Data-driven Verification of Synthetic Gene Networks

Ebru Aydin Gol, Douglas Densmore, and Calin Belta

Abstract— Automatic design of synthetic gene networks with specific functions is emerging as an area of interest in synthetic biology. Quantitative evaluation of gene network designs is a missing feature of the existing automatic design tools. In this work, we address this issue and present a framework to probabilistically analyze the dynamic behavior of a gene network against specifications given in a rich and high level language. Given a gene network built from primitive DNA parts, and given experimental data for the parts, the tool proposed here allows for the automatic construction of a stochastic model of the gene network and in silico probabilistic verification against a rich specification.

I. INTRODUCTION

Synthetic biology’s mandate is to forward engineer living systems using engineering principles. DNA parts are encapsulated, composed to realize novel functionality, and introduced into “host organisms” that carry out that functionality. These techniques have been recently used for a wide variety of applications [1]–[4]. Such engineered organisms can be coupled with mechanical and electronic systems to create a whole new class of cyber-physical systems not yet explored [5]. However, these systems will be extremely complex and current experimental methods are ad-hoc at best. Bio-design automation frameworks are emerging [6]–[9], which generate synthetic gene networks from specifications, assign these networks discrete DNA segments, and physically assemble these circuits. What is notably absent from these workflows is a verification stage which quantitatively evaluates these designs against their specification.

In [10], we used Linear Temporal Logic (LTL) as a specification language and discrete-time piecewise affine (PWA) systems with polyhedral parameter uncertainty as mathematical models for synthetic gene networks. We showed that such models can be derived from experimental data and checked against arbitrary LTL formulas by constructing finite abstractions. Due to the wide uncertainty ranges from the experimental data and the conservativeness of the approach, the results were inconclusive. In this paper, we propose to capture the distribution of the experimental data into stochastic discrete-time PWA models and to use probabilistic verification techniques to analyze the behavior of the system. We find that this approach is much more conclusive and reflects more closely the biology it is modeling.

The probabilistic verification of finite state stochastic systems, such as Markov chains, is a well understood problem [11]. There are efficient tools, such as PRISM [12], that model check the system against a probabilistic temporal logic property. However, such tools cannot deal with stochastic systems with infinite state spaces. In [13], the authors proposed to use partitions of the state space to produce abstractions in the form of Markov chains, which can then be model checked with an explicit error bound on the probability of satisfaction. However, due to the dependence of the error bound on the partition size, the computation of the abstraction is infeasible for high-dimensional systems.

An alternative approach to the probabilistic verification problem of systems with large or infinite state spaces is statistical model checking (SMC) [14], [15], which applies statistical inference techniques to solve the verification problem. Since SMC relies on the model checking results of sample system traces, the outcome is probabilistic in nature, i.e., it is correct with a certain probability. The key advantage of this technique is that it can handle complex and high-dimensional systems with infinite state spaces in an efficient way, since the computation necessary for trace generation and model checking can be parallelized.

In this work, SMC is used to verify the dynamic behavior of a synthetic gene network assuming that the gene network is built from primitive DNA parts for which experimental data exists. The composite behavior of these parts is captured in a stochastic dynamical system whose parameters are obtained from experimental data. We use this model to solve two problems. The first is a verification problem against a specification expressed as a probabilistic bounded LTL (PBLTL) formula, which is used to check the correctness of the design. The second is a parameter optimization problem, in which we use SMC to find time bounds and species concentration threshold values that make a formula satisfiable with a given probability. This problem allows us to tune a design parameter to improve the performance. The optimized parameters can be further used to compare gene networks designed to satisfy the same specification.

II. PROBLEM FORMULATION

A. Gene Network

A synthetic gene network (circuit) is composed of two basic biological parts (encapsulated DNA sequences): promoters and genes. A gene $g$ codes for a certain protein that degrades at a rate $\alpha_g$. The concentration of the protein is denoted by $x_g$ and it is assumed to be bounded in a relevant range $x_g^{\text{min}} \leq x_g \leq x_g^{\text{max}}$. The rate of expression of a gene is regulated by a promoter that precedes the gene (to the left when DNA is depicted visually) in the DNA. Regulators bind to the promoters and define regulations motifs. A regulator, which can be either a protein coded by a gene in the...
network or a small molecule (external regulator), can enable (activator) or disable (repressor) the ability of a promoter to initiate transcription (production of mRNA). The mRNA is translated into protein by the ribosome that recognizes and binds to the Ribosome Binding Site (RBS) of the mRNA. A promoter \( p \) can be regulated by multiple regulators, and the rate of expression \( \beta_p \) of a gene \( g \) that proceeds the promoter \( p \) depends on the concentrations of the regulators of the promoter \( p \). In our simplified description of gene regulation, the rate of expression captures both transcription and translation.

