_1$		$\phi_2 = \diamondsuit\square\pi_2$	
device	satisfying	violating	satisfying	violating
1	0%	46.9%	35.1%	0%
2	0%	88.8%	88.8%	0%
3	8.4%	58.4%	26.7%	8.4%

REFERENCES

[1] Registry of Standard Biological Parts. http://partsregistry.org/.
[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. In *PSB'03*, volume 58, pages 116–127, 2003.
[3] G. Batt, C. Belta, and R. Weiss. Temporal Logic Analysis of Gene Networks Under Parameter Uncertainty. *IEEE Trans. Autom. Control*, 53(Special Issue):215–229, 2008.
[4] 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 E. coli. *Bioinformatics*, 21 Suppl 1:i19–28, June 2005.
[5] G. Batt, B. Yordanov, R. Weiss, and C. Belta. Robustness analysis and tuning of synthetic gene networks. *Bioinformatics*, 23(18):2415–22, Sept. 2007.
[6] Y. Cai, M. W. Lux, L. Adam, and J. Peccoud. Modeling structure-function relationships in synthetic DNA sequences using attribute grammars. *PLoS computational biology*, 5(10):e1000529, Oct. 2009.
[7] B. Canton, A. Labno, and D. Endy. Refinement and standardization of synthetic biological parts and devices. *Nature biotechnology*, 26(7):787–93, July 2008.
[8] E. M. Clarke. *Model Checking*. MIT Press, 1999.
[9] H. de Jong. Modeling and Simulation of Genetic Regulatory Systems: A Literature Review. *J. Comput. Biol.*, 9(1):67–103, 2002.
[10] H. de Jong. Genetic Network Analyzer: qualitative simulation of genetic regulatory networks. *Bioinformatics*, 19(3):336–344, 2003.
[11] D. Densmore, A. Van Devender, M. Johnson, and N. Sritanyaratana. A platform-based design environment for synthetic biological systems. In *TAPIA '09*, page 24, New York, New York, USA, 2009. ACM Press.
[12] S. Drulhe, G. Ferrari-Trecate, and H. de Jong. The Switching Threshold Reconstruction Problem for Piecewise-Affine Models of Genetic Regulatory Networks. *IEEE Trans. Autom. Control*, 53(Special Issue):153–165, 2008.
[13] T. Gardner, C. Cantor, and J. Collins. Construction of a genetic toggle switch in Escherichia coli. *Nature*, 403:339–342, 2000.
[14] E. Gasteiger, C. Hoogland, A. Gattiker, S. Duvaud, M. R. Wilkins, R. D. Appel, and A. Bairoch. Protein Identification and Analysis Tools on the ExPASy Server. In J. M. Walker, editor, *The Proteomics Protocols Handbook*, pages 571–607. Humana Press, 2005.
[15] A. L. Juloski, W. Heemels, G. Ferrari-Trecate, R. Vidal, S. Paoletti, and J. H. G. Niessen. Comparison of Four Procedures for the Identification of Hybrid Systems. In *Hybrid Systems: Computation and Control*, volume 3414 of *LNCS*, pages 354–369. 2005.
[16] T. Knight. Idempotent Vector Design for Standard Assembly of Biobricks Standard Biobrick Sequence Interface, 2003.
[17] J. N. Lin and R. Unbehauen. Canonical piecewise-linear approximations. *IEEE Trans. Circuits Syst. I, Fundam. Theory Appl.*, 39(8):697–699, 1992.
[18] S. Little, D. Walter, K. Jones, C. Myers, and A. Sen. Analog/Mixed-Signal Circuit Verification Using Models Generated from Simulation Traces. *IJFCS*, 21(02):191, 2010.
[19] R. Milner. Communication and concurrency. *Prentice-Hall*, 1989.
[20] M. Pedersen and A. Phillips. Towards programming languages for genetic engineering of living cells. *Journal of The Royal Society Interface*, 6(Suppl 4):S437–S450, 2009.
[21] P. E. M. Purnick and R. Weiss. The second wave of synthetic biology: from modules to systems. *Nature reviews. Molecular cell biology*, 10(6):410–22, June 2009.
[22] N. Rosenfeld, J. W. Young, U. Alon, P. S. Swain, and M. B. Elowitz. Gene regulation at the single-cell level. *Science (New York, N.Y.)*, 307(5717):1962–5, Mar. 2005.
[23] E. Sontag. Nonlinear regulation: The piecewise linear approach. *IEEE Trans. Autom. Control*, 26(2):346–358, Apr. 1981.
[24] B. Yordanov and C. Belta. Formal Analysis of Discrete-Time Piecewise Affine Systems. *IEEE Trans. Autom. Control*, 55(12):2834–2840, Dec. 2010.
[25] B. Yordanov and C. Belta. A formal verification approach to the design of synthetic gene networks. Technical Report arXiv:1109.1275v1 [cs.SY], Sept. 2011.