

Control of Gene Regulatory Networks with Noisy Measurements and Uncertain Inputs

Mahdi Imani, *Student Member, IEEE* and Ulisses M. Braga-Neto, *Senior Member, IEEE*

Abstract—This paper is concerned with the problem of stochastic control of gene regulatory networks (GRNs) observed indirectly through noisy measurements and with uncertainty in the intervention inputs. The partial observability of the gene states and uncertainty in the intervention process are accounted for by modeling GRNs using the partially-observed Boolean dynamical system (POBDS) signal model with noisy gene expression measurements. Obtaining the optimal infinite-horizon control strategy for this problem is not attainable in general, and we apply reinforcement learning and Gaussian process techniques to find a near-optimal solution. The POBDS is first transformed to a directly-observed Markov Decision Process in a continuous belief space, and the Gaussian process is used for modeling the cost function over the belief and intervention spaces. Reinforcement learning then is used to learn the cost function from the available gene expression data. In addition, we employ sparsification, which enables the control of large partially-observed GRNs. The performance of the resulting algorithm is studied through a comprehensive set of numerical experiments using synthetic gene expression data generated from a melanoma gene regulatory network.

Index Terms—Infinite-Horizon Control, Gene Regulatory Networks, Partially-Observed Boolean Dynamical Systems, Reinforcement Learning, Gaussian Process.

I. INTRODUCTION

A key purpose of control of gene regulatory networks (GRNs) is to derive appropriate strategies to avoid undesirable states, such as those associated with disease. GRNs play a crucial role in every process of cellular life, including cell differentiation, metabolism, the cell cycle and signal transduction [1]. Several models have been introduced in the literature to mathematically capture the behavior of gene regulatory networks. These models include probabilistic Boolean network (PBN) [2], Bayesian networks [3], Boolean control networks [4] and ordinary differential equations (ODE) [5]–[8]. Several intervention strategies have been also developed for the control of GRNs; e.g. [9]–[11].

Most of the existing approaches assume that the Boolean state of genes is directly observable. In the current paper, the goal is to obtain appropriate intervention strategies to beneficially alter network dynamics, while assuming that the GRN is only indirectly observable through noisy gene expression measurements. In addition, we assume that the intervention input itself has uncertain effects. The signal model used in our approach is the partially-observable Boolean dynamical system (POBDS) model [12], [13]. Several tools for POBDSs have

been developed in recent years, such as the optimal filter and smoother based on the minimum mean square error (MMSE) criterion, called the Boolean Kalman Filter (BKF) [12] and Boolean Kalman Smoother (BKS) [14], respectively. In addition, particle filtering implementations of these filters, as well as schemes for handling correlated noise, simultaneous state and parameter estimation, and network inference for POBDSs were developed [13], [15]–[17].

In [18], [19], a state feedback controller for POBDSs is proposed based on optimal infinite horizon control of the Boolean state process, with the Boolean Kalman filter as state observer. This method, which is called V_BKF in this paper, has similarities to the Q_MDP method introduced in [20] for a general nonlinear state space model, which also does not employ the belief space when obtaining the control policy. Although this type of controller can be effective in some domains, the obtained policies do not take informed control action and might perform poorly in domains where repeated information gathering is necessary [21], [22]. In addition, the point-based value iteration method is used in [23] to control POBDSs with finite observation spaces. However, point-based techniques are only suitable for relatively small state spaces [24]–[26].

In this paper, we transform the partially-observed Boolean state space into belief space, which is a continuous observed state space, and learn the optimal policy in this space. We use the Gaussian process as a nonparametric technique to model the cost function over both belief and intervention spaces, and reinforcement learning is employed to learn the cost function by collecting a finite set of sample points. It should be noted that unlike parametric representation techniques in which the uncertainty of the cost function is encoded in the estimate of the parameters, nonparametric Gaussian processes are Bayesian representation of the cost function, which yields several benefits such as:

- 1) Prior knowledge about the cost in the belief and intervention spaces can be easily used to increase the learning rate.
- 2) The exploration/exploitation trade-off, which is a crucial fact in the performance of any reinforcement learning technique, can be easily addressed using the notion of uncertainty that is provided by Gaussian process model.
- 3) The concept of risk can be taken into account in obtaining a robust intervention strategy.

The above benefits will be discussed in detail throughout the text.

The article is organized as follows. In Section II, the POBDS model used in this paper is introduced. Then, the

M. Imani and U.M. Braga-Neto are with the Department of Electrical and Computer Engineering, Texas A&M University, College Station, TX 77843 USA (e-mail: m.imani88@tamu.edu, ulisses@ece.tamu.edu)

infinite-horizon control problem is formulated in Section III. In Section IV, reinforcement learning and Gaussian processes are used for control of partially-observed GRNs. The sparsification technique for control of large GRNs is discussed in Section V. Results of a numerical experiment using a melanoma gene regulatory network observed through synthetic gene expression time series are reported and discussed in Section VI. Finally, Section VII contains concluding remarks.

II. POBDS MODEL

In this section, the POBDS model is briefly introduced. It consists of a state model that describes the evolution of the Boolean dynamical system, which includes the system input, and an observation model that relates the state to the system output (measurements). More details can be found in [12], [13].

A. POBDS State Model

Assume that the system is described by a *state process* $\{\mathbf{X}_k; k = 0, 1, \dots\}$, where $\mathbf{X}_k \in \{0, 1\}^d$ represents the activation/inactivation state of the genes at time k . The state of the genes is affected by a sequence of *control inputs* $\{\mathbf{u}_k; k = 0, 1, \dots\}$, where $\mathbf{u}_k \in \mathbb{U} = \{0, 1\}^r$, $r \leq d$, represents a purposeful control input. The states are assumed to be updated at each discrete time through the following nonlinear signal model:

$$\mathbf{X}_k = \mathbf{f}(\mathbf{X}_{k-1}, \mathbf{u}_{k-1} \oplus \mathbf{v}_{k-1}) \oplus \mathbf{n}_k, \quad (1)$$

for $k = 1, 2, \dots$, where $\mathbf{f} : \{0, 1\}^d \times \{0, 1\}^r \rightarrow \{0, 1\}^d$ is a Boolean function called the *network function*, “ \oplus ” indicates componentwise modulo-2 addition, $\mathbf{n}_k \in \{0, 1\}^d$ is Boolean transition noise, and \mathbf{v}_k is Boolean noise that makes the control input uncertain. The noise processes $\{\mathbf{n}_k; k = 1, 2, \dots\}$ and $\{\mathbf{v}_k; k = 0, 1, \dots\}$ are assumed to be “white” in the sense that the noise at distinct time points are independent random variables. We also assume that noise processes are independent of each other and independent of the initial state \mathbf{X}_0 . The way that the input influences state evolution is part of the function \mathbf{f} ; typically, as will be the case here, each bit in the input \mathbf{u}_{k-1} , if it is one, flips the value of a specified bit of the Boolean state \mathbf{X}_k . Note that, in some cases, an input bit will not have any effect, since it may be reset by the corresponding bit in the noise \mathbf{v}_{k-1} .

We assume a noise distribution where the bits in \mathbf{n}_k and \mathbf{v}_k are i.i.d. (the general non-i.i.d. case can be similarly handled, at the expense of introducing more parameters), with $P(\mathbf{n}_k(i) = 1) = p$ and $P(\mathbf{v}_k(j) = 1) = q$, for $i = 1, \dots, d$, $j = 1, \dots, r$. Parameters $0 < p, q < 1/2$ correspond to the amount of “perturbation” to the Boolean state and intervention processes, respectively — the cases $p = 1/2$ and $q = 1/2$ correspond to maximum uncertainty.

Let $(\mathbf{x}^1, \dots, \mathbf{x}^{2^d})$ and $(\mathbf{v}^1, \dots, \mathbf{v}^{2^r})$ be arbitrary enumeration of the possible state and intervention noise vectors. The

prediction matrix is the transition matrix of the underlying controlled Markov chain, given by:

$$\begin{aligned} (M_k(\mathbf{u}))_{ij} &= P(\mathbf{X}_k = \mathbf{x}^i \mid \mathbf{X}_{k-1} = \mathbf{x}^j, \mathbf{u}_{k-1} = \mathbf{u}) \\ &= \sum_{s=1}^{2^r} P(\mathbf{X}_k = \mathbf{x}^i \mid \mathbf{X}_{k-1} = \mathbf{x}^j, \mathbf{u}_{k-1} = \mathbf{u}, \mathbf{v}_{k-1} = \mathbf{v}^s) \\ &\quad \times P(\mathbf{v}_{k-1} = \mathbf{v}^s) \\ &= \sum_{s=1}^{2^r} q^{||\mathbf{u} \oplus \mathbf{v}^s||_1} (1-q)^{r-||\mathbf{u} \oplus \mathbf{v}^s||_1} p^{||\mathbf{f}(\mathbf{x}^j, \mathbf{u} \oplus \mathbf{v}^s) \oplus \mathbf{x}^i||_1} \\ &\quad \times (1-p)^{d-||\mathbf{f}(\mathbf{x}^j, \mathbf{u} \oplus \mathbf{v}^s) \oplus \mathbf{x}^i||_1}, \end{aligned} \quad (2)$$

for $i, j = 1, \dots, 2^d$ and given $\mathbf{u} \in \mathbb{U}$.