In the absence of arabinose, \( x_{\text{Ara}} < T_{\text{Ara}} \) for all \( i = 0, \ldots, k-1 \) and \( x_{\text{RFP}}(k) > T_{\text{RFP}} \).

The inverter gate specification from Example 2.1 can be formally stated as the following BLTL formula:

\[
\Phi_I = ((x_{\text{Ara}} < T_{\text{Ara}}) \rightarrow (F^k (G^2 (x_{\text{RFP}} > T_{\text{RFP}})))) \land ((x_{\text{Ara}} > T_{\text{Ara}}) \rightarrow (F^k (G^2 (x_{\text{RFP}} < T_{\text{RFP}})))).
\]

Formula \( \Phi_I \) requires the circuit to respond in \( k_1 \) time steps to the input and the output is interpreted as high (low) only when \( x_{\text{RFP}} > T_{\text{RFP}} \) and \( x_{\text{RFP}} < T_{\text{RFP}} \) are satisfied for \( k_2 \) consecutive time steps. Hence, \( k_1 \) can be considered as the response time and \( k_2 \) can be considered as the signaling time.

As explained above, a trajectory of a gene network can be checked against a BLTL formula. A gene network is inherently a stochastic system due to complex biochemistry involved in the protein production, e.g., a regulator binds to a promoter with a certain probability. For this reason, we use Probabilistic BLTL (PBLTL) to specify the behavior of a gene network.

A PBLTL formula is a formula of the form \( P_{\geq\theta}(\Phi) \), where \( \Phi \) is a BLTL formula and \( \theta \in [0, 1] \) is a probability. A gene network satisfies PBLTL formula \( P_{\geq\theta}(\Phi) \) if and only if the probability that a trajectory of the gene network satisfies BLTL formula \( \Phi \) is greater than or equal to \( \theta \).

C. Verification and Parameter Optimization

**Problem 2.1 (Verification):** Assume we have a gene network and a specification expressed as a BLTL formula \( \Phi \) over linear inequalities over the concentrations of the proteins and the external regulators.

(i) Assume a probability \( \theta \in [0, 1] \) is given. Decide whether the network satisfies the PBLTL formula \( P_{\geq\theta}(\Phi) \).

(ii) Compute the probability \( \theta \) with which the gene network satisfies the BLTL formula \( \Phi \).

In the second problem, our goal is to optimize a parameter that appears in the specification formula, i.e., either a threshold used in an inequality or a time bound of a temporal logic operator, while the rest of the parameters are assumed fixed.
Problem 2.2 (Parameter Optimization): Assume we have a gene network, a probability \( \theta \in [0,1] \), and a specification expressed as a BLTL formula \( \Phi \) in which the thresholds and the time bounds are fixed except one that is denoted as \( T \). Find the minimum (or maximum) value of \( T \) such that the gene network satisfies the PBLTL formula \( P_{>\theta}(\Phi) \).

To illustrate the usefulness of Problem 2.2, consider formula \( \Phi_I \) from Equation (2). Since \( T_{RFP}^H \) is used as a lower bound for \( x_{RFP} \), increasing \( T_{RFP}^H \) decreases the probability that a trajectory of the gene network satisfies \( \Phi_I \). On the other hand, the gene network works as an inverter if it satisfies \( P_{>\theta}(\Phi_I) \) for a high probability \( \theta \) when \( T_{RFP}^H > T_{RFP}^L \) and \( T_{ARA}^H > T_{ARA}^L \). We can use the solution of Problem 2.2 to find the maximum value for \( T_{RFP}^H \) for a given probability of satisfaction. Similarly, we can find the minimum value for \( T_{RFP}^L \) for a given probability of satisfaction. Solving these types of problems allows us to optimize the “qualitative” behavior of the circuit. Moreover, different network designs can be compared with respect to the optimized thresholds.

