Optimal Intervention in Asynchronous Genetic Regulatory Networks

Babak Faryabi, *Student Member, IEEE*, Jean-François Chamberland, *Member, IEEE*, Golnaz Vahedi, *Student Member, IEEE*, Aniruddha Datta, *Senior Member, IEEE*, and Edward R. Dougherty, *Member, IEEE*

***Abstract*—There is an ongoing effort to design optimal intervention strategies for discrete state-space synchronous genetic regulatory networks in the context of probabilistic Boolean networks; however, to date, there has been no corresponding effort for asynchronous networks. This paper addresses this issue by postulating two asynchronous extensions of probabilistic Boolean networks and developing control policies for both. The first extension introduces deterministic gene-level asynchronism into the constituent Boolean networks of the probabilistic Boolean network, thereby providing the ability to cope with temporal context sensitivity. The second extension introduces asynchronism at the level of the gene activity profiles. Whereas control policies for both standard probabilistic Boolean networks and the first proposed extension are characterized within the framework of Markov decision processes, asynchronism at the profile level results in control being treated in the framework of semi-Markov decision processes. The advantage of the second model is the ability to obtain the necessary timing information from sequences of gene-activity profile measurements. Results from the theory of stochastic control are leveraged to determine optimal intervention strategies for each class of proposed asynchronous regulatory networks, the objective being to reduce the time duration that the system spends in undesirable states.**

***Index Terms*—Asynchronous genetic regulatory networks, optimal stochastic control, semi-Markov decision processes, translational genomics.**

I. Introduction

A salient problem in genomic signal processing is the design of intervention strategies to beneficially alter the dynamics of a gene regulatory network, for instance, to reduce the steady-state mass of states favorable to metastasis in cancer cells [1], [2]. To date, regulatory intervention has been studied in the context of probabilistic Boolean networks (PBNs) [3], specifically, the development of intervention strategies based on associated Markov chains. Methods have progressed from one-time intervention based on first-passage times [4], to using dynamic programming to design optimal finite-horizon control policies [5], to stationary infinite-horizon policies designed to alter the steady-state distribution [6]. Common to all of these approaches is the assumption of synchronous timing. In this paper, we relax that assumption and consider intervention in asynchronous networks. This involves defining asynchronous probabilistic Boolean networks and treating the problem in the framework of asynchronous processes. We consider two approaches: asynchronism relative to the genes themselves and asynchronism relative to the state-space of gene-activity profiles.

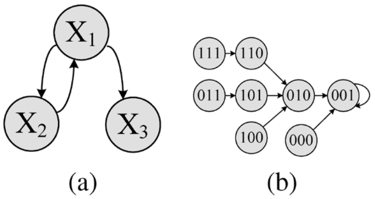

Fig. 1. Presentation of a regulatory graph and its corresponding oriented graph for an arbitrary 3-gene Boolean network. (a) Regulatory graph; (b) oriented graph.

In a rule-based regulatory network, a *regulatory graph* defines the multivariate interactions among the components. From here on, we use the term *gene* in place of any general biological components, e.g., genes and proteins, involved in a regulatory network. The vertices of a regulatory graph are the genes or *nodes*. A directed edge starts from a predictor vertex and ends at an influenced vertex. All the vertices directly connected to a node are its predictors. A regulatory rule defines the multivariate effects of predictors on the vertex. The node values are selected from a set of possible quantization levels to facilitate the modeling of gene interactions by logical rules. The discrete formalism of rule-based regulatory networks is plausible for many classes of biological systems. Strong evidences suggest that the input-output relations of regulatory interactions are sigmoidal and can be well approximated by step functions [7], [8]. Fig. 1(a) shows the regulatory graph of a hypothetical three-gene network. Per the regulatory graph in Fig. 1(a), node (gene) x_1 is predicted by nodes (genes) x_2. Node (gene) x_1 is the predictor of both nodes (genes) x_2 and x_3.

To completely specify a class of regulatory networks, we need to adopt an updating scheme. The choice of the updating scheme plays a crucial rule in the dynamical behavior of the network. Given the updating scheme, we can depict the dynamical evolution of genes by translating the information of the regulatory graph and the regulatory rules into an *oriented graph*. The

Manuscript received September 11, 2007; revised March 8, 2008. This work was supported in part by the National Science Foundation under Grants ECS-0355227, CCF-0514644, and ECCS-0701531, in part by the National Cancer Institute under Grants R01 CA-104620 and CA-90301, and in part by the Translational Genomics Research Institute. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Dan Schonfeld.

B. Faryabi, J.-F. Chamberland, G. Vahedi, and A. Datta are with the Department of Electrical and Computer Engineering, Texas A&M University, College Station, TX 77843 USA (e-mail: bfariabi@ece.tamu.edu; chmbrlnd@ece.tamu.edu; golnaz@ece.tamu.edu; datta@ece.tamu.edu).

E. R. Dougherty is with the Department of Electrical and Computer Engineering, Texas A&M University, College Station, TX 77843 USA, and also with the Computational Biology Division, Translational Genomics Research Institute, Phoenix, AZ 85004 USA (e-mail: edward@ece.tamu.edu).

Digital Object Identifier 10.1109/JSTSP.2008.923853

vertex of an oriented graph is a *logical state*, which is the aggregated values of all the nodes at a given time. There is an edge between two logical states of an oriented graph if a transition can occur from one vertex to the other. For instance, Fig. 1(b) shows the oriented graph corresponding to the regulatory graph in Fig. 1(a). According to this oriented graph, whenever the values of the nodes x_1, x_2, and x_3 are 1, 0, and 1, respectively, if all the nodes update synchronously, then the next logical state is "010" ($x_1 = 0$, $x_2 = 1$, and $x_3 = 0$). A regulatory graph is a static representation of interactions among biological components, whereas an oriented graph shows the dynamics of interactions among these components. We can practically observe timing information related to the dynamical representation of biological component interactions, that is, timing relative to the oriented graph.

Two factors motivate the adoption of synchronous updating schemes in rule-based regulatory networks: they are more mathematically tractable and they require significantly less data for inference. In particular, substantial time-course data are required to characterize asynchronism. To date, the lack of sufficient time-course data has prohibited the inference of any realistic alternative asynchronous models; however, the situation can be expected to improve in the future.

Synchronous abstraction is used under the implicit assumption that asynchronous updating will not unduly alter the properties of a system central to the application of interest [9]. Clearly, some properties will be altered. In Fig. 1(b), if all nodes are not simultaneously updated, then the transition from 101 to 010 may not occur. Various potential issues with synchronous networks have been noted. For instance, synchronous abstraction may produce spurious attractors in rule-based networks [10]. In the same vein, deviation from synchronous updating modifies the attractor structure of Boolean networks [11] and can change their long-run behavior [12]. From a biological perspective, interactions among genes causing transcription, translation, and degeneration occur over a wide range of time-scales.