B. POBDS Observation Model

In this paper, we assume a POBDS observation model that corresponds to Gaussian gene expression measurements at each time point. This is an appropriate model for many important gene-expression measurement technologies, such as cDNA microarrays [27] and live cell imaging-based assays [28], in which gene expression measurements are continuous and unimodal (within a single population of interest).

Let $\mathbf{Y}_k = (\mathbf{Y}_k(1), \dots, \mathbf{Y}_k(d))$ be a vector containing the measurements at time k , for $k = 1, 2, \dots$. The component $\mathbf{Y}_k(j) \in \mathbb{R}$ is the abundance measurement corresponding to transcript j , for $j = 1, \dots, d$. We assume conditional independency of the measurements given the state as:

$$\begin{aligned} P(\mathbf{Y}_k = \mathbf{y} \mid \mathbf{X}_k = \mathbf{x}) \\ &= \prod_{j=1}^d P(\mathbf{Y}_k(j) = \mathbf{y}(j) \mid \mathbf{X}_k(j) = \mathbf{x}(j)), \end{aligned} \quad (3)$$

and adopt a Gaussian model,

$$\begin{aligned} P(\mathbf{Y}_k(j) = \mathbf{y}(j) \mid \mathbf{X}_k(j) = \mathbf{x}(j)) \\ &= \frac{1}{\sqrt{2\pi\sigma_j^2}} \exp\left(-\frac{(\mathbf{y}(j) - \mu_j)^2}{2\sigma_j^2}\right), \end{aligned} \quad (4)$$

where μ_j and $\sigma_j > 0$ are the mean and standard deviation of the abundance of transcript j , respectively, for $j = 1, \dots, d$.

According to the Boolean state model, there are two possible states for the abundance of transcript j : high, if $\mathbf{x}(j) = 1$, and low, if $\mathbf{x}(j) = 0$. Accordingly, we model μ_j and σ_j as:

$$\begin{aligned} \mu_j &= \mu_j^0 (1 - \mathbf{x}(j)) + \mu_j^1 \mathbf{x}(j), \\ \sigma_j &= \sigma_j^0 (1 - \mathbf{x}(j)) + \sigma_j^1 \mathbf{x}(j), \end{aligned} \quad (5)$$

where the parameters $(\mu_j^0, \sigma_j^0 > 0)$ and $(\mu_j^1, \sigma_j^1 > 0)$ specify the means and standard deviations of the abundance of transcript j in the inactivated and activated states, respectively.

Based on equations (3), (4) and (5), the *update matrix*, which is a diagonal matrix of size $2^d \times 2^d$, is given by:

$$\begin{aligned} (T_k(\mathbf{y}))_{ii} &= P(\mathbf{Y}_k = \mathbf{y} \mid \mathbf{X}_k = \mathbf{x}^i) \\ &= \left(\prod_{j=1}^d \frac{1}{\sqrt{2\pi(\sigma_j^0(1 - \mathbf{x}^i(j)) + \sigma_j^1 \mathbf{x}^i(j))^2}} \right) \\ &\quad \times \exp\left(-\sum_{j=1}^d \frac{(\mathbf{y}(j) - \mu_j^0(1 - \mathbf{x}^i(j)) - \mu_j^1 \mathbf{x}^i(j))^2}{2(\sigma_j^0(1 - \mathbf{x}^i(j)) + \sigma_j^1 \mathbf{x}^i(j))^2}\right), \end{aligned} \quad (6)$$

for $i = 1, \dots, 2^d$ and observed $\mathbf{y} \in R^d$. Typical values for the parameters are given in Section VI when we discuss the numerical experiments performed to evaluate the proposed approach.

III. INFINITE-HORIZON CONTROL

In this section, the infinite-horizon control problem for the POBDS model is formulated. The goal of control in this paper is to select the appropriate external input $\mathbf{u}_k \in \mathbb{U}$ at each time k to make the network spend the least amount of time, on average, in undesirable states (e.g., states corresponding to cell proliferation, which are undesirable, as they may be associated with cancer [29]).

For the infinite-horizon control problem, we assume that the system prediction matrix $M_k(\mathbf{u})$ and update matrix $T_k(\mathbf{y})$ can only depend on time through the control input $\mathbf{u} \in \mathbb{U}$ and measurement $\mathbf{y} \in R^d$, respectively. We will thus drop the index k and write simply $M(\mathbf{u})$ and $T(\mathbf{y})$.

Since the state of the system is not observed directly, all available for decision making at each time step are the observations up to current time $\mathbf{y}_{1:k} = (\mathbf{y}_1, \dots, \mathbf{y}_k)$, and the control input applied to the system up to previous time step $\mathbf{u}_{0:k-1} = (\mathbf{u}_1, \dots, \mathbf{u}_{k-1})$. Rather than storing the history of observations and control inputs, we record the probability of states given that information at each time step. This probability distribution is known as the *belief state* at time k , given by:

$$\mathbf{b}_k(i) = P(\mathbf{X}_k = \mathbf{x}^i \mid \mathbf{y}_{1:k}, \mathbf{u}_{0:k-1}), \quad (7)$$

for $i = 1, \dots, 2^d$. The initial belief state is simply the initial state distribution, $\mathbf{b}_0(i) = P(\mathbf{X}_0 = \mathbf{x}^i)$, for $i = 1, \dots, 2^d$. Since $0 \leq \mathbf{b}(i) \leq 1$ and $\sum_{i=1}^{2^d} \mathbf{b}(i) = 1$, a belief vector \mathbf{b}_k is a point in a $(2^d - 1)$ -dimensional simplex \mathbb{B} , called the *belief space*.

Assuming \mathbf{b} is the current belief state of the system, if the control input \mathbf{u} is applied and observation \mathbf{y} is made, the new belief can be obtained by using Bayes' rule as:

$$\mathbf{b}^{\mathbf{u}, \mathbf{y}} = \frac{T(\mathbf{y}) M(\mathbf{u}) \mathbf{b}}{\|T(\mathbf{y}) M(\mathbf{u}) \mathbf{b}\|_1}, \quad (8)$$

where $\|\cdot\|_1$ denotes the L_1 -norm of a vector. Thus, by using the concept of belief state, a POBDS can be transformed into a Markov decision process (MDP) with a state transition probability in the belief space \mathbb{B} , given by:

$$p(\mathbf{b}' \mid \mathbf{b}, \mathbf{u}) = \int_{\mathbf{y} \in R^d} \|T(\mathbf{y}) M(\mathbf{u}) \mathbf{b}\|_1 I_{\mathbf{b}' = \mathbf{b}^{\mathbf{u}, \mathbf{y}}} d\mathbf{y}, \quad (9)$$

where $I_{\mathbf{b}' = \mathbf{b}^{\mathbf{u}, \mathbf{y}}}$ is an indicator function which returns 1 if $\mathbf{b}' = \mathbf{b}^{\mathbf{u}, \mathbf{y}}$ and 0 otherwise.

Now, let $c(\mathbf{x}^i, \mathbf{u})$ be a bounded cost of control for state \mathbf{x}^i and control input \mathbf{u} , for $i = 1, \dots, 2^d$ and $\mathbf{u} \in \mathbb{U}$. The cost can be transformed to belief space as follows:

$$g(\mathbf{b}, \mathbf{u}) = \sum_{i=1}^{2^d} c(\mathbf{x}^i, \mathbf{u}) \mathbf{b}(i). \quad (10)$$

The goal of infinite-horizon control is to minimize the following cost function by choosing the appropriate control input at each time step:

$$J_\infty = E \left[\sum_{k=1}^{\infty} \gamma^k g(\mathbf{b}_k, \mathbf{u}_k) \mid \mathbf{b}_0 \right], \quad (11)$$

where \mathbf{b}_0 is the known initial belief state, and the discount factor γ places a premium on minimizing the costs of early interventions as opposed to later ones, which is sensible from a medical perspective [10]. The classical results proved in [30] for MDPs can be used here. For an infinite-horizon control problem with discount factor γ , the Bellman operator for the belief space \mathbb{B} can be written as follows:

$$\begin{aligned} T[\mathbf{J}](\mathbf{b}) &= \min_{\mathbf{u} \in \mathbb{U}} \left[g(\mathbf{b}, \mathbf{u}) + \gamma \int_{\mathbf{b}' \in \mathbb{B}} p(\mathbf{b}' \mid \mathbf{b}, \mathbf{u}) \mathbf{J}(\mathbf{b}') \right] \\ &= \min_{\mathbf{u} \in \mathbb{U}} \left[g(\mathbf{b}, \mathbf{u}) + \gamma \int_{\mathbf{y} \in R^d} \|T(\mathbf{y}) M(\mathbf{u}) \mathbf{b}\|_1 \mathbf{J}(\mathbf{b}^{\mathbf{u}, \mathbf{y}}) d\mathbf{y} \right]. \end{aligned} \quad (12)$$

Equation (12) denotes the Bellman operator required for performing the dynamic programming technique for partially-observed GRNs. However, since the belief \mathbf{b} is in the $(2^d - 1)$ -dimensional simplex \mathbb{B} , computing the Bellman operator in (12) for all belief points is not possible. It should be noted that in the case of directly observed GRNs, the optimal stationary control policy can be obtained by finding the fixed-point solution of the Bellman operator in a finite state space without the challenges presented by a continuous state space. For more information see [10], [18].