To provide solutions to the above problems, we assume that the degradation rates of all the proteins are (statistically) known (see Section III) and characterization data for all the promoters is available (more information on this type of data is given in Section III-A). We will use the available data to model the gene network as a discrete-time stochastic dynamical system. We will employ statistical model checking techniques to solve the problems presented above. Specifically, we will simulate the model, model check the produced trajectories against the specification formula, and use the sample set of model checking results to solve the problems by using statistical inference.

### III. MATHEMATICAL MODEL

A gene network \( S \) composed of \( n \) genes and \( s \) promoters is modeled by

\[
x_{g_i}(k + 1) = \alpha_{g_i}(k)x_{g_i}(k) + \beta_{\gamma(g_i)}(k), \quad i = 1, \ldots, n
\]

where \( x_{g_i}(k) \in [x_{g_i}^{\min}, x_{g_i}^{\max}] \) and \( \alpha_{g_i}(k) \in (0,1) \) are the concentration and the degradation rate, respectively, of the protein coded by gene \( g_i \) at time \( k \in \mathbb{Z}_+ \). Function \( \gamma : \{g_1, \ldots, g_n\} \rightarrow \{p_1, \ldots, p_s\} \) maps a gene to the promoter that regulates it, i.e., gene \( g_i \) is expressed at rate \( \beta_{\gamma(g_i)}(k) \) from the promoter \( \gamma(g_i) \) at time \( k \in \mathbb{Z}_+ \). The expression and the degradation rates are modeled by random variables, whose distributions depend on their value at the previous time step, i.e.,

\[
\alpha_{g_i}(k) \sim P_{\alpha_{g_i}}(\cdot | \alpha_{g_i}(k-1)), \quad i = 1, \ldots, n,
\]

\[
\beta_{p_j}(k) \sim P_{\beta_{p_j}}(\cdot | \beta_{p_j}(k-1)), \quad j = 1, \ldots, s,
\]

where \( \alpha_{g_i}(k) \) is a vector containing the concentrations of the regulators of promoter \( p_j \) at time \( k \in \mathbb{Z}_+ \). We use \( \pi_0 \) to denote the distribution of the initial states, and \( S^{\pi_0} \) to denote the system initialized at \( \pi_0 \).

The stochastic model from Equation (3) captures our simplified view of the gene expression mechanism introduced in Section II. It also allows us to capture that the degradation rate of a protein or an expression rate from a promoter can not change drastically in a short time period in a living cell.

In our subsequent analysis, we assume that the distributions of the degradation rates are known and characterization data for each promoter is available. The distributions of the degradation rates are often available in literature [16] or can be obtained computationally [17].

#### A. Promoter Characterization

Our promoters are characterized by a rate of expression that depends on the corresponding set of regulator concentrations and the probability that the regulators bind to the promoter. The relationship between the rate of expression from the promoter and the regulators can be captured from experimental data that simultaneously measures the concentrations of the regulators and the concentration of a protein whose expression is directly regulated by only the promoter. In our experimental set-up, a characterization circuit is constructed for each promoter. The characterization circuit involves the promoter and a gene coding for a fluorescent protein (a reporter protein). Thus, as the promoter is regulated to different levels of transcription, different levels of fluorescence will be observed. The cell culture is allowed to fluoresce, and then the fluorescence level in each cell is measured using a flow cytometer. The flow cytometer excites the fluorescent proteins with laser. The light emitted by the fluorescent proteins is measured and translated to fluorescence units [18]. The cells are assumed to be in steady state when the measurements are taken. From Equation (3), it follows that, for each gene \( g_i \):

\[
x_{g_i} = \alpha_{g_i}x_{g_i} + \beta_{\gamma(g_i)}(k).
\]

The characterization circuit for a promoter that is regulated by only an external regulator consists the promoter and a gene coding a fluorescent protein. In the case that the promoter is regulated by a protein, a more complex characterization circuit is required as the concentration of the regulator protein can not be controlled directly. The concentrations of both the regulator protein and a protein expressed by the promoter should be measured. Generally, the regulator protein can not be measured and a fluorescent protein is used as a reporter. Characterizing a promoter with multiple regulators requires combining the techniques explained above.