These observations suggest that we examine intervention in asynchronous models. Since relaxing the synchronous assumption will alter the long-run behavior of a regulatory model, we need alternative approaches to influence network dynamics in asynchronous models. In this paper we propose two new rule-based asynchronous models and methods to derive effective intervention strategies based on these models. The first model introduces asynchronism in probabilistic Boolean networks at the node level. The second model extends this approach by considering asynchronism at the logical-state level. Whereas the first method is akin to currently proposed asynchronous models, we will argue that the second approach is more suitable from a translational perspective.

To date, asynchronism has been introduced into Boolean networks by updating each node based on its period. These studies try to understand generic characteristics of asynchronous updating schemes in randomly generated Boolean networks. To accomplish this aim, a wide range of artificial asynchronous updating protocols with different degrees of freedom in the selection of the updating period for each node has been postulated.

We categorize previously proposed asynchronous protocols into two groups. In the first category, termed *stochastic asynchronous* protocols, the updating period of each node is randomly selected based on a given distribution [11]–[14]. In the second category, termed *deterministic asynchronous* protocols, the updating period of each node is fixed, and can differ from one node to another [9], [12], [15]. There have also been studies that consider both stochastic and deterministic asynchronous protocols in an effort to investigate the predictability of Boolean networks when asynchronous updating schemes are used instead of synchronous ones [16], [17].

The study of both randomly generated and experimentally validated Boolean networks reveals that stochastic asynchronism has some limitations. Stochastic asynchronous updating methods can significantly change the properties of oriented graphs [9], [15]. Starting from wild-type gene expressions, neither the Boolean networks of [16] or [17] successfully predict the anticipated long-run attractors of their networks. Earlier studies indicate that constraining the degrees of freedom in the asynchronous protocols can improve the predictability of Boolean networks. More structured asynchronous protocols predict the long-run behavior of Boolean networks more effectively by representing their cyclic attractors [14], [17]. It must be remembered that in all of these studies of asynchronism, the timing protocols have been modeled mathematically without biological verification. At this point, perhaps, all we can say is that synchronism or asynchronism are modeling assumptions and the choice in a specific circumstance depends upon the available data and application.

When the context of a biological system is known, there is a consensus that asynchronism in regulatory networks is deterministic rather than random [9]; however, deterministic asynchronous Boolean networks pose practical challenges. Even if we can measure the level of each node in isolation while the other nodes remain constant, at best, we could produce estimates for updating periods. Owing to the effects of measurement noise and the existence of latent variables, we cannot exactly specify them. Focusing on the effects of latent variables, as is customary when considering probabilistic Boolean networks, at best we can estimate a set consisting of the most probable updating periods for each gene in the network; each set depending on the (unknown) status of latent variables. A set of updating periods, whose members are the deterministic periods of each node in the regulatory network, defines the updating protocol of a *deterministic asynchronous Boolean network*. This means that there is a finite collection of deterministic asynchronous Boolean networks that defines the dynamics of the system. The updating periods of nodes depend on the temporal context of the biological system, which can be influenced by latent variables. Having the probabilities of selecting each context, the model selects one of the constituent deterministic asynchronous Boolean networks at each updating instant. The system evolves according to the selected constituent deterministic asynchronous Boolean network until its constituent network changes. This approach of introducing asynchronism into PBNs extends the currently favored approach of studying asynchronism in regulatory models. The proposed model, called a *deterministic-asynchronous context-sensitive probabilistic Boolean network* (DA-PBN), is an extension of probabilistic Boolean networks in which the time scales of various biological updates can be different. The term *proba-*

bilistic emphasizes the random selection of a context, while the term *deterministic* refers to the asynchronous protocol within each context of the regulatory network.

Although DA-PBN provides a means to study intervention using an asynchronous regulatory model, earlier research suggests that the assumption of node asynchronism has drawbacks. Although it appears not to have been directly mentioned in previous studies, asynchronously updating the nodes changes the global behavior of regulatory networks due to changing their oriented graph, which models the dynamics of the system. Along this line, it has been shown that small perturbations do not settle down in a random Boolean network with node asynchronism. Consequently, the asynchronous network is in the chaotic regime while its synchronous counterpart is in the critical regime [18]. The studies of experimentally validated Boolean networks in [17] and [19] suggest that oriented graphs of given Boolean networks provide accurate predictability, whereas the oriented graphs of networks utilizing the same Boolean rules with asynchronously updated nodes are very complex and possess many incompatible or unrealistic pathways.

From these observations, we gather that an asynchronous regulatory model should maintain the topology of the oriented graph as specified by the logical rules governing the interactions between genes. In other words, regulatory models should accurately translate the logical relationships, i.e., the regulatory graph, governing the interactions of nodes to the oriented graph specifying the dynamics of the model. Moreover, they should enable the analysis of the temporal behaviors of biological systems. Since our objective here is to alter the long-run behavior of biological systems via an effective intervention strategy, our regulatory models should not only possess the previous two characteristics, but they should also be inferable from the empirical data.

Due to the aforementioned observations, we propose a second asynchronous regulatory network model, termed *semi-Markov asynchronous regulatory networks* (SM-ARN). In the SM-ARN, the asynchronism is at the logical state instead of the node. In the SM-ARN model, the empirically measurable timing information of biological systems is incorporated into the model. This measurable timing information determines the typical time delay between transitions from one logical state to another. Since the order of updating nodes and their relative time delays depends on the levels of other regulatory components, estimating the updating time of each gene in isolation, and independent of the values of other genes, is highly problematic, if not impossible. Time-course data enable the estimation of intertransition times between logical states, not the updating time of each node, and it is at the logical-state level that we will introduce asynchronism.

An SM-ARN is specified with two sets of information. The first set determines the rule-based multivariate interactions between genes. Considering simultaneous updating, we can specify the oriented graph of the model based on this information. In other words, the first set of information specifies a PBN, which is generated from a given set of Boolean functions for updating each gene. The generated oriented graph guarantees the predictability of the rule-based topology. The second set of information consists of the distributions of intertransition intervals between any two logical states that are directly connected. These values can be empirically inferred from time-course data.