IV. CONTROL USING REINFORCEMENT LEARNING AND GAUSSIAN PROCESS

A. Q-function as a Gaussian Process

In this section, the cost function over belief and intervention spaces is modeled using a Gaussian process. A *policy* is a function $\pi : \mathbb{B} \rightarrow \mathbb{U}$, which associates a control input to each belief state. Given a policy π , the discounted return for time step k can be defined as:

$$C_k^\pi = \sum_{m=0}^{\infty} \gamma^m g(\mathbf{b}_{k+m+1}, \mathbf{u}_{k+m+1}), \quad (13)$$

where C_k^π is the total accumulated cost obtained over time following policy π . Note that C_k^π can be written in a recursive fashion as:

$$C_k^\pi = g(\mathbf{b}_{k+1}, \mathbf{u}_{k+1}) + \gamma C_{k+1}^\pi. \quad (14)$$

Due to the stochasticity in belief transition, which arises from stochasticity of state, observation and intervention processes, the discounted return is a random variable which can be decomposed into a mean $Q^\pi(\mathbf{b}, \mathbf{u})$ and a residual $\Delta Q^\pi(\mathbf{b}, \mathbf{u})$, for $\mathbf{u} \in \mathbb{U}$, as:

$$C_k^\pi(\mathbf{b}_k = \mathbf{b}, \mathbf{u}_k = \mathbf{u}) = Q^\pi(\mathbf{b}, \mathbf{u}) + \Delta Q^\pi(\mathbf{b}, \mathbf{u}), \quad (15)$$

where

$$Q^\pi(\mathbf{b}, \mathbf{u}) = E_\pi[C_k^\pi \mid \mathbf{b}_k = \mathbf{b}, \mathbf{u}_k = \mathbf{u}], \quad (16)$$

where the expectation is taken over all possible successor belief state sequences that can be observed. Notice that the mean and residual of the return are assumed to be independent of k .

Replacing (15) into (14), the immediate cost can be written as:

$$g(\mathbf{b}_{k+1}, \mathbf{u}_{k+1}) = Q^\pi(\mathbf{b}_k, \mathbf{u}_k) - \gamma Q(\mathbf{b}_{k+1}, \mathbf{u}_{k+1}) + \Delta Q^\pi(\mathbf{b}_k, \mathbf{u}_k) - \gamma \Delta Q^\pi(\mathbf{b}_{k+1}, \mathbf{u}_{k+1}). \quad (17)$$

Let $\mathbf{B}_k = [(\mathbf{b}^0, \mathbf{u}^0), \dots, (\mathbf{b}^k, \mathbf{u}^k)]^T$ be the sequence of observed belief states and taken interventions between time steps 0 and k , under policy π , we have:

$$\begin{aligned} g(\mathbf{b}_1, \mathbf{u}_1) &= Q^\pi(\mathbf{b}_0, \mathbf{u}_0) - \gamma Q^\pi(\mathbf{b}_1, \mathbf{u}_1) \\ &\quad + \Delta Q^\pi(\mathbf{b}_0, \mathbf{u}_0) - \gamma \Delta Q^\pi(\mathbf{b}_1, \mathbf{u}_1), \\ g(\mathbf{b}_2, \mathbf{u}_2) &= Q^\pi(\mathbf{b}_1, \mathbf{u}_1) - \gamma Q^\pi(\mathbf{b}_2, \mathbf{u}_2) \\ &\quad + \Delta Q^\pi(\mathbf{b}_1, \mathbf{u}_1) - \gamma \Delta Q^\pi(\mathbf{b}_2, \mathbf{u}_2), \\ &\vdots \\ g(\mathbf{b}_k, \mathbf{u}_k) &= Q^\pi(\mathbf{b}_{k-1}, \mathbf{u}_{k-1}) - \gamma Q^\pi(\mathbf{b}_k, \mathbf{u}_k) \\ &\quad + \Delta Q^\pi(\mathbf{b}_{k-1}, \mathbf{u}_{k-1}) - \gamma \Delta Q^\pi(\mathbf{b}_k, \mathbf{u}_k). \end{aligned} \quad (18)$$

The above equation can be written in a more compact form as [31]:

$$\mathbf{c}_k = \mathbf{H}_k \mathbf{Q}_k^\pi + \mathbf{H}_k \Delta \mathbf{Q}_k^\pi, \quad (19)$$

where

$$\begin{aligned} \mathbf{c}_k &= [g(\mathbf{b}_1, \mathbf{u}_1), \dots, g(\mathbf{b}_k, \mathbf{u}_k)]^T, \\ \mathbf{Q}_k^\pi &= [Q^\pi(\mathbf{b}_0, \mathbf{u}_0), \dots, Q^\pi(\mathbf{b}_k, \mathbf{u}_k)]^T, \\ \Delta \mathbf{Q}_k^\pi &= [\Delta Q^\pi(\mathbf{b}_0, \mathbf{u}_0), \dots, \Delta Q^\pi(\mathbf{b}_k, \mathbf{u}_k)]^T, \\ \mathbf{H}_k &= \begin{bmatrix} 1 & -\gamma & \dots & 0 & 0 \\ 0 & 1 & \dots & 0 & 0 \\ \vdots & & & & \vdots \\ 0 & 0 & \dots & 1 & -\gamma \end{bmatrix}. \end{aligned} \quad (20)$$

Due to the changes in the policy π during the learning process, which will be addressed later in this section, $Q^\pi(\mathbf{b}, \mathbf{u})$ is a random variable. In order to specify a complete probabilistic generative model connecting Q-function and costs, one needs to define a prior distribution for the Q-function and the distribution of $\Delta \mathbf{Q}$. A Gaussian process is a stochastic process which allows the extension of multivariate Gaussians to infinite-sized collections of real valued variables [32]. In this paper, we use Gaussian processes for non-parametric Bayesian representation of our cost function. The prior distribution of the Q-function is defined as:

$$Q^\pi(\mathbf{b}, \mathbf{u}) = \mathcal{GP}(0, k((\mathbf{b}, \mathbf{u}), (\mathbf{b}, \mathbf{u}))), \quad (21)$$

where $k(\cdot, \cdot)$ is a real-valued kernel function over both belief and intervention spaces. In addition, we assume the residual $\Delta \mathbf{Q}$ is generated independently from a zero mean Gaussian distribution as $\Delta Q^\pi(\mathbf{b}, \mathbf{u}) \sim \mathcal{N}(0, \sigma_q^2)$, where the variance σ_q^2 is to be determined.

The kernel function $k(\cdot, \cdot)$ encodes our prior beliefs on correlations between different points in belief and intervention

spaces. We consider kernels that decompose over the belief state and intervention space as:

$$k((\mathbf{b}, \mathbf{u}), (\mathbf{b}', \mathbf{u}')) = k_B(\mathbf{b}, \mathbf{b}') k_U(\mathbf{u}, \mathbf{u}'). \quad (22)$$

We employ the direct probabilistic representation of our intervention process in defining the kernel function in the intervention space as:

$$k_U(\mathbf{u}, \mathbf{u}') = q^{||\mathbf{u} \oplus \mathbf{u}'||_1} (1 - q)^{r - ||\mathbf{u} \oplus \mathbf{u}'||_1}. \quad (23)$$

Notice that taking control input \mathbf{u} consecutively affects the cost function associated to all different $\mathbf{u}' \in \mathbb{U}$, for any $0 < q \leq 0.5$ where q is the intensity of Bernoulli intervention process.

For the belief state kernel, we consider the well-known exponential kernel function which has several features such as being infinitely differentiable, which means that the GP with this covariance function has mean square derivatives of all orders, and contains a small number of parameters [32]. This kernel function can be represented as:

$$k_B(\mathbf{b}, \mathbf{b}') = \sigma_f^2 \exp\left(-\frac{||\mathbf{b} - \mathbf{b}'||^2}{2l^2}\right), \quad (24)$$

where σ_f^2 determines the prior variance and l denotes the correlation at different belief points (the large values of l model more correlation of Q-function in the belief space). The parameters σ_f^2 and l are to be determined.

It is worth mentioning that factorization in equation (22) depends on the fact that the multiplication of two separate kernels results in another kernel [33].

Using the above assumptions, the posterior distribution of $Q^\pi(\mathbf{b}, \mathbf{u})$ in equation (19) can be obtained as [32]:

$$Q^\pi(\mathbf{b}, \mathbf{u}) | \mathbf{c}_k, \mathbf{B}_k \sim \mathcal{N}(\bar{Q}(\mathbf{b}, \mathbf{u}), \text{cov}((\mathbf{b}, \mathbf{u}), (\mathbf{b}, \mathbf{u}))), \quad (25)$$

where

$$\begin{aligned} \bar{Q}(\mathbf{b}, \mathbf{u}) &= \mathbf{k}_k(\mathbf{b}, \mathbf{u})^T \mathbf{H}_k^T (\mathbf{H}_k \mathbf{K}_k \mathbf{H}_k^T + \sigma_q^2 \mathbf{H}_k \mathbf{H}_k^T)^{-1} \mathbf{c}_k, \\ \text{cov}((\mathbf{b}, \mathbf{u}), (\mathbf{b}, \mathbf{u})) &= k((\mathbf{b}, \mathbf{u}), (\mathbf{b}, \mathbf{u})) - \mathbf{k}_k(\mathbf{b}, \mathbf{u})^T \mathbf{H}_k^T \\ &\quad (\mathbf{H}_k \mathbf{K}_k \mathbf{H}_k^T + \sigma_q^2 \mathbf{H}_k \mathbf{H}_k^T)^{-1} \mathbf{H}_k \mathbf{k}_k(\mathbf{b}, \mathbf{u}), \end{aligned} \quad (26)$$

with

$$\begin{aligned} \mathbf{k}_k(\mathbf{b}, \mathbf{u}) &= [k((\mathbf{b}_0, \mathbf{u}_0), (\mathbf{b}, \mathbf{u})), \dots, k((\mathbf{b}_k, \mathbf{u}_k), (\mathbf{b}, \mathbf{u}))]^T, \\ \mathbf{K}(\mathbf{b}, \mathbf{u}) &= [\mathbf{k}_k(\mathbf{b}_0, \mathbf{u}_0), \dots, \mathbf{k}_k(\mathbf{b}_k, \mathbf{u}_k)]^T. \end{aligned} \quad (27)$$

Using the above formulation, the Q-function before observing any data is a zero-mean Gaussian process with covariance $k((\mathbf{b}, \mathbf{u}), (\mathbf{b}, \mathbf{u}))$, while at time step k , this posterior can be obtained based on the sequence of costs \mathbf{c}_k and sequence of observed beliefs and interventions $\mathbf{B}_k = [(\mathbf{b}_0, \mathbf{u}_0), \dots, (\mathbf{b}_k, \mathbf{u}_k)]$ using equation (25). The uncertainty in the Q-function, which is modeled by the covariance function in equation (25), gets small as more measurements are acquired.