![Characterization circuit for a promoter regulated by a protein.](image)

**Example 3.1:** A circuit that is composed of the \( pBad \) promoter and a gene coding for a fluorescent gene such as \( GFP \) can be used to characterize \( pBad \) from Example 2.1. Since the \( pTet \) promoter is regulated by the \( tetR \) protein and
tetR is not fluorescent, the circuit given in Figure 3.1 is built in vivo and used to characterize both of the promoters as follows. A population of cells is partitioned into 7 parts and each part is subjected to a different arabinose concentration, (0, 0.5, 1, 2.5, 5, 7.5, 10) millimolar (mM) arabinose. Then, the green (GFP) and the red (RFP) fluorescent proteins are measured simultaneously in fluorescent units. The GFP data, $x_{GFP}$, obtained at different arabinose levels is used to characterize the $pBad$ promoter. The paired data ($x_{GFP}, x_{RFP}$) for all arabinose levels is used to characterize the $pTet$ promoter, where $x_{GFP}$ is used as the reporter of $x_{tetR}$.

We compute the distributions of the expression rates from the characterization data of the promoters as follows. Consider promoter $p$ that regulates gene $g$, and let $x_p = (x_{p,g}, x_{p,e})$ denote the set of concentrations of the corresponding regulators, where $x_{p,g} \in \mathbb{R}^n_g$ is the vector containing concentrations of the protein regulators, and $x_{p,e} \in \mathbb{R}^n_e$ is the vector containing concentrations of the external regulators. Even though a promoter usually has either one or two regulators, the general case, $n_g + n_e$, regulators is considered here.

To characterize the promoter, a set of concentration levels $\{x_{p,e,i}, \ldots, x_{p,e,1}\}$ is set for each external regulator $x_{p,e,i}$ and an experiment is conducted for each concentration combination $\{x_{p,e,1}, \ldots, x_{p,e,1}\}$, where $1 \leq c_i \leq b_i$ for all $i = 1, \ldots, n_e$. Hence, $\Pi_{i=1}^{n_e} b_i$ experiments are necessary to characterize the promoter. In each of these experiments, the concentrations of both regulator proteins (or a reporter protein), $x_{p,g}$, and the concentration of the protein coded by gene $g$, $x_g$, are measured, which results in the following data set:

$$D^{p,g,e} = \{(x_{p,g}, x_g)\}_{k \in \mathbb{Z}_+}, \quad x_{p,g} \in \mathcal{X}_{p,g}, \text{ where }$$

$$\mathcal{X}_{p,g} = \left\{x_{p,e,1}, \ldots, x_{p,e,1}\right\} \mid 1 \leq c_i \leq b_i, \forall i = 1, \ldots, n_e\}.$$

As mentioned before, we assume that the distribution of the degradation rate $P_{\alpha_g}(\cdot)$ is known and it is independent from the regulator concentrations. Therefore, we can compute the distribution of the expression rate conditioned on the regulator concentrations from $P_{\alpha_g}(\cdot)$, (4), and $D^{p,g,e}$ as:

$$P_{\beta_{p,g}}(\beta \mid x_{p,g}, x_{p,e}) = \int_{\alpha \in (0,1)} P_{\alpha_g}(\alpha) P_{x_g}(\beta \mid x_{p,g}, x_{p,e}).$$

The conditional distribution of $x_g$, $P_{x_g}(\cdot \mid x_{p,g}, x_{p,e})$, can be computed from the data set $D^{p,g,e}$ by using the Bayesian rule as follows:

$$P_{x_g}(x \mid x_{p,g}, x_{p,e}) = \frac{\text{Prob}(x \mid x_{p,g}, x_{p,e})}{\text{Prob}(x_{p,g} \mid x_{p,e})}.$$

This computation requires distribution fitting steps. To avoid additional computational burden and errors introduced by distribution fitting, we use the data directly to construct a piecewise constant conditional density function (a multi-dimensional histogram) for each $x_{p,e} \in \mathcal{X}_{p,e}$.

To construct the density function in the form of a multi-dimensional histogram, first, $P_{\alpha_g}(\cdot)$ is approximated by a histogram $H^{\alpha_g}$ with a sufficiently large number of intervals. Then for each measured point $(x_{p,g}, x_g) \in D^{p,g,e}$ and $\alpha_g$, which is the center point of each interval $i$ of $H^{\alpha_g}$, an expression rate is computed as

$$\beta_{p,i} = (1 - \alpha_{g,i})x_g.$$

These expression rates, the concentrations of the protein regulators $x_{p,g}$ and the frequency $f_i$ of the corresponding intervals are used to construct a data set $D^{p_g, \alpha_{p,g}}$. Specifically, for each expression rate $\beta_{p,i}$ as computed above, $(x_{p,g}, \beta_{p,i})$ is added to the set $D^{p_g, \alpha_{p,g}} f_i$ times. Finally, a multi-dimensional histogram $H^{p,g, \alpha_{p,g}}$ is constructed from the data set $D^{p_g, \alpha_{p,g}}$ for the expression rate $\beta_g$.