We show that the design of optimal intervention strategies for the DA-PBN model can be mapped to the existing infinite-horizon intervention methods based on Markov decision processes (MDP) [6], although its corresponding oriented graph has a larger state space. To design optimal intervention strategies based on the SM-ARN model, we apply results from the theory of semi-Markov decision processes (SMDP). Appropriately formulating the problem of intervention in the SM-ARN model, we devise an optimal control policy that minimizes the duration that the system spends in undesirable states.

In Section II, we define a DA-PBN and show that a DA-PBN can be represented as a Markov chain. Thereafter, using Markov decision processes, we design optimal intervention strategies to control the dynamical behavior of the model in the long run. Reduction of the aggregated probability of undesirable states is the objective of an intervention strategy. The SM-ARN model is introduced in Section III. Having the objective of reducing the time that the regulatory network spends in undesirable states, we derive optimal intervention strategies for both general and special intertransition time distributions based on semi-Markov decision processes. As a numerical study, in Section IV, we apply the SM-ARN intervention method to control a regulatory model of the mammalian cell-cycle.

II. Intervention in Deterministic-Asynchronous Context-Sensitive PBNs

A DA-PBN is an extension of probabilistic Boolean network in which different time-scales for various biological processes are allowed. Each node, or biological component, is updated based on an individual period, which may differ from one component to another. Yet, the updating period of each node is fixed given the context of the network. The intent of context-sensitivity is to incorporate the effect of latent variables not directly captured in the model. The behavior of these latent variables influences both regulation and updating periods of nodes. The uncertainty about the context of a regulatory network resulting from latent variables is captured through a probability measure on the possible states. The exact updating periods and functions of nodes cannot be practically specified. At best, we can estimate the set of possible updating periods and corresponding updating functions for each node. As a stochastic Boolean network model with asynchronous updates, a DA-PBN expands the benefits of traditional PBNs by adding the ability to cope with temporal context as well as regulatory context.

Introducing asynchronism at the node level follows the existing approaches to study asynchronous regulatory networks. Our objectives for introducing asynchronism via the DA-PBN model are twofold. First, we show that the synchronous formalism of PBNs can be relaxed to introduce asynchronous PBNs. Second, we provide a methodology to derive optimal intervention strategies for the DA-PBN model.

A. Deterministic-Asynchronous Context-Sensitive PBN

As with a synchronous PBN, in a DA-PBN, node values are quantized to a finite number of levels. In the framework of gene regulation, a DA-PBN consists of a sequence $V = \{x_i\}_{i=1}^n$, of n nodes, where $x_i \in \{0, 1, \ldots, d-1\}$. Each $x_i (i = 1, \ldots, n)$ represents the expression value of a gene selected from d possible quantization levels. It is common to mix terminology by referring to x_i as the ith gene. A DA-PBN is composed of a collection of N constituent deterministic-asynchronous Boolean networks (DA-BN). In a DA-PBN, the active DA-BN changes at updating instants selected by a binary *switching* random variable. A DA-PBN acts like one of its constituent DA-BNs, each being referred to as a *context*, between two switching instants.

The lth DA-BN $(V, \boldsymbol{f}_l, \boldsymbol{\Theta}_l)$ is defined by two vector-valued functions. The vector-valued function $\boldsymbol{f}_l$ consists of n predictors, $\boldsymbol{f}_l = (f_{l1}, \ldots, f_{ln})$, where $f_{li} : \{0, \ldots, d-1\}^n \rightarrow \{0, \ldots, d-1\}$ denotes the predictor of gene i. The vector-valued function $\boldsymbol{\Theta}_l$ consists of n updating components, $\boldsymbol{\Theta}_l = \{\theta_{l1}, \ldots, \theta_{ln}\}$. Each function $\theta_{li} : \mathbb{N} \rightarrow \{0, 1\}$ is defined with a pair of fixed parameters, (a_{li}, b_{li}). The parameter $a_{li} \in \mathbb{N}$ specifies the updating period of gene i, when context l is selected. The parameter $b_{li} \in \{0, \ldots, a_{li} - 1\}$ further differentiates the exact updating instant of each gene within its updating period. The two degrees of freedom in θ_{li} are sufficient to assign any instant of time as the updating time of gene i

$$\theta_{li}(t) = \begin{cases} 1, & \text{if } t \equiv b_{li} \pmod{a_{li}} \\ 0, & \text{otherwise.} \end{cases} \tag{1}$$

At each updating instant a decision is made whether to switch the current constituent DA-BN. The switching probability q is a system parameter. If the current DA-BN is not switched, then the DA-PBN behaves as a fixed DA-BN and genes are updated synchronously according to the current constituent network

$$x_i(t+1) = \begin{cases} f_{li}(x_1(t), \ldots, x_n(t)), & \text{if} \quad \theta_{li}(t+1) = 1 \\ x_i(t), & \text{if} \quad \theta_{li}(t+1) = 0. \end{cases} \tag{2}$$

If a switch occurs, then a new constituent network is randomly selected according to a selection probability measure $\{r_l\}_{l=1}^N$. After selecting the new constituent network $\hat{l}$, the values of the genes are updated using (2), but with $\boldsymbol{f}_{\hat{l}}$ and $\boldsymbol{\Theta}_{\hat{l}}$ instead.

We consider PBNs with perturbation, in which each gene may change its value with small probability p at each time unit. If $\gamma_i(t)$ is a Bernoulli random variable with parameter p then the value of gene i could be perturbed at each dynamical step as follows:

$$x_i(t+1) = x_i(t) \oplus \gamma_i(t+1). \tag{3}$$

Such a perturbation model enables us to capture the realistic situation where the activity of a gene undergoes a random alteration. As we will see later, in addition, it guarantees that the Markov chain modeling the oriented graph of DA-PBN has a unique steady-state distribution.

To date, PBNs have been applied with $d = 2$ and $d = 3$. If $d = 2$, then the constituent networks are Boolean, with 0 or 1 meaning OFF or ON, respectively. The case where $d = 3$ arises when we consider a gene to be down-regulated (0), up-regulated (2), or invariant (1). This situation commonly occurs with cDNA microarrays, where a ratio is taken between the expression values on the test channel (usually red) and the base channel (usually green). The binary or ternary cases are considered in numerical studies [20]. Although we focus on the binary case in our numerical studies, the methodologies developed in this paper are applicable to any general d. Indeed, the goal of this paper is to introduce frameworks for optimal intervention in asynchronous genetic regulatory networks.

B. Stochastic Control of a DA-PBN

Although the definition of a DA-PBN enables us to study the behavior of a regulatory network in the long run, it does not provide a systematic means for its alteration. We propose a synchronization method for DA-PBNs. The synchronization method provides a synchronous version of a DA-PBN's oriented graph. The synchronized oriented graph sets the stage for designing optimal intervention strategies to alter the dynamical behavior of DA-PBNs.