The parameters of the Gaussian process such as the variance σ_q and the kernel parameters σ_f and l , can be updated at each time point using maximum likelihood, given that the marginal likelihood of the observed cost has the following distribution:

$$\mathbf{c}_k | \mathbf{B}_k \sim \mathcal{N}(\mathbf{0}, \mathbf{H}_k (\mathbf{K}_k + \sigma_q^2 \mathbf{I}_k \mathbf{H}_k^T)), \quad (28)$$

where \mathbf{I}_k is the identity matrix of size $k \times k$. For more information, the reader is referred to [32].

B. Learning the Q -function using GP-SARSA

The Gaussian Process Temporal Difference (GPTD) approach [34] is a modification of the well-known temporal difference learning method, when the cost function over the whole belief space is modeled by the Gaussian process and the cost is learned based on samples of the discounted sums of returns. A SARSA (State-Action-Reward-State-Action) type-algorithm [35], called GP-SARSA, estimates the Q function using the GPTD method.

Defining appropriate exploration/exploitation strategies for data collection has a major effect on the performance of reinforcement learning techniques. The exploration/exploitation tradeoff specifies the balance between the need to explore the space of all possible policies, and the necessity to focus exploitation towards policies that yield lower cost. Several policies are introduced in literature such as ϵ -greedy and Boltzmann [35]. In this paper, the following policy is used for decision making [36]:

$$\pi(\mathbf{b}) = \operatorname{argmin}_{\mathbf{u} \in \mathbb{U}} \hat{Q}(\mathbf{b}, \mathbf{u}), \quad (29)$$

where $\hat{Q}(\mathbf{b}, \mathbf{u})$ is a sample from $\mathcal{N}(\bar{Q}(\mathbf{b}, \mathbf{u}), \operatorname{cov}((\mathbf{b}, \mathbf{u}), (\mathbf{b}, \mathbf{u})))$, for $\mathbf{u} \in \mathbb{U}$. Notice that the exploration and exploitation trade-off of this policy is fully-adaptive and no parameter should be tuned. The GP-SARSA algorithm for control of partially-observed GRNs is presented in Algorithm 1. Here $\mathbf{0}_{|\mathbb{V}|}$ denotes a vertical vector of the same size as vector \mathbf{v} with all elements equal to 0.

V. SPARSE APPROXIMATION OF GP-SARSA

The computational complexity of Algorithm 1 is of order $O(k^3)$ at the time of observing the k th measurement. The reason for this complexity is the need for computation of the inverse matrix in the posterior update of GP in equation (25). The growth of this computation over time can make the GP-SARSA algorithm computationally infeasible, especially for large POBDS, in which the need for more data for learning Q -function seems essential.

Several techniques have been developed to limit the size of the kernel during the learning process, such as kernel principal component analysis (KPCA) [37], novelty criterion (NC) [38] and the approximate linear dependence (ALD) method [34]. Here, we apply the ALD method, which constructs a dictionary of representative pairs of beliefs and interventions online, resulting from the approximate linear dependency condition in the feature space [34].

A kernel function can be interpreted as an inner product of a set of basis functions as:

$$k((\mathbf{b}, \mathbf{u}), (\mathbf{b}, \mathbf{u})) = \|\Phi(\mathbf{b}, \mathbf{u}) \bullet \Phi(\mathbf{b}, \mathbf{u})\|_1, \quad (30)$$

where \bullet denotes the dot product of two vectors and

$$\Phi(\mathbf{b}, \mathbf{u}) = [\phi_1(\mathbf{b}, \mathbf{u}), \phi_2(\mathbf{b}, \mathbf{u}), \dots]^T. \quad (31)$$

Algorithm 1 GP-SARSA: Control of POBDS

```

1: Initialization:  $\mathbf{c} \leftarrow []$ .
2: for each episode do
3:    $\mathbf{b} = \mathbf{b}_0$ .
4:   if first episode then
5:     Select  $\mathbf{u} \in \mathbb{U}$  randomly.
6:      $\mathbf{B} = (\mathbf{b}, \mathbf{u})$ ,  $\mathbf{K} \leftarrow k((\mathbf{b}, \mathbf{u}), (\mathbf{b}, \mathbf{u}))$ ,  $\mathbf{H} = [1 - \gamma]$ .
7:   else
8:     Choose  $\mathbf{u} \leftarrow \pi(\mathbf{b})$  (Eq. (29)).
9:   end if
10:  for each step in episode do
11:     $\mathbf{b}' = \frac{T(\mathbf{y})M(\mathbf{u})\mathbf{b}}{\|T(\mathbf{y})M(\mathbf{u})\mathbf{b}\|_1}$ ,  $c' \leftarrow g(\mathbf{b}', \mathbf{u})$ ,  $\mathbf{c} \leftarrow [\mathbf{c}, c']$ .
12:    if non-terminal step then
13:      Choose new control  $\mathbf{u}' \leftarrow \pi(\mathbf{b}')$  (Eq. (29)).
14:       $\mathbf{K} \leftarrow \begin{bmatrix} \mathbf{K} & \mathbf{k}(\mathbf{b}', \mathbf{u}') \\ \mathbf{k}(\mathbf{b}', \mathbf{u}')^T & k((\mathbf{b}', \mathbf{u}'), (\mathbf{b}', \mathbf{u}')) \end{bmatrix}$ .
15:       $\mathbf{B} \leftarrow [\mathbf{B}, (\mathbf{b}', \mathbf{u}')]$ ,  $\mathbf{H} = \begin{bmatrix} \mathbf{H} & \mathbf{0}_{|\mathbb{C}|-1} \\ \mathbf{0}_{|\mathbb{C}|-1}^T & 1 - \gamma \end{bmatrix}$ .
16:    else
17:       $\mathbf{H} = \begin{bmatrix} \mathbf{H} \\ \mathbf{0}_{|\mathbb{C}|-1}^T \end{bmatrix}$ .
18:    end if
19:    Update  $Q$ -function Posterior  $Q^\pi \mid \mathbf{c}, \mathbf{B}$  (Eq. (25)).
20:    if non-terminal step then
21:       $\mathbf{b} \leftarrow \mathbf{b}'$ ,  $\mathbf{u} \leftarrow \mathbf{u}'$ .
22:    end if
23:  end for
24: end for
```

Given a set of observed beliefs and inputs $\mathbf{B} = [(\mathbf{b}_0, \mathbf{u}_0), \dots, (\mathbf{b}_k, \mathbf{u}_k)]$, any linear combination of $\Phi(\mathbf{b}_0, \mathbf{u}_0), \dots, \Phi(\mathbf{b}_k, \mathbf{u}_k)$ is referred to as a *feature span*. The goal is to find the subset of points of minimum size that approximates this kernel span. This set is called *dictionary*, denoted by $D = [(\tilde{\mathbf{b}}_1, \tilde{\mathbf{u}}_1), \dots, (\tilde{\mathbf{b}}_m, \tilde{\mathbf{u}}_m)]$ where $D \in \mathbf{B}$.

The ALD condition for a new feature vector $\Phi(\mathbf{b}_k, \mathbf{u}_k)$ is:

$$\min_{\mathbf{t}_k} \left\| \sum_{i=1}^m t_{ki} \Phi(\tilde{\mathbf{b}}_i, \tilde{\mathbf{u}}_i) - \Phi(\mathbf{b}_k, \mathbf{u}_k) \right\|^2 \leq \nu \quad (32)$$

where $(\mathbf{b}_k, \mathbf{u}_k)$ is the current point, $\mathbf{t}_k = [t_{k1}, \dots, t_{km}]^T$ is the vector of coefficients, ν is the threshold to determine the approximation accuracy and sparsity level, and m is the size of the current dictionary, $D = [(\tilde{\mathbf{b}}_1, \tilde{\mathbf{u}}_1), \dots, (\tilde{\mathbf{b}}_m, \tilde{\mathbf{u}}_m)]$. It is shown in [39] that an equivalent minimization to that in (32) can be written as:

$$\min_{\mathbf{t}_k} \left(k((\mathbf{b}_k, \mathbf{u}_k), (\mathbf{b}_k, \mathbf{u}_k)) - \tilde{\mathbf{K}}_{k-1}(\mathbf{b}_k, \mathbf{u}_k)^T \mathbf{t}_k \right) \leq \nu, \quad (33)$$

where

$$\tilde{\mathbf{K}}_{k-1}(\mathbf{b}_k, \mathbf{u}_k) = [k((\mathbf{b}_k, \mathbf{u}_k), (\tilde{\mathbf{b}}_0, \tilde{\mathbf{u}}_0)), \dots, k((\mathbf{b}_k, \mathbf{u}_k), (\tilde{\mathbf{b}}_m, \tilde{\mathbf{u}}_m))]. \quad (34)$$

The closed-form solution for minimization of equation (32) is $\mathbf{t}_k = \tilde{\mathbf{K}}_{k-1}^{-1} \tilde{\mathbf{K}}_{k-1}(\mathbf{b}_k, \mathbf{u}_k)$ where $\tilde{\mathbf{K}}_{k-1}$ is the Gram matrix of the points in the current dictionary. If the threshold in equation (33) exceeds ν , then $(\mathbf{b}_k, \mathbf{u}_k)$ is added to the dictionary, otherwise the dictionary stays the same.