The presented method for constructing the conditional density function $H^{p,g, \alpha_{p,g}}$ has several advantages. First, it is faster than fitting multi-dimensional density functions given in (6) and provides a general method, since the shapes of these density functions are unknown. Second, by considering the discretization levels of the flow cytometry instrument and the intervals of $H^{\alpha_g}$, $H^{p,g, \alpha_{p,g}}$ can capture $P_{\beta_{p,g}}(\cdot \mid x_{p,g})$ precisely, where a deviation can only occur due to the approximation of $P_{\alpha_g}(\cdot)$. However, a degradation rate is either known or has a distribution with a low variance and compact support, i.e. $\text{supp}(P_{\alpha_g}(\cdot)) \subset (0, 1)$. If the degradation rate is known then the derived density function $H^{p,g, \alpha_{p,g}}$ is an exact representation of $P_{\beta_{p,g}}(\cdot \mid x_{p,g})$ with respect to the available data. If, however, $P_{\alpha_g}(\cdot)$ is given, $H^{p,g, \alpha_{p,g}}$ can approximate $P_{\beta_{p,g}}(\cdot \mid x_{p,g})$ with arbitrarily high accuracy, since $\text{supp}(P_{\alpha_g}(\cdot))$ is a compact set.

Example 3.2: The characterization data obtained as explained in Example 3.1 is used to construct histograms shown in Figure 3 for $\beta_{pBad}$ and $\beta_{pTet}$ by assuming that the degradation rates are known, i.e. for each $g \in \{tetR, GFP, RFP\}$ $P_{\alpha_g}(\bar{\alpha}_g) = 1$ for some $\bar{\alpha}_g \in (0, 1)$.

B. Gene Network Simulator

In this section, we describe a simulator that generates trajectories of the stochastic model defined in (3). The simulator is initialized by constructing the density functions from characterization data for each promoter as described in Section III-A. Then for a given trajectory length $N$, and initial state $x(0)$, first a degradation rate $\alpha_{g,i}$ for each gene $g_i$, $i = 1, \ldots, n$, and an expression rate $\beta_{p,j}$ for each promoter $p_j$, $j = 1, \ldots, s$ is sampled from the corresponding distributions and the state at time $k = 1$, $x(1)$, is computed according to (3). In the subsequent time steps, i.e. for $1 \leq k \leq N$, the random variables ($\alpha_{g,i}$, $\beta_{p,j}$) are sampled from a distribution that depends on the sampled value of the random variable in the previous iteration. In particular, we use truncated sampling that is explained next for a random variable $\alpha_g$.

Truncated sampling The value of $\alpha_g(k + 1)$ is sampled from the distribution of $\alpha_g$ truncated to the semi-open interval $[\alpha_g(k) - w, \alpha_g(k) + w]$, where $w \in \mathbb{R}_+$. Specifically, $\alpha_g(k + 1)$ is sampled from the distribution function

$$F_{\alpha_g}(\alpha)\{\alpha_g(k)\} = \frac{F_{\alpha_g}(\alpha) - F_{\alpha_g}(\alpha_g(k) - w)}{F_{\alpha_g}(\alpha_g(k) + w) - F_{\alpha_g}(\alpha_g(k) - w)},$$

where $F_{\alpha_g}(\alpha)$ is the cumulative distribution function of $\alpha_g$. In particular, if $\alpha_g$ is a standard normal distribution, the truncated sampling is a normal truncated sampling.
where $F_{\alpha}(\alpha) = P\{\alpha \leq \alpha\}$. Note that, when $t$ tends to infinity, $F_{\alpha}(\cdot|$\(\alpha(k))$ converges to $F_{\alpha}(\cdot$).

**Example 3.3:** We generate the trajectories of the gene network from Example 2.1 as explained above. The random variables $\beta_p, p = \{pBad, pTet\}$, are sampled from the histograms constructed as in Example 3.2. The expression rate from the $pTet$ promoter is sampled with respect to $x_{tetR}$. For example, if $x_{tetR} \in [3382.8, 4745.1)$, $\beta_{pBad}$ is sampled from the histogram shown in Figure 3 (d).

### IV. Statistical Analysis

In this section, we provide solutions to the problems given in Section II based on statistical analysis. We use the Bayesian Interval Estimation and Bayesian Hypothesis Testing algorithms presented in [15]. Both of the algorithms iteratively generate trajectories of system $S$ (3) and model check the trajectories against the specification formula.

**A. Statistical Hypothesis Testing**

Statistical hypothesis testing is a widely used tool to prove (with bounded error) statistical assumptions on a stochastic system. In [15] a statistical model checking algorithm based on iterative Bayesian Hypothesis Testing is proposed, where the hypothesis is defined as the satisfaction of a PBLTL formula $H_0 : S \models P_{>0}(\Phi)$, and the iterative algorithm decides between $H_0 : S \models P_{>0}(\Phi)$ and $H_1 : S \models P_{<0}(\Phi)$ for a stochastic system $S$.

We use the Bayesian Hypothesis Testing algorithm to solve Problem 2.1-(i). The algorithm sequentially draws a sample $\delta$ (model checking result of a trajectory $\sigma$),

$$\delta = \begin{cases} 1 & \text{if } \sigma \models \Phi \\ 0 & \text{otherwise} \end{cases}$$

updates the available data set $\Delta$ of model checking results of sample trajectories, computes the Bayes factor $B = \frac{P(\Delta|H_0)}{P(\Delta|H_1)}$ with respect to the prior knowledge, and then compares it against a fixed threshold $\lambda \geq 1$ : i) accepts $H_0$ if $B > \lambda$; ii) accepts $H_1$ if $B < 1/\lambda$. If both i) and ii) are not satisfied, the algorithm continues by drawing another sample.

**Error:** For any discrete random variable and prior, both probabilities of accepting $H_0$ when it is wrong (Type 2 error) and rejecting $H_0$ when it is correct (Type 1 error) are upper bounded by $1/\lambda$. Consequently, when the algorithm accepts $H_0$, $S \models P_{>0}(\Phi)$ is correct with probability $1 - 1/\lambda$.

**Example 4.1:** We use the algorithm outlined above to decide whether the gene network described in Example 2.1 works as an inverter. Since we do not consider the dynamics of the external regulators and our data only covers fixed values of the external regulators, we initialize the external regulator and model check the gene network against PBLTL formulas $P_{>0.95}(\Phi_{IL})$ and $P_{>0.95}(\Phi_{1H})$, where $\Phi_{IL}$ and $\Phi_{1H}$ are sub-formulas of formula $\Phi_I$ (2):

$$\Phi_{IL} = F^{360}G^{240}(x_{RFP} > 5200),$$
$$\Phi_{1H} = F^{360}G^{240}(x_{RFP} < 3200).$$

We define the hypothesis as $H_0 : S_{\pi^0} \models P_{>0.95}(\Phi_{IL})$, where $\pi^0$ is the distribution of initial states such that $x_{Ara} = 0 mM$ with probability 1, and the concentration of each protein is uniformly distributed over its domain. The hypothesis testing algorithm terminates after 18676 iterations, with 17763 satisfying trajectories by proving that $S_{\pi^0} \models P_{>0.95}(\Phi_{IL})$ holds with probability 0.99, ($\lambda = 100$). Next, we define the hypothesis as $H_0 : S_{\pi^0} \models P_{>0.95}(\Phi_{1H})$, where $\pi^0$ is the same as $\pi^0$ except that $x_{Ara} = 10 mM$ with probability 1. The hypothesis testing algorithm terminates after 87 iterations, with 75 satisfying trajectories by proving that the alternative hypothesis $H_1 : S_{\pi^0} \models P_{<0.95}(\Phi_{1H})$ holds with probability 0.99 ($\lambda = 100$).

**B. Bayesian Interval Estimation**

Interval estimation is used to find a probability range $\Theta$ for a well defined but unknown probability $\theta$ such that $\theta \in \Theta$ with arbitrarily high probability. We use the Bayesian Interval Estimation algorithm [15] to solve Problem 2.1-(ii). The half size $\delta \in (0, 1)$ of the desired interval estimate $\Theta$ and a coverage goal $c \in (\frac{1}{2}, 1)$, $c \leq P\text{Prob}(\theta \in \Theta)$, are the parameters of the algorithm. Similar to the Bayesian Hypothesis Testing algorithm, the algorithm sequentially draws a sample model checking result and updates the Bayesian estimate. The algorithm stops and outputs the current estimate $\hat{\theta}$ when the coverage goal is achieved, otherwise it continues by drawing another sample.

**Error:** For any discrete random variable and prior, the probability that $\theta \not\in \Theta = [\theta - \delta, \theta + \delta]$ is upper bounded by $\frac{1 - c^2}{(1 - \pi)^2}$, where $\pi$ is the prior probability that $\theta \in \Theta$.
Example 4.2: Consider the gene network from Example 2.1, and BLTL formulas $\Phi_{IL}(8)$ and $\Phi_{IH}(9)$, and the initial distributions $\pi^L_0$ and $\pi^H_0$ from Example 4.1. System $S_{\pi^L_0}$ satisfies $\Phi_{IL}$ with probability 0.958 and system $S_{\pi^H_0}$ satisfies $\Phi_{IH}$ with probability 0.843, which shows that the circuit works as an inverter with high probability. These probabilities are found by using the Bayesian Interval Estimation algorithm. A beta prior with $\alpha = \beta = 1$ is used and the algorithm parameters are set to $\delta = 0.01$ (half interval size), $c = 0.99$ (coverage goal), meaning that when the algorithm results in a probability estimate $\hat{\theta}$, then the unknown probability $\theta$ that the circuit satisfies the specification is in $[\hat{\theta} - 0.01, \hat{\theta} + 0.01]$ with probability $1 - \frac{(1-0.99)\times 0.02}{0.99 \times 0.98}$.

C. Parameter Optimization

The problems solved in the previous sections require to specify the formula fully, i.e. all the thresholds and time bounds must be set. Here we show that we can use the Bayesian Hypothesis Testing algorithm to solve Problem 2.2. Specifically, we propose to iteratively use the testing algorithm to minimize or maximize one of these parameters when the rest of them together with a probability bound $\theta$ are given. Algorithm 1 presents an example of such a search routine, where a threshold is minimized through a binary search.

Algorithm 1 Threshold Minimization

**Input:** A system $S$, a BLTL formula $\Phi$ with $x_o < T$ appearing in $\Phi$, a probability bound $\theta$, a precision bound $\tau$.

**Output:** $T$ such that $S \models P_{\geq \theta}(\Phi)$.

**Threshold Minimization**

1. $T = T^o_{max}$ and $T^H = x_o^o$.
2. $testT = \frac{T^L + T^H}{2}$.
3. while $T^H - T^L > \tau$ do
   1. $T^L = testT$.
   2. $T^H = testT$. 
   3. if $BHT(S, T^L, \theta)$ then $BHT(S, T^H, \theta)$
   4. else $T^L = testT$. 
   5. end if
   6. $testT = \frac{T^L + T^H}{2}$.
4. end while

Example 4.3: Consider the gene network from Example 2.1, and $\Phi_{IL}(8)$, $\Phi_{IH}(9)$, $\pi^L_0$ and $\pi^H_0$ from Example 4.1. The minimum output threshold $T^o_{RFP} = 3554$ for formula $\Phi_{IH}(9)$ and system $S^o_0$ is found by Algorithm 1 with $\tau = 10$, $\theta = 0.95$. Via a similar algorithm with $\tau = 10$, $\theta = 0.95$, the maximum output threshold for formula $\Phi_{IL}(8)$ and system $S^o_0$ is found as $T^o_{RFP} = 5219$.

These optimized thresholds can be used to compare different gene networks designed as inverters. Assume a set of gene networks $S_i, i = 1, \ldots, l$ and corresponding characterization data are given. High $T^o_{RFP}$ and low $T^L_{RFP}$ output thresholds can be found for each $S_i$ as explained above. Then, the gene network with the maximum threshold gap, i.e. $arg \max_i \ldots l T^o_{RFP} - T^L_{RFP}$, can be considered as the most robust design. Moreover, such optimized thresholds can further be used to couple the engineered cells with electronic systems.

V. Conclusion

We developed a computational framework that allows for statistical verification of a synthetic gene network given information on the decay rates of its proteins and fluorescent microscopy experimental data characterizing its promoters. The framework is based on (1) the construction of a mathematical model in the form of a discrete-time stochastic system with parameter distributions derived from the experimental data, and (2) statistical model checking over simulated trajectories of the model. We applied the proposed computational tool to verify the behavior of a synthetic gene circuit designed to behave as a logical inverter.

**References**