To study the dynamical behavior of a PBN, the gene-activity profile is considered as the logical state of its oriented graph. The gene-activity profile (GAP) is an n-digit vector $\boldsymbol{x}(t) = (x_1(t), \ldots, x_n(t))$ giving the expression values of genes at time t, where $\boldsymbol{x}(t) \in \{0, \ldots, d-1\}^n$ [7]. There is a natural bijection between the GAP, $\boldsymbol{x}(t)$, and the decimal number $z(t)$ taking values in $\mathcal{W} = \{0, \ldots, d^n - 1\}$. The decimal representation of a GAP facilitates the visualization of the intervention in a DA-PBN.

In the synchronization of a DA-PBN, we augment the GAP of the network and define the *augmented logical state*, $\hat{\boldsymbol{x}}(t)$. To synchronize the oriented graph of a DA-PBN, we encode all the dynamical steps within an interval of duration equal to the least common multiple of all the updating periods. The least common multiple of all the updating periods a_{ij}, for $i \in \{1, \ldots, N\}$ and $j \in \{1, \ldots, n\}$

$$\xi = LCM(a_{11}, \ldots, a_{1n}, \ldots, a_{N1}, \ldots, a_{Nn}) \tag{4}$$

defines the number of a new nodes, m, added to the GAP. The integer m is the smallest integer larger than the logarithm to the base d of ξ

$$m = \lceil \log_d(\xi) \rceil . \tag{5}$$

The value of m determined by (5) is a nonoptimal number of nodes required to distinguish all the time steps within one ξ. Hence, the augmented logical state of a DA-PBN is composed of the GAP and m new nodes, along with the context $\kappa(t)$ of the DA-PBN at each time step t

$$\hat{\boldsymbol{x}}(t) = (x_1(t), \ldots, x_n(t), x_{n+1}(t), \ldots, x_{n+m}(t), \kappa(t)) . \tag{6}$$

Fig. 2 shows the time instants at which the genes of a hypothetical three-gene DA-PBN are updated. The updating function θ_{l1} of x_1 has the parameters $(a_{l1} = 2, b_{l1} = 1)$. Similarly, the parameters of the updating functions of genes x_2 and x_3 are $(a_{l2} = 2, b_{l2} = 0)$ and $(a_{l3} = 3, b_{l3} = 0)$, respectively. The pattern of updates is repeated after each 6 updating instants. We

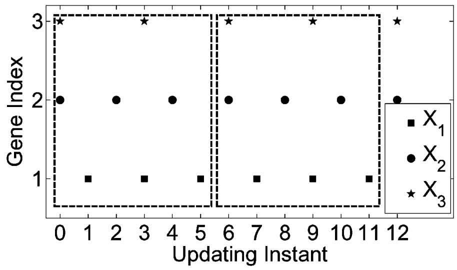

Fig. 2. Schematic of updating instants of genes of a DA-PBN with $(a_{l1} = 2, b_{l1} = 1)$, $(a_{l2} = 2, b_{l2} = 0)$, and $(a_{l3} = 3, b_{l3} = 0)$. The pattern of updates is repeated at each LCM ξ shown with a dashed-line box. Each marker indicates the instant in which the corresponding gene updates its value.

can use three extra nodes to code all the instants in the duration of $\xi = 6$.

The evolution of a synchronized oriented graph with its augmented logical state can be modeled by a stationary discrete-time equation

$$\hat{z}(t+1) = f\left(\hat{z}(t), w(t)\right), \quad \text{for} \quad t = 0, 1, \ldots, \tag{7}$$

where state $\hat{z}(t)$ is an element of the state space $\mathcal{S} = \{(s, c) : s \in \{0, 1, \ldots, d^{n+m} - 1\} \wedge c \in \{1, \ldots, N\}\}$. The disturbance $w(t)$ is the manifestation of uncertainties in the DA-PBN. It is assumed that both the gene perturbation distribution and the network switching distribution are independent and identical for all time steps t.

Hence, the n-gene DA-PBN is modeled as a synchronous context-sensitive PBN with augmented state space. The oriented graph of a synchronous context-sensitive PBN with $n+m$ nodes is a Markov chain with $(d^{n+m} \times N)$ states. Hence, the oriented graph of the system described by (7) can be represented by a Markov chain [21]. Originating from an augmented logical state i, the successor augmented logical state j is selected randomly within the set $\mathcal{S}$ according to the transition probability

$$p_{ij} \triangleq P\left\{\hat{z}(t+1) = j | \hat{z}(t) = i\right\} \tag{8}$$

for all i and j in $\mathcal{S}$. Gene perturbation insures that all the states in the Markov chain communicate with one another. Hence, the finite-state Markov chain is ergodic and has a unique invariant distribution equal to its limiting distribution [4], [22].

Now that the dynamical behavior of a DA-PBN is described by a Markov chain, the theory of Markov decision processes (MDP) can be utilized to find an optimal sequence of interventions [6]. Reducing the likelihood of visiting undesirable states in the long run is the objective of the intervention problem. In the presence of external controls, we suppose that the DA-PBN has α binary control inputs: $c_1(t), \ldots, c_\alpha(t)$. A control $c_i(t)$ can take binary values in {0,1} at each updating instant t. The decimal bijection of the control vector, $u(t) \in \mathcal{C} = \{0, 1, \ldots, 2^\alpha - 1\}$, describes the complete status of all the control inputs. The external control alters the status of the control gene, which can be selected among all the genes in the network. If the ith control at decision epoch t is on, $c_i(t) = 1$, then the state of the control gene is toggled; if $c_i(t) = 0$, then the state of the control gene remains unchanged. In the presence of external control, the system evolution in (7) can be modeled by a discrete-time equation

$$\hat{z}(t+1) = f\left(\hat{z}(t), u(t), w(t)\right) \quad \text{for} \quad t = 0, 1, \ldots \tag{9}$$

Optimal intervention in the DA-PBN is then modeled as a Markov decision process (MDP) with $(d^{n+m} \times N)$ states, the state $\hat{z}(t)$ at any time step t being an augmented logical state. Originating from state i, the successor state j is selected randomly within the set $\mathcal{S}$ according to the transition probability

$$p_{ij}(u) \triangleq P\left(\hat{z}(t+1) = j | \hat{z}(t) = i, u(t) = u\right). \tag{10}$$

We associate a reward-per-stage, $r(i, u, j)$, to each intervention in the system. The reward-per-stage could depend on the origin state i, the successor state j, and the control input u. We also assume that the reward-per-stage is stationary and bounded for all states i, j, and all controls u. We define the expected immediate reward earned in state i, when control u is selected, by

$$\overline{r}(i, u) = \sum_{j \in \mathcal{S}} p_{ij}(u) r(i, u, j). \tag{11}$$

The reward of a transition from a desirable state to an undesirable state is the lowest, and the reward of a transition from an undesirable state to a desirable state is the highest. A state in which metastatic biological components are active is considered to be undesirable.

To define the infinite-horizon problem, we consider the discounted reward formulation. The discounting factor, $\lambda \in (0, 1)$, insures the convergence of the expected total reward over the long run [23]. Including a discounting factor in the expected total reward signifies that the incurred reward at a later time is less significant than the incurred reward at an earlier time. In the case of cancer therapy, the discounting factor attempts to capture the fact that obtaining treatment earlier is better than postponing treatment to a later stage.

Among all admissible policies Π, the infinite-horizon MDP methodology identifies a policy $\pi = \{\boldsymbol{\mu}_0, \boldsymbol{\mu}_1, \ldots\}$, where $\mu_t : \mathcal{S} \rightarrow \mathcal{C}$ is the decision rule at time step t that maximizes the expected total discounted reward. The infinite expected total discounted reward, given the policy π and the initial state i, is

$$J_\pi(i) = \lim_{N \rightarrow \infty} E\left\{\sum_{t=0}^{N-1} \lambda^t r\left(i, \mu_t(i), j\right)\right\}. \tag{12}$$

The vector, $\boldsymbol{J}_\pi$, of expected accumulated-rewards for all the logical states in $\mathcal{S}$ is called the *value function*. We seek a policy that maximizes the value function for each state i. An optimal control policy, π^*, is a solution to the infinite-horizon MDP with discounted reward

$$\pi^*(i) = \arg\max_{\pi \in \Pi} J_\pi(i), \quad \forall i \in \mathcal{S}. \tag{13}$$

The corresponding optimal value function is denoted by J^*. Assuming there is only one control gene as an example, an optimal policy π^* determines a stationary decision rule $\boldsymbol{\mu}^*$ which specifies whether the status of the control gene should be toggled or not at each state $i \in \mathcal{S}$.

A stationary policy is an admissible policy of the form $\pi = \{\boldsymbol{\mu}, \boldsymbol{\mu}, \ldots\}$. The vector $\boldsymbol{J}_\mu$ is its corresponding value function. The stationary policy π is optimal if $J_\mu(i) = J^*(i)$ for any state i. It is known that an optimal policy exists for the discounted infinite-horizon MDP problem, and it is given by the fixed-point solution of the Bellman optimality equation. Moreover, an optimal policy determined by the Bellman optimality equation is also a stationary policy [23].

The intervention problem in a DA-PBN has a discrete-time formulation. On the contrary, as we will show in the next section, the intervention problem in an SM-ARN has a continuous-time formulation. The objective of intervention in the discrete-time problem is to reduce the chance of visiting undesirable states. Since the time between two consecutive epochs of a DA-PBN is fixed, the effect of intervention is equivalent to the reduction of the time spent in undesirable states.

III. Intervention in Semi-Markov Asynchronous Regulatory Networks

According to the discussion in the Introduction, assuming asynchronism at the node level for Boolean networks has practical and theoretical impediments which may prevent independent node updating to serve as a basis for designing effective therapeutic intervention strategies. In particular, the delay and the order of updating a given gene is only observable with respect to the activity level of other genes and proteins involved in the regulatory process. Thus, it is impractical to study the alteration of one specific gene over time, while keeping the levels of all other genes in the model constant. Practically, we can measure GAPs at each observation instant and the intertransition interval between two GAPs can be modeled by a random variable. Introduction of the SM-ARN model is a reasonable modeling strategy to build regulatory networks via the assumption of asynchronously updated logical states. SM-ARN models can practically be inferred from observing biological systems. Using these observations, we can potentially design better intervention strategies.

The results of [17] and [19] implicitly suggest that rule-based regulatory models should maintain the topology of the oriented graph generated by experimentally validated predictors of genes, as if the genes are coupled. Hence, we define the oriented graph of an SM-ARN based on the experimentally validated predictors when the genes are updated simultaneously. To this end, we define a PBN for any SM-ARN based on predictors of genes in the model. This PBN in turn specifies the oriented graph of the SM-ARN.

The SM-ARN model is an application of semi-Markov processes to model rule-based asynchronous networks with the state space consisting of GAPs. Semi-Markov processes constitute a generalization of Markov chains in which the time intervals between transitions are random [24]. We first define the SM-ARN model. Then, we devise optimal intervention methods using Semi-Markov decision processes. To this end, we consider the general formulation of the control problem, as well as three special cases with postulated intertransition interval distributions.

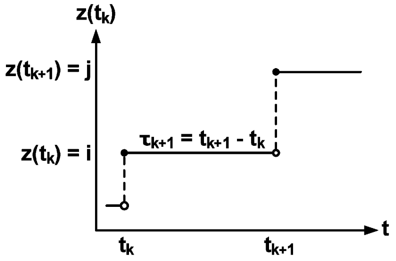

Fig. 3. Schematic of transition in SM-ARN with two consecutive epoch times t_k and t_{k+1}. The intertransition interval, τ_{k+1}, is the sojourn time in state i prior to the transition to state j.

A. Semi-Markov Asynchronous Regulatory Networks

For consistency with DA-PBNs, we use the same notation to define SM-ARNs. We consider a sequence of n genes, $V = \{x_i\}_{i=1}^n$, representing the expression values of genes involved in the model. The expression value, $x_i(t)$, of gene i at time t is selected from d possible quantization levels. We define the states of an SM-ARN as the gene-activity profiles of the nodes in V. As in Section II, the GAP can be considered to be an n-tuple vector $\boldsymbol{x}(t) = (x_1(t), \ldots, x_n(t))$ giving the values of genes at time t, where $\boldsymbol{x}(t) \in \{0, \ldots, d-1\}^n$. The decimal bijection of the GAP is called $z(t)$. At each time $t \in \mathbb{R}^+$, the state, $z(t)$, of the SM-ARN is selected from the set of all possible states, $\mathcal{W} = \{0, \ldots, d^n - 1\}$.

Considering two consecutive epoch times t_k and t_{k+1} per Fig. 3, the state of the SM-ARN for all the times $t_k \le t < t_{k+1}$ is $z(t_k) = i$. At t_{k+1}, the model enters a new state $z(t_{k+1}) = j$. If τ_{k+1} is the time spent in state i prior to transition to state j, then we have $\tau_{k+1} = t_{k+1} - t_k$. In the SM-ARN model, this intertransition interval is modeled with a non-negative random variable with probability distribution

$$P_{ij}(\tau) = P\left(\tau_{k+1} \le \tau | z(t_k) = i, z(t_{k+1}) = j\right). \qquad (14)$$

According to (14), the probability distribution of sojourn time in the current state i prior to transition to the successive state j could depend on both states. We require the intertransition interval distributions, $P_{ij}(\tau)$, for any two directly connected states as one of the two sets of information needed to define an SM-ARN. Time-course data could provide the information leading to these distributions.

Borrowing the methodology proposed in [3], we proceed to define the *embedded-PBN* of an SM-ARN. The embedded-PBN of an SM-ARN models the probabilistic rule-based connections of gene interactions and constitutes the other set of information required for specification of an SM-ARN. The embedded-PBN specifies the oriented graph of the SM-ARN based on the predictors of the genes. The oriented graph of an SM-ARN is a directed graph whose vertices are the states of the SM-ARN in $\mathcal{W}$, and for which there is an edge between any two directly connected states. The weight of each edge is the transition probability between two vertices connected by that edge.

Let $\{\boldsymbol{f}_l\}_{l=1}^N$ be the set of N realizations of the embedded-PBN. If the genes are coupled, then at each simultaneous updating instant, one of the N possible realizations of the embedded-PBN is selected. Each vector-valued function $\boldsymbol{f}_l$ has the form $\boldsymbol{f}_l = (f_{l1}, \ldots, f_{ln})$. Each function $f_{li} : \{0, \ldots, d-1\}^n \to$

$\{0,\ldots,d-1\}$ denotes the predictor of gene i, when the realization l is selected. At each simultaneous updating instant a decision is made whether to switch the context of the network. The switching probability q is a system parameter. If at a particular updating instant, it is decided that the realization of the network should not be switched, then the embedded-PBN behaves as a fixed Boolean network and simultaneously updates the values of all the genes according to their current predictors. If it is decided that the network should be switched, then a realization of the embedded-PBN is randomly selected according to a selection distribution $\{r_l\}_{l=1}^{N}$. After selecting the vector-valued function $\boldsymbol{f}_l$, the values of the genes are updated according to the predictors determined by $\boldsymbol{f}_l$. We assume that the probability of selecting the ith realization, r_i, of the embedded-PBN is known [3]. In addition, we allow perturbations in the embedded-PBN, whereby each gene may change its value with a small probability p at each updating instant.

The graph specifying the relationships among the GAPs of an embedded-PBN, defined as above, can be represented as a Markov chain [3]. On the other hand, the graph of the relationships among the GAPs specified by the embedded-PBN is the regulatory graph of the SM-ARN. Originating from a state $z(t_k)=i$, the successor state $z(t_{k+1})=j$ is selected randomly within the set $\mathcal{W}$ according to the transition probability p_{ij} defined by regulatory graph of the SM-ARN

$$p_{ij}=P\left(z(t_{k+1})=j|z(t_k)=i\right), \quad \text{for all } i,j\in\mathcal{W}. \tag{15}$$

In other words, the oriented graph of an SM-ARN is the same as its regulatory graph. However, the update of states in the oriented graph of an SM-ARN occurs on various time-scales according to intertransition interval distributions. Therefore, the oriented graph of the SM-ARN defined by the embedded-PBN maintains the topology of the oriented graph generated by the experimentally validated predictors of genes.

Gene perturbation ensures that all the states of the SM-ARN communicate in the oriented graph. Hence, the fraction of time that the SM-ARN spends in state i in the long run can be computed [24]

$$p_i=\frac{\pi_i\overline{\tau}(i)}{\sum_{i=0}^{d^n-1}\pi_i\overline{\tau}(i)} \quad \text{w.p. 1.} \tag{16}$$

Here, $\{\pi_i\}_{i=0}^{d^n-1}$ is the steady-state distribution of the GAP in the Markov chain representing the oriented graph of the SM-ARN, and $\{\overline{\tau}(i)\}_{i=0}^{d^n-1}$ is the expected sojourn time in state i, which can be computed from the information in (15) and (14). One can easily verify that the fraction, p_i, of time spent in state i in the long run will be equal to the fraction, π_i, of the transitions to state i if all the nodes are synchronously updated.

B. Stochastic Control of an SM-ARN

In considering the stochastic control of an SM-ARN, as in Section II-B, we suppose that the SM-ARN has α binary control inputs, and $u(t)\in\mathcal{C}$ describes the complete status of all the control inputs at time t. In the presence of external controls, the SM-ARN is modeled as a semi-Markov decision process. At any time t, the state $z(t)$ is selected from $\mathcal{W}$. Originating from state i, the successor state j is selected randomly within the set $\mathcal{W}$ according to the transition probability

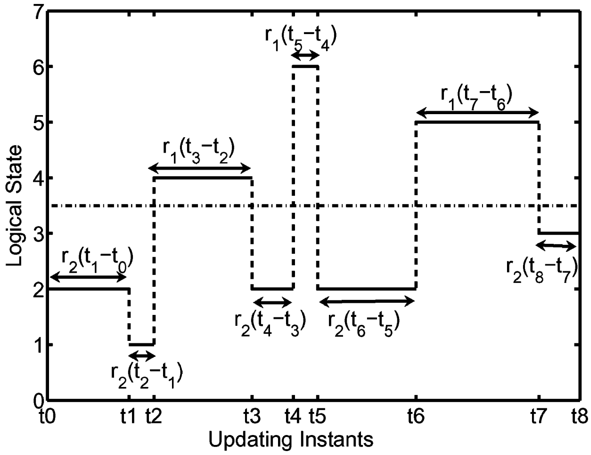

Fig. 4. Schematic of transitions in a hypothetical three-gene SM-ARN along with their epoch times and reward during each sojourn interval. The total reward between two epoch times t_1 and t_2 is less than the total reward between two epoch times t_5 and t_6.

$$p_{ij}(u)\triangleq P\left(z(t_{k+1})=j|z(t_k)=i,u(t_k)=u\right) \tag{17}$$

for all i and j in $\mathcal{W}$ and for all u in $\mathcal{C}$. Moreover, the intertransition interval distribution is also a function of control u

$$P_{ij}(\tau,u)\triangleq P\left(\tau_{k+1}\leq\tau|z(t_k)=i,z(t_{k+1})=j,u(t_k)=u\right) \tag{18}$$

for all states i and j in $\mathcal{W}$, and all controls u in $\mathcal{C}$. The external control alters the status of the control gene, which can be selected among all the genes in the network. If the ith control at decision epoch t is on, $c_i(t)=1$, then the state of the control gene is toggled; if $c_i(t)=0$, then the state of the control gene remains unchanged. We associate a rate of reward $r(z(t),u(t))$ for sojourning in state $z(t)$ per unit of time while the control is $u(t)$. Considering consecutive epoch times t_k and t_{k+1}, the rate of reward $r(z(t),u(t))$ is constant for all $t_k\leq t<t_{k+1}$. It is equal to $r(i,u)$, whenever $z(t_k)=i$ and $u(t_k)=u$. The rate of reward of undesirable states is lower than those for desirable states. We also consider the cost of applying a control action, which reduces the rate of reward of each state.

Fig. 4 shows several epoch times of a hypothetical three-gene SM-ARN. We assume that the undesirable states are the ones with an up-regulated gene in the most significant position in the GAP. We then assign higher rates of reward to desirable states 0 through 3 compared to the undesirable states 4 through 7. Given that r_1 and r_2 are the rates of reward when the model is in undesirable and desirable states, respectively, the reward, $(t_2-t_1)r_2$, gained between two epoch times t_1 and t_2 is less than the reward, $(t_6-t_5)r_2$, gained between the two epoch times t_5 and t_6. We desire an effective intervention policy that maximizes the accumulated reward over time. In other words, we seek a control policy to reduce the time spent in undesirable states with lower rate of reward compared to desirable states with higher rate of reward. In practice, the rates of reward have to capture the relative preferences for the different states. For instance, the reward gained between the two epoch times t_6 and

t_7 may need to be smaller than the reward gained between the two epoch times t_1 and t_2. In order to penalize the sojourn time in undesirable states, the ratio of r_2 to r_1 should be large enough to capture the relative preference for the desirable states. If the intervals between any two epoch times in Fig. 4 were equal, then the problem of intervention in an SM-ARN would reduce to the intervention problem in PBNs. In this intervention problem, the objective is to reduce the number of visits to undesirable states, because the sojourn time in all states is the same, so that reducing the number of visits to undesirable states is directly equivalent to reducing the amount of time spent in these states.

For the reasons articulated in Section II-B, we consider a discounted reward formulation to define the expected total reward. If $\lambda \in (0,1)$ is the discounted factor per unit time and we divide the time unit to small intervals δ, then at each interval the discount is λ/δ, given the initial value is 1. Hence, $(1-(\lambda/\delta))^{\delta t}$ represents the discount over t units of time. As δ goes to zero, the discount goes to $e^{-\lambda t}$.

Among all admissible policies Π, the SMDP methodology finds a policy $\pi = \{\boldsymbol{\mu}_0, \boldsymbol{\mu}_1, \ldots\}$, where $\mu_t : \mathcal{W} \to \mathcal{C}$ is the decision rule at time t, that maximizes the expected total discounted reward. The infinite-horizon expected total discounted reward, given the policy π and the initial state i, is

$$J_\pi(i) = \lim_{N\to\infty} E\left\{\int_0^{t_N} e^{-\lambda t} r\left(z(t), u(t)\right) dt\right\} \tag{19}$$

where t_N is the Nth epoch time. We seek a policy π^* that maximizes the value function for each state i. An optimal control policy is a solution of the SMDP with discounted reward

$$\pi^*(i) = \arg\max_{\pi\in\Pi} J_\pi(i), \quad \forall i \in \mathcal{S}. \tag{20}$$

Intervention using the policy π^* increases the time spent in desirable states determined through appropriate assignment of rate of rewards $r(z(t), u(t))$ to each state-control pair $(z(t), u(t))$. A general solution for this optimization problem is presented in [23]. To make the paper self-contained, we next present the steps of the solution.

Using the intertransition interval distributions in (18) and the transition probabilities in (17), one can define the joint transition distribution of an intertransition interval and the successor state, given the current state and control

$$Q_{ij}(\tau, u) = P\left\{\tau_{k+1} \le \tau, z(t_{k+1}) = j | z(t_k) = i, u(t_k) = u\right\}. \tag{21}$$

Consequently, the expected reward, $R(i,u)$, of a single transition from state $z(t_k) = i$ and control $u(t_k) = u$ can be computed

$$\begin{aligned} R(i,u) &= E\left[\int_0^{\tau} e^{-\lambda t} r(i,u) dt\right] \\ &= r(i,u) \sum_{j=1}^{|\mathcal{W}|} \int_0^{\infty} \frac{1-e^{-\lambda\tau}}{\lambda} Q_{ij}(d\tau, u) \end{aligned} \tag{22}$$

where $|\mathcal{W}|$ is the cardinality of the state space.

There is a recursive relation between the value function $J^N_{\pi_N}$ of stage N and the value function $J^{N-1}_{\pi_{N-1}}$ of stage $(N-1)$

$$\begin{aligned} J^N_{\pi_N}(i) &= \sum_{k=0}^{N-1} E\left[\int_{t_k}^{t_{k+1}} e^{-\lambda t} r\left(z_k, \mu_k(z_k)\right) dt | z(t_0) = i\right] \\ &= R\left(i, \mu_0(i)\right) \\ &\quad + \sum_{j=1}^{|\mathcal{W}|} \int_0^{\infty} e^{-\lambda\tau} Q_{ij}\left(d\tau, \mu_0(i)\right) J^{N-1}_{\pi_{N-1}}(j) \end{aligned} \tag{23}$$

where $z(t_k) = z_k$. The $(N-1)$-stage policy π_{N-1} is the subset of the N-stage policy $\pi_N = \{\mu_0, \mu_1, \ldots, \mu_{N-1}\}$ when μ_0 is excluded. Equation (23) can be rewritten as

$$J^N_{\pi_N}(i) = R\left(i, \mu_0(i)\right) + \sum_{j=1}^{|\mathcal{W}|} m_{ij}\left(\mu_0(i)\right) J^{N-1}_{\pi_{N-1}}(j) \tag{24}$$

where $m_{ij}(u)$ is defined as

$$m_{ij}(u) = \int_0^{\infty} e^{-\lambda\tau} Q_{ij}(d\tau, u). \tag{25}$$

Equation (24) is similar to the Bellman optimality equation in dynamic programming algorithms, in which the expected immediate reward is replaced by the expected reward, $R(i,u)$, of a single transition from state i under control $\mu_0(i) = u$, and $\lambda p_{ij}(u)$ is replaced by $m_{ij}(u)$. Hence, the optimal value function is the unique fixed-point of the Bellman optimality equation

$$J^*(i) = \max_{u\in\mathcal{C}} \left[R(i,u) + \sum_{j=1}^{|\mathcal{W}|} m_{ij}(u) J^*(j)\right]. \tag{26}$$

Using (22) and (25), we can compute the expected single transition reward and $m_{ij}(u)$, respectively, for any SM-ARN. These values define the Bellman optimality equation (26). Any algorithm that solves the infinite-horizon Markov decision process, e.g., value iteration, can be used to find the fixed-point of (26), and also provides an optimal policy which is a solution to problem (20). Here, we consider three hypothetical cases for the intertransition interval distribution. For each case, we formulate the Bellman optimality equation (26).

1) Discrete Distribution: We postulate that the duration of the transcription of a specific gene is almost fixed, given the expression status of other genes in the network. Due to latent variables, we assume that this value is drawn from a set of possible values $\{\tau_{ij}(k,u)\}_{k=1,\ldots,m}$ with probabilities $\{\varrho_{ij}(k,u)\}_{k=1,\ldots,m}$. Using (22), we have

$$R(i,u) = r(i,u) \sum_{j=1}^{|\mathcal{W}|} \sum_{k=1}^{m} \frac{1-\exp(-\lambda\tau_{ij}(k,u))}{\lambda} p_{ij}(u) \varrho_{ij}(k,u). \tag{27}$$

The value of $R(i,u)$ can easily be computed. Using (25), we have

$$m_{ij}(u) = \sum_{k=1}^{m} p_{ij}(u) \varrho_{ij}(k,u) \exp\left(-\lambda\tau_{ij}(k,u)\right) \tag{28}$$

so $m_{ij}(u)$ can also be computed. Having (27) and (28), we appropriately formulate the Bellman optimality equation (26).

2) Uniform Distribution: We assume that, given the expression status of other genes in the network, we can measure the maximum and the minimum duration of transcription of a specific gene. We postulate that the intertransition interval between two states can take any value within the range from the maximum value to the minimum value with an equal probability. The intertransition interval between two logical states i and j has a uniform distribution in the interval $[c_{ij}(u), d_{ij}(u)]$.

Using (22), we have

$$R(i,u) = \frac{r(i,u)}{\lambda} \times \sum_{j=1}^{|\mathcal{W}|} \left(1 - \frac{\exp(-\lambda c_{ij}(u)) - \exp(-\lambda d_{ij}(u))}{\lambda(d_{ij}(u) - c_{ij}(u))}\right) p_{ij}(u). \quad (29)$$

Again, the value of $R(i,u)$ can easily be computed. Using (25), we have

$$m_{ij}(u) = \frac{\exp(-\lambda c_{ij}(u)) - \exp(-\lambda d_{ij}(u))}{\lambda(d_{ij}(u) - c_{ij}(u))} p_{ij}(u) \quad (30)$$

so $m_{ij}(u)$ can also be computed. Having (29) and (30), we again appropriately formulate the Bellman optimality equation (26).

3) Exponential Distribution: The amount of data observed from a biological system is usually limited. Instead of using the data to estimate an arbitrary intertransition interval distribution, we can postulate a class of parametric distributions whose members can be defined with a single parameter, e.g., the expected value. Here, we assume the distribution of the intertransition interval follows an exponential distribution. If all the intertransition intervals of state i are exponentially distributed, then the sojourn time of state i possesses an exponential distribution

$$P_i(\tau, u) = 1 - e^{-\nu_i(u)\tau} \quad \tau \geq 0. \quad (31)$$

In (31), $\nu_i(u)$ is the rate of transition from state i whenever the control has value u. Practically, the rates $\nu_i(u)$ are bounded for all states $i \in \mathcal{W}$, and all controls $u \in \mathcal{C}$.

Assuming the distribution the intertransition interval is exponentially distributed, we use an alternative approach, termed uniformization, to derive the Bellman optimality equation (26). Uniformization speeds up transitions that are slow on the average by allowing fictitious transitions from a state to itself, so sometimes after a transition the state may stay unchanged [23]. In uniformization, a uniform transition rate ν is assigned to all the states. The uniform transition rate ν is selected such that $\nu_i(u) \leq \nu$ for all $i \in \mathcal{W}$ and $u \in \mathcal{C}$. Using the uniform transition rate ν, we define the set of new transition probabilities for each state of the uniformed SMDP by

$$\tilde{p}_{ij}(u) = \begin{cases} \frac{\nu_i(u)}{\nu} p_{ij}(u), & \text{if } i \neq j \\ \frac{\nu_i(u)}{\nu} p_{ii}(u) + 1 - \frac{\nu_i(u)}{\nu}, & \text{if } i = j. \end{cases} \quad (32)$$

It can be shown that leaving state i at a rate $\nu_i(u)$ in the original SMDP is statistically identical to leaving state i at the faster rate ν, but returning back to i with the probability $(1 - \nu_i(u)/\nu)$ in the uniformed SMDP with transition probabilities defined in (32).

Since states of the SMDP remain constant between epoch times, the expected total discounted reward in (19) can be expressed as

$$J_\pi(i) = \sum_{k=0}^{\infty} E\left\{ \int_{t_k}^{t_{k+1}} e^{-\lambda t} r(z(t_k), u(t_k))\, dt | z(t_0) = i \right\}. \quad (33)$$

Considering the memoryless property of the exponential distribution, we can express (33) as

$$J_\pi(i) = E\left\{ \sum_{k=0}^{\infty} \left(\frac{\nu}{\nu + \lambda}\right)^k \frac{r(z(t_k), u(t_k))}{\lambda + \nu} \right\}. \quad (34)$$

According to the latest form of the expected total discounted reward in (34), we can exploit the MDP results to determine the Bellman optimality transformation (26). The expected reward of a single transition is

$$R(i,u) = \frac{r(i,u)}{\lambda + \nu} \quad (35)$$

and the value of parameter $m_{ij}(u)$ is determined by

$$m_{ij}(u) = \frac{\nu}{\lambda + \nu} \tilde{p}_{ij}(u). \quad (36)$$

IV. Control of a Mammalian Cell Cycle Related Network

In this section, we present an SM-ARN that is a probabilistic version of the Boolean model of the mammalian cell cycle recently proposed in [17]. Using this Boolean model, we construct an SM-ARN that postulates the cell-cycles with mutated phenotype. The proposed intervention method is then applied to hinder the cell growth in the absence of growth factors.

During the late 1970s and early 1980s, yeast geneticists identified the cell-cycle genes encoding for new classes of molecules, including the cyclins (so-called because of their cyclic pattern of activation) and their cyclin dependent