The exact Gram matrix can be represented by:

$$\mathbf{K}_k = \Phi_k^T \Phi_k, \quad (35)$$

where $\Phi_k = [\Phi(\mathbf{b}_0, \mathbf{u}_0), \dots, \Phi(\mathbf{b}_k, \mathbf{u}_k)]$. The feature functions are approximated as $\Phi(\mathbf{b}_i, \mathbf{u}_i) \approx \sum_{j=1}^m t_{ij} \Phi(\tilde{\mathbf{b}}_j, \tilde{\mathbf{u}}_j)$, for $i = 0, \dots, k$. Defining the coefficients in a single matrix as $\mathbf{T}_k = [\mathbf{t}_0, \dots, \mathbf{t}_k]^T$, we have:

$$\begin{aligned} \mathbf{K}_k &= \Phi_k^T \Phi_k \approx \mathbf{T}_k \tilde{\mathbf{K}}_k \mathbf{T}_k^T, \\ \mathbf{k}_k(\mathbf{b}_k, \mathbf{u}_k) &\approx \mathbf{T}_k \tilde{\mathbf{k}}_k(\mathbf{b}, \mathbf{u}). \end{aligned} \quad (36)$$

Using equation (36), equation (25) can be approximated as:

$$Q^\pi(\mathbf{b}, \mathbf{u}) \mid \mathbf{c}_k, \mathbf{B}_k \sim \mathcal{N} \left(\tilde{Q}(\mathbf{b}, \mathbf{u}), \widetilde{\text{cov}}((\mathbf{b}, \mathbf{u}), (\mathbf{b}, \mathbf{u})) \right), \quad (37)$$

where

$$\begin{aligned} \tilde{Q}(\mathbf{b}, \mathbf{u}) &= \tilde{\mathbf{k}}_k(\mathbf{b}, \mathbf{u})^T \tilde{\mathbf{H}}_k^T (\tilde{\mathbf{H}}_k \tilde{\mathbf{K}}_k \tilde{\mathbf{H}}_k^T + \sigma_q^2 \tilde{\mathbf{H}}_k \tilde{\mathbf{H}}_k^T)^{-1} \mathbf{c}_k, \\ \widetilde{\text{cov}}((\mathbf{b}, \mathbf{u}), (\mathbf{b}, \mathbf{u})) &= k((\mathbf{b}, \mathbf{u}), (\mathbf{b}, \mathbf{u})) - \tilde{\mathbf{k}}_k(\mathbf{b}, \mathbf{u})^T \tilde{\mathbf{H}}_k^T \\ &\quad (\tilde{\mathbf{H}}_k \tilde{\mathbf{K}}_k \tilde{\mathbf{H}}_k^T + \sigma_q^2 \tilde{\mathbf{H}}_k \tilde{\mathbf{H}}_k^T)^{-1} \tilde{\mathbf{H}}_k \tilde{\mathbf{k}}_k(\mathbf{b}, \mathbf{u}), \end{aligned} \quad (38)$$

where $\tilde{\mathbf{H}}_k = \mathbf{H}_k \mathbf{T}_k$. Using this sparsification approach allows observations to be processed sequentially and reduces the complexity of Algorithm 1 from $O(k^3)$ to $O(km^2)$ where m is usually much smaller than k in practice. The reader is referred to [31] for more details. The full process of sparsification of the GP-SARSA algorithm for learning the cost function of POBDS is presented in Algorithm 2.

VI. NUMERICAL EXPERIMENTS

In this section, we conduct numerical experiments using a Boolean gene regulatory network involved in metastatic melanoma [40]. The network contains 7 genes: WNT5A, pirin, S100P, RET1, MART1, HADHB and STC2. The regulatory relationship for this network is shown in Fig. 1 and Boolean function is presented in Table I. The i th output binary string specifies the output value for i th input gene(s) in binary representation. For example, the last row of Table I specifies the value of STC2 at the current time step k from different pairs of (pirin, STC2) values at the previous time step $k-1$:

$$\begin{aligned} (\text{pirin}=0, \text{STC2}=0)_{k-1} &\rightarrow \text{STC2}_k=1 \\ (\text{pirin}=0, \text{STC2}=1)_{k-1} &\rightarrow \text{STC2}_k=1 \\ (\text{pirin}=1, \text{STC2}=0)_{k-1} &\rightarrow \text{STC2}_k=0 \\ (\text{pirin}=1, \text{STC2}=1)_{k-1} &\rightarrow \text{STC2}_k=1 \end{aligned}$$

In the study conducted in [41], the expression of WNT5A was found to be a highly discriminating difference between cells with properties typically associated with high metastatic competence versus those with low metastatic competence. Furthermore, the result of the study presented in [42] suggests to reduce the activation of WNT5A indirectly through control of other genes' activities. The reason is that an intervention that blocked the WNT5A protein from activating its receptor, could substantially reduce WNT5A's ability to induce a metastatic phenotype. For more information about the biological rationale for this, the reader is referred to [40].

In our experiments, the intervention is applied to either RET1 or HADHB. Recall that the intervention has uncertainty that is modeled by a Bernoulli distribution with parameter

Algorithm 2 SGP-SARSA: Sparse approximation of GP-SARSA for control of POBDS

```

1: Initialize:  $\alpha \leftarrow []$ ,  $\tilde{\mathbf{R}} \leftarrow []$ ,  $\tilde{\mathbf{r}} \leftarrow []$ ,  $s \leftarrow 0$ ,  $\frac{1}{\nu} \leftarrow 0$ .
2: for each episode do
3:   if first episode then
4:     Select  $\mathbf{u} \in \mathbb{U}$  randomly.
5:      $\mathbf{D} = \{(\mathbf{b}, \mathbf{u})\}$ ,  $\tilde{\mathbf{K}} = 1/k((\mathbf{b}, \mathbf{u}), (\mathbf{b}, \mathbf{u}))$ .
6:   else
7:     Choose  $\mathbf{u} \leftarrow \pi(\mathbf{b})$  (Eq. (29)).
8:   end if
9:    $\tilde{\mathbf{r}} = \mathbf{0}$ ,  $s \leftarrow 0$ ,  $\frac{1}{\nu} \leftarrow 0$ .
10:   $\mathbf{t} \leftarrow \tilde{\mathbf{K}}^{-1} \tilde{\mathbf{k}}(\mathbf{b}, \mathbf{u})$ ,  $\delta \leftarrow k((\mathbf{b}, \mathbf{u}), (\mathbf{b}, \mathbf{u})) - \tilde{\mathbf{k}}(\mathbf{b}, \mathbf{u})^T \mathbf{t}$ .
11:  if  $\delta > \nu$  then
12:     $\mathbf{D} \leftarrow \{(\mathbf{b}, \mathbf{u})\} \cup \mathbf{D}$ ,  $\tilde{\mathbf{K}}^{-1} \leftarrow \frac{1}{\delta} \begin{bmatrix} \delta \tilde{\mathbf{K}}^{-1} + \mathbf{t} \mathbf{t}^T & -\mathbf{t} \\ -\mathbf{t}^T & 1 \end{bmatrix}$ .
13:     $\mathbf{t} \leftarrow [0^T, 1]^T$ ,  $\alpha \leftarrow \begin{bmatrix} \alpha \\ 0 \end{bmatrix}$ ,  $\tilde{\mathbf{R}} \leftarrow \begin{bmatrix} \tilde{\mathbf{R}} & \mathbf{0} \\ \mathbf{0}^T & 0 \end{bmatrix}$ ,  $\tilde{\mathbf{r}} \leftarrow \begin{bmatrix} \tilde{\mathbf{r}} \\ 0 \end{bmatrix}$ .
14:  end if
15:  for each step in episode do
16:     $\mathbf{b}' = \frac{T(\mathbf{y})M(\mathbf{u})\mathbf{b}}{\|T(\mathbf{y})M(\mathbf{u})\mathbf{b}\|_1}$ ,  $\mathbf{c}' \leftarrow g(\mathbf{b}', \mathbf{u})$ .
17:    if non-terminal step then
18:      Choose new control  $\mathbf{u}' \leftarrow \pi(\mathbf{b}')$  (Eq. (29)).
19:       $\mathbf{t}' \leftarrow \tilde{\mathbf{K}}^{-1} \tilde{\mathbf{k}}(\mathbf{b}, \mathbf{u})$ .
20:       $\delta \leftarrow k((\mathbf{b}', \mathbf{u}'), (\mathbf{b}', \mathbf{u}')) - \tilde{\mathbf{k}}(\mathbf{b}', \mathbf{u}')^T \mathbf{t}'$ .
21:       $\Delta \tilde{\mathbf{k}} \leftarrow \tilde{\mathbf{k}}(\mathbf{b}, \mathbf{u}) - \gamma \tilde{\mathbf{k}}(\mathbf{b}', \mathbf{u}')$ .
22:    else
23:       $\mathbf{t}' \leftarrow \mathbf{0}$ ,  $\delta \leftarrow 0$ ,  $\Delta \tilde{\mathbf{k}} \leftarrow \tilde{\mathbf{k}}(\mathbf{b}, \mathbf{u})$ .
24:    end if
25:     $s \leftarrow \frac{\gamma \sigma_q^2}{\nu} s + \mathbf{c}' - \Delta \tilde{\mathbf{k}}^T \alpha$ .
26:    if  $\delta > \nu$  then
27:       $\mathbf{D} \leftarrow \{(\mathbf{b}', \mathbf{u}')\} \cup \mathbf{D}$ .
28:       $\tilde{\mathbf{K}}^{-1} \leftarrow \frac{1}{\delta} \begin{bmatrix} \delta \tilde{\mathbf{K}}^{-1} + \mathbf{t} \mathbf{t}^T & -\mathbf{t} \\ -\mathbf{t}^T & 1 \end{bmatrix}$ .
29:       $\mathbf{t}' \leftarrow [0^T, 1]^T$ ,  $\mathbf{h} \leftarrow [\mathbf{t}^T, -\gamma]^T$ .
30:       $\Delta k_{kk} \leftarrow \mathbf{t}^T (\tilde{\mathbf{k}}(\mathbf{b}, \mathbf{u}) - 2\gamma \tilde{\mathbf{k}}(\mathbf{b}', \mathbf{u}'))$ 
         $+ \gamma^2 k((\mathbf{b}', \mathbf{u}'), (\mathbf{b}', \mathbf{u}'))$ .
31:       $\tilde{\mathbf{r}}' \leftarrow \frac{\gamma \sigma_q^2}{\nu} \tilde{\mathbf{r}} + \mathbf{h} - \begin{bmatrix} \tilde{\mathbf{R}} \Delta \tilde{\mathbf{k}} \\ 0 \end{bmatrix}$ .
32:       $\nu \leftarrow (1 + \gamma^2) \sigma_q^2 + \Delta k_{kk} - \Delta \tilde{\mathbf{k}}^T \tilde{\mathbf{R}} \Delta \tilde{\mathbf{k}} + \frac{2\gamma \sigma_q^2}{\nu} \tilde{\mathbf{r}} \Delta \tilde{\mathbf{k}}$ 
         $- \frac{\gamma^2 \sigma_q^4}{\nu}$ .
33:       $\alpha \leftarrow \begin{bmatrix} \alpha \\ 0 \end{bmatrix}$ ,  $\tilde{\mathbf{R}} \leftarrow \begin{bmatrix} \tilde{\mathbf{R}} & \mathbf{0} \\ \mathbf{0}^T & 0 \end{bmatrix}$ .
34:    else
35:       $\mathbf{h} \leftarrow \mathbf{t} - \gamma \mathbf{t}'$ ,  $\tilde{\mathbf{r}}' \leftarrow \frac{\gamma \sigma_q^2}{\nu} \tilde{\mathbf{r}} + \mathbf{h} - \tilde{\mathbf{R}} \Delta \tilde{\mathbf{k}}$ .
36:      if non-terminal step then
37:         $\nu \leftarrow (1 + \gamma^2) \sigma_q^2 + \Delta \tilde{\mathbf{k}}^T (\tilde{\mathbf{r}}' + \frac{\gamma \sigma_q^2}{\nu} \tilde{\mathbf{r}}) - \frac{\gamma^2 \sigma_q^4}{\nu}$ .
38:      else
39:         $\nu \leftarrow \sigma_q^2 + \Delta \tilde{\mathbf{k}}^T (\tilde{\mathbf{r}}' + \frac{\gamma \sigma_q^2}{\nu} \tilde{\mathbf{r}}) - \frac{\gamma^2 \sigma_q^4}{\nu}$ .
40:      end if
41:    end if
42:     $\alpha \leftarrow \alpha + \frac{\tilde{\mathbf{r}}}{\nu} s$ ,  $\tilde{\mathbf{R}} \leftarrow \tilde{\mathbf{R}} + \frac{1}{\nu} \tilde{\mathbf{r}} \tilde{\mathbf{r}}^T$ ,  $\tilde{\mathbf{r}} \leftarrow \tilde{\mathbf{r}}'$ ,  $\mathbf{t} \leftarrow \mathbf{t}'$ .
43:    if non-terminal step then
44:       $\mathbf{b} \leftarrow \mathbf{b}'$ ,  $\mathbf{u} \leftarrow \mathbf{u}'$ .
45:    end if
46:  end for
47: end for

```

q . The cost of control is assumed to be 1 for any taken intervention and 0 when there is no intervention. Since the

TABLE I: Boolean functions for the melanoma Boolean network.

Genes	Input Gene(s)	Output
WNT5A	HADHB	10
pirin	prin, RET1,HADHB	00010111
S100P	S100P,RET1,STC2	10101010
RET1	RET1,HADHB,STC2	00001111
MART1	pirin,MART1,STC2	10101111
HADHB	pirin,S100P,RET1	01110111
STC2	pirin,STC2	1101

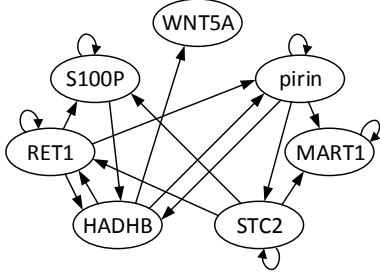


Fig. 1: Melanoma Gene Regulatory Network.

goal of control is preventing WNT5A gene to be upregulated, the cost function can be defined as follows:

$$c(\mathbf{x}^j, \mathbf{u}) = \begin{cases} 5 + \|\mathbf{u}\|_1 & \text{if WNT5A is 1 for state } j, \\ \|\mathbf{u}\|_1 & \text{if WNT5A is 0 for state } j. \end{cases} \quad (39)$$

Table II displays the parameter values used in the experiments. The reported results are taken over 10 different runs of system during execution each with time series of length 1000.

A. Effect of GP Parameters on the Performance of the SGP-SARSA Algorithm

In the experiments of Sections VI-A–VI-C, RET1 is used as the control gene and the parameters are set as follows: $p = 0.01$, $q = 0.01$, $\nu = 0.1$, $N_{ep} = 10$, $T = 1000$, $\sigma_j^0 = \sigma_j^1 = 10$, $l = 0.1$, $\nu = 0.1$. Fig. 2 displays the average cost of the system under control of SGP-SARSA for different correlation parameters l . The horizontal axis shows the number of training points used in the learning process before starting execution. It is clear that $l = 0.1$ has the lowest cost for different number of training points. In addition, $l = 0.2$ has similar cost as $l = 0.1$, while $l = 0.01$ behaves poorly for small number of training points and converges to the others as the number of training points increases. Overall, we conclude that the correlation coefficient does not greatly affect the resulting policy, and it only influences the speed of learning.

B. Effect of Sparsification Parameter on the Performance of the SGP-SARSA Algorithm

Fig. 3 displays the effect of the sparsification parameter ν on the performance of control. The right plot shows the increase in the average number of points kept in dictionary as parameter ν gets smaller. The effect of large dictionary size

TABLE II: Parameter values for numerical experiments.

Parameter	Value
Number of genes d	7
Number of episodes N_{ep}	1, 5, 10, 15, 20
Number of steps T	1000
Transition noise intensity p	0.01, 0.05
Scaling variance σ_j^2	5
Correlation parameter l	0.01, 0.1, 0.2
Noise residual σ_q	1
Intervention noise intensity q	0.01, 0.1, 0.2, 0.3, 0.4, 0.5
Initial belief $\mathbf{b}_0(i)$, $i = 1, \dots, 128$	1/128
Mean in inactivated state μ_j^0 , $j = 1, \dots, 7$	40
Mean in activated state μ_j^1 , $j = 1, \dots, 7$	60
Standard deviations in inactivated state σ_j^0	10, 15
Standard deviations in activated state σ_j^1	10, 15
Discount factor γ	0.95
Control genes	RET1, HADHB
Cost function	Equation (39)
Sparsification threshold ν	0.1, 1
Value Iteration threshold β [18]	10^{-8}

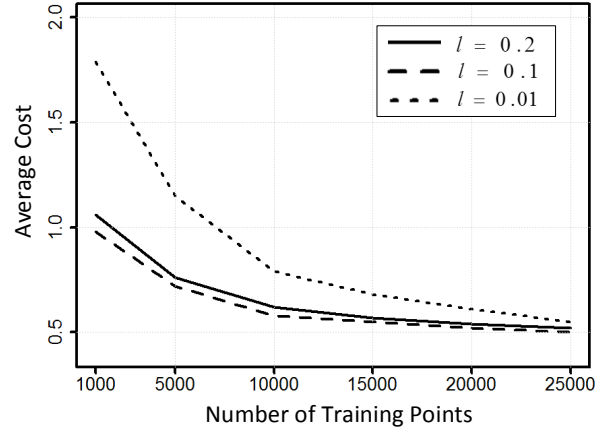


Fig. 2: Average cost per step achieved by SGP-SARSA as a function of the correlation coefficient.

can be clearly seen in the average cost presented in left plot in Fig. 3, in which lower cost is achieved on average for smaller ν for different training points.

C. Effect of Transition and Intervention Noise on the Performance of the SGP-SARSA Algorithm

Fig. 4 displays the performance of control for various process and intervention noise levels. It is clear that the average cost increases as the uncertainty in transition and intervention increases, as expected. For the system without control, the average cost is almost 2.65. By comparing this to the curves in Fig. 4, we reach the interesting conclusion

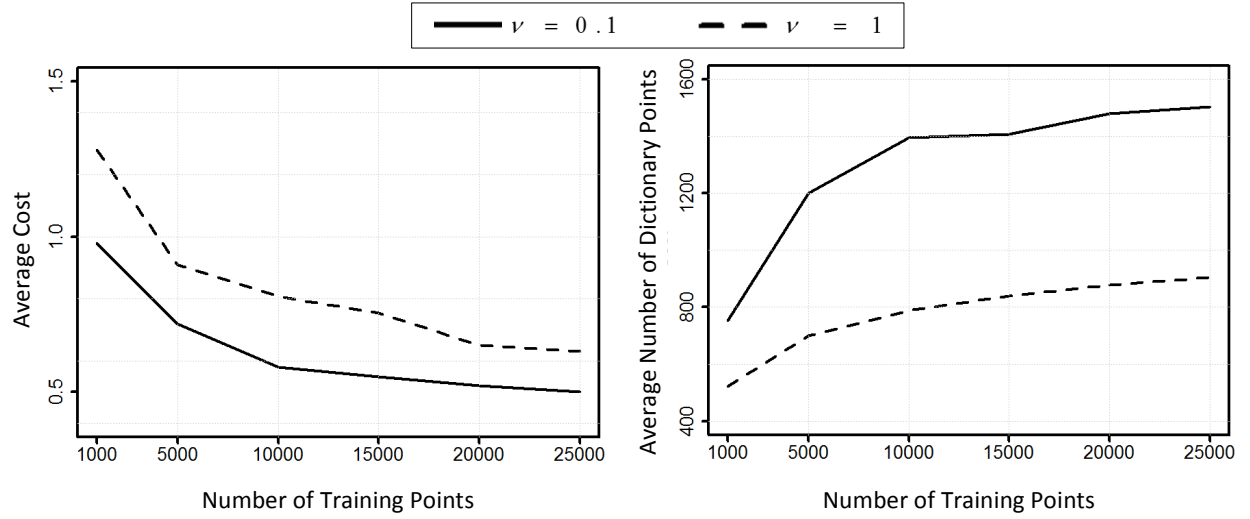


Fig. 3: Effect of the choice of sparsification parameter and dictionary size on the performance of SGP-SARSA.

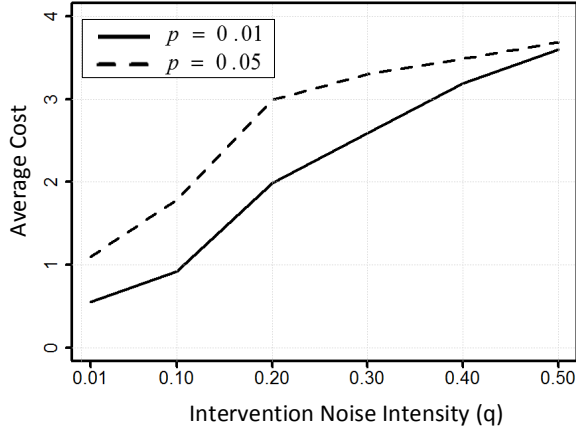


Fig. 4: Average cost per step achieved by SGP-SARSA as a function of the transition and intervention noise level.

that high uncertainty in the intervention process can make the situation worse than no control condition.

D. Distribution of Visited States for System under Control by SGP-SARSA Algorithm and without Control

Here we assess the probability mass over visited states for systems with and without control. Fig. 5 displays the long-run relative frequencies of visited states under the control policy obtained by SGP-SARSA and under no control. Desirable (inactive WNT5A) and undesirable (active WNT5A) states are indicated by black and red colors, respectively. We can observe that the control policy obtained by SGP-SARSA is able to shift the probability mass of visited states from undesirable to desirable states.

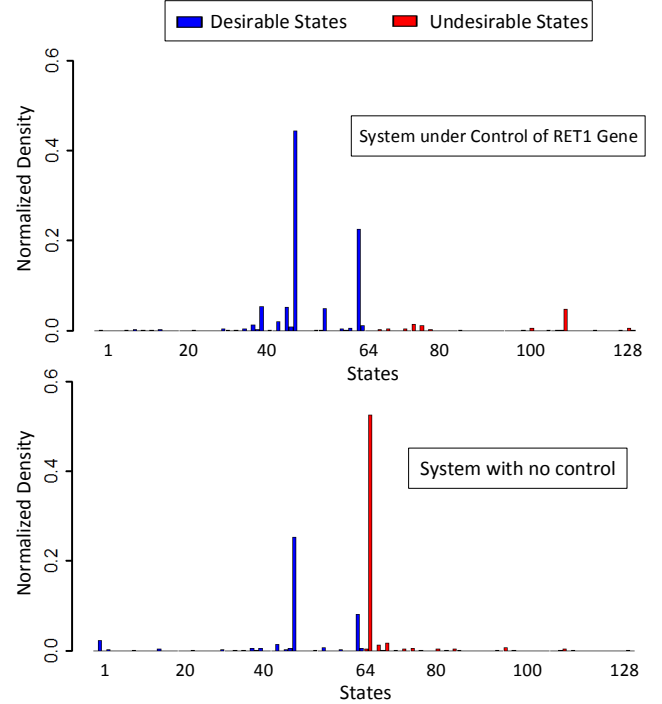


Fig. 5: Relative frequency of visited states under the control policy obtained by SGP-SARSA and under no control. Desirable (inactive WNT5A) and undesirable (active WNT5A) states are indicated by black and red colors, respectively.

E. Comparison of Performance of V_BKF and Q_MDP Algorithms against the SGP-SARSA Algorithm

Finally, we compare the performance of SGP-SARSA with two state-feedback controllers V_BKF [18] and Q_MDP [20]. The intensity of uncertainty of intervention process is set to be $q = 0.1$. The average cost per step and the fraction of observed desirable states in the long run for the three algorithms, for different process and observation noise levels, are presented

in Table III.

We can observe that SGP-SARSA obtains lower average cost per step than Q_MDP and V_BKF, especially in the presence of high measurement noise. The reason is that the underlying Boolean dynamical system is less identifiable in the presence of noisy measurements, and therefore, the policies obtained by Q_MDP and V_BKF, which are not based on the belief space but solely on the results of estimation of the underlying Boolean dynamical system, become less valid. In addition, we observe that RET1 is a better control input in comparison to HADHB for reducing the activation of WNT5A in all cases.

TABLE III: Average cost per step and average fraction of desirable states visited for different controllers and control genes.

p	$\sigma_j^0 = \sigma_j^1$	Method	RET1		HADHB	
			Cost	Fraction	Cost	Fraction
0.01	10	GP-SARSA	0.85	0.84	1.59	0.69
		Q_MDP	0.99	0.78	1.82	0.64
		V_BKF	0.98	0.78	1.86	0.63
	15	GP-SARSA	1.28	0.74	1.84	0.64
		Q_MDP	1.64	0.67	2.19	0.56
		V_BKF	1.62	0.67	2.17	0.57
0.05	10	GP-SARSA	1.79	0.64	2.18	0.58
		Q_MDP	2.09	0.58	2.32	0.54
		V_BKF	2.07	0.58	2.34	0.54
	15	GP-SARSA	2.08	0.60	2.53	0.50
		Q_MDP	2.35	0.54	2.73	0.46
		V_BKF	2.37	0.53	2.75	0.47

VII. CONCLUSION

In this paper, The POBDS model was used in conjunction with Gaussian process and reinforcement learning to achieve near-optimal infinite-horizon control of gene regulatory networks with uncertainty in both the inputs (intervention) and outputs (measurements). The cost function in the belief and intervention spaces was modeled by Gaussian process and learning was achieved using a sparsified version of the GP-SARSA algorithm. The methodology was investigated thoroughly by a series of numerical experiments using synthetic gene-expression data generated by a gene regulatory network involved in melanoma metastasis. An interesting fact observed in the experiments is that if the uncertainty in the control input is large, the behavior of the controlled system is worse than that of a free-evolving system evolving without control. Future work will consider adaptive version of the controllers described here.

ACKNOWLEDGMENT

The authors acknowledge the support of the National Science Foundation, through NSF award CCF-1320884.

REFERENCES

- [1] G. Karlebach and R. Shamir, "Modelling and analysis of gene regulatory networks," *Nature Reviews Molecular Cell Biology*, vol. 9, no. 10, pp. 770–780, 2008.
- [2] I. Shmulevich, E. R. Dougherty, and W. Zhang, "From Boolean to probabilistic Boolean networks as models of genetic regulatory networks," *Proceedings of the IEEE*, vol. 90, no. 11, pp. 1778–1792, 2002.
- [3] N. Friedman, M. Linial, I. Nachman, and D. Pe'er, "Using Bayesian networks to analyze expression data," *Journal of computational biology*, vol. 7, no. 3-4, pp. 601–620, 2000.
- [4] D. Cheng and H. Qi, "Controllability and observability of Boolean control networks," *Automatica*, vol. 45, no. 7, pp. 1659–1667, 2009.
- [5] M. Xiao, W. X. Zheng, G. Jiang, and J. Cao, "Stability and bifurcation analysis of arbitrarily high-dimensional genetic regulatory networks with hub structure and bidirectional coupling," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 63, no. 8, pp. 1243–1254, 2016.
- [6] M. Xiao and J. Cao, "Genetic oscillation deduced from hopf bifurcation in a genetic regulatory network with delays," *Mathematical biosciences*, vol. 215, no. 1, pp. 55–63, 2008.
- [7] M. Xiao, W. X. Zheng, and J. Cao, "Stability and bifurcation of genetic regulatory networks with small RNAs and multiple delays," *International Journal of Computer Mathematics*, vol. 91, no. 5, pp. 907–927, 2014.
- [8] J. Cao and F. Ren, "Exponential stability of discrete-time genetic regulatory networks with delays," *IEEE Transactions on Neural Networks*, vol. 19, no. 3, pp. 520–523, 2008.
- [9] A. Datta, A. Choudhary, M. L. Bittner, and E. R. Dougherty, "External control in markovian genetic regulatory networks," *Machine learning*, vol. 52, no. 1-2, pp. 169–191, 2003.
- [10] R. Pal, A. Datta, and E. R. Dougherty, "Optimal infinite-horizon control for probabilistic Boolean networks," *Signal Processing, IEEE Transactions on*, vol. 54, no. 6, pp. 2375–2387, 2006.
- [11] D. Cheng and H. Qi, "A linear representation of dynamics of Boolean networks," *IEEE Transactions on Automatic Control*, vol. 55, no. 10, pp. 2251–2258, 2010.
- [12] U. Braga-Neto, "Optimal state estimation for Boolean dynamical systems," in *Signals, Systems and Computers (ASIOMAR), 2011 Conference Record of the Forty Fifth Asilomar Conference on*, pp. 1050–1054, IEEE, 2011.
- [13] M. Imani and U. Braga-Neto, "Maximum-likelihood adaptive filter for partially-observed Boolean dynamical systems," *IEEE Transactions on Signal Processing*, vol. 65, no. 2, pp. 359–371, 2017.
- [14] M. Imani and U. Braga-Neto, "Optimal state estimation for Boolean dynamical systems using a Boolean kalman smoother," in *2015 IEEE Global Conference on Signal and Information Processing (GlobalSIP)*, pp. 972–976, IEEE, 2015.
- [15] M. Imani and U. Braga-Neto, "Particle filters for partially-observed Boolean dynamical systems," *arXiv preprint arXiv:1702.07269*, 2017.
- [16] L. D. McCleenny, M. Imani, and U. Braga-Neto, "Boolean kalman filter with correlated observation noise," in *the 42nd IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP 2017)*, IEEE, 2017.
- [17] M. Imani and U. Braga-Neto, "Optimal gene regulatory network inference using the Boolean kalman filter and multiple model adaptive estimation," in *2015 49th Asilomar Conference on Signals, Systems and Computers*, pp. 423–427, IEEE, 2015.
- [18] M. Imani and U. Braga-Neto, "State-feedback control of partially-observed Boolean dynamical systems using RNA-seq time series data," in *2016 American Control Conference (ACC2016)*, IEEE, 2016.
- [19] M. Imani and U. Braga-Neto, "Multiple model adaptive controller for partially-observed Boolean dynamical systems," in *Proceedings of the 2017 American Control Conference (ACC2017)*, Seattle, WA, 2017.
- [20] M. L. Littman, A. R. Cassandra, and L. P. Kaelbling, "Learning policies for partially observable environments: Scaling up," in *International Conference on Machine Learning (ICML)*, Morgan Kaufmann, 1995.
- [21] J. Pineau, G. Gordon, and S. Thrun, "Anytime point-based approximations for large pomdps," *Journal of Artificial Intelligence Research*, pp. 335–380, 2006.
- [22] J. Pineau, G. Gordon, S. Thrun, *et al.*, "Point-based value iteration: An anytime algorithm for pomdps," in *IJCAI*, vol. 3, pp. 1025–1032, 2003.
- [23] M. Imani and U. Braga-Neto, "Point-based value iteration for partially-observed Boolean dynamical systems with finite observation space," in *the 55th IEEE Conference on Decision and Control (CDC)*, IEEE, 2016.
- [24] T. Smith and R. Simmons, "Heuristic search value iteration for pomdps," in *Proceedings of the 20th conference on Uncertainty in artificial intelligence*, pp. 520–527, AUAI Press, 2004.

- [25] T. Smith and R. Simmons, "Point-based pomdp algorithms: Improved analysis and implementation," *arXiv preprint arXiv:1207.1412*, 2012.
- [26] J. M. Porta, N. Vlassis, M. T. Spaan, and P. Poupart, "Point-based value iteration for continuous pomdps," *The Journal of Machine Learning Research*, vol. 7, pp. 2329–2367, 2006.
- [27] Y. Chen, E. Dougherty, and M. Bittner, "Ratio-based decisions and the quantitative analysis of cDNA microarray images," *Journal of Biomedical Optics*, vol. 2, no. 4, pp. 364–374, 1997.
- [28] J. Hua, C. Sima, M. Cypert, G. Gooden, S. Shack, L. Alla, E. Smith, J. Trent, E. Dougherty, and M. Bittner, "Dynamical analysis of drug efficacy and mechanism of action using gfp reporters," *Journal of Biological Systems*, vol. 20, p. 403, 2012.
- [29] R. Weinberg, *The Biology of Cancer*. Princeton: Garland Science, 2006.
- [30] D. P. Bertsekas, *Dynamic programming and optimal control*, vol. 1. Athena Scientific Belmont, MA, 1995.
- [31] Y. Engel, S. Mannor, and R. Meir, "Reinforcement learning with gaussian processes," in *Proceedings of the 22nd international conference on Machine learning*, pp. 201–208, ACM, 2005.
- [32] C. E. Rasmussen, "Gaussian processes for machine learning," 2006.
- [33] B. Schölkopf and A. J. Smola, *Learning with kernels: support vector machines, regularization, optimization, and beyond*. MIT press, 2002.
- [34] Y. Engel, S. Mannor, and R. Meir, "Bayes meets bellman: The gaussian process approach to temporal difference learning," in *ICML*, vol. 20, p. 154, 2003.
- [35] R. S. Sutton and A. G. Barto, "Reinforcement learning: An introduction," 1998.
- [36] M. Gavić and S. Young, "Gaussian processes for pomdp-based dialogue manager optimization," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 22, no. 1, pp. 28–40, 2014.
- [37] B. Schölkopf, A. Smola, and K.-R. Müller, "Nonlinear component analysis as a kernel eigenvalue problem," *Neural computation*, vol. 10, no. 5, pp. 1299–1319, 1998.
- [38] W. Liu, I. Park, and J. C. Principe, "An information theoretic approach of designing sparse kernel adaptive filters," *IEEE Transactions on Neural Networks*, vol. 20, no. 12, pp. 1950–1961, 2009.
- [39] Y. Engel, *Algorithms and representations for reinforcement learning*. Citeseer, 2005.
- [40] E. R. Dougherty, R. Pal, X. Qian, M. L. Bittner, and A. Datta, "Stationary and structural control in gene regulatory networks: basic concepts," *International Journal of Systems Science*, vol. 41, no. 1, pp. 5–16, 2010.
- [41] M. Bittner, P. Meltzer, Y. Chen, Y. Jiang, E. Seftor, M. Hendrix, M. Radmacher, R. Simon, Z. Yakhini, A. Ben-Dor, *et al.*, "Molecular classification of cutaneous malignant melanoma by gene expression profiling," *Nature*, vol. 406, no. 6795, pp. 536–540, 2000.
- [42] A. T. Weeraratna, Y. Jiang, G. Hostetter, K. Rosenblatt, P. Duray, M. Bittner, and J. M. Trent, "Wnt5a signaling directly affects cell motility and invasion of metastatic melanoma," *Cancer cell*, vol. 1, no. 3, pp. 279–288, 2002.



Ulisses M. Braga-Neto is an Associate Professor at the Department of Electrical and Computer Engineering and a member of the Center for Bioinformatics and Genomic Systems Engineering at Texas A&M University, College Station, TX. He holds a Ph.D. degree in Electrical and Computer Engineering from The Johns Hopkins University, Baltimore, MD and has held post-doctoral positions at the University of Texas M.D. Anderson Cancer Center, Houston, TX and at the Oswaldo Cruz Foundation, Recife, Brazil. His research interests include Pattern Recognition and Statistical Signal Processing. Dr. Braga-Neto is a Senior Member of the IEEE. He is author of the textbook *Error Estimation for Pattern Recognition* (IEEE-Wiley, 2015) and has received the NSF CAREER Award for his work in this area.



Mahdi Imani received his B.Sc. degree in Mechanical Engineering and his M.Sc. degree in Electrical Engineering, both from University of Tehran in 2012 and 2014. He is currently a Ph.D. student at the Department of Electrical and Computer Engineering of Texas A&M University, College Station, TX. His current research interests include estimation and control of stochastic Boolean dynamical systems, with applications in genomic signal processing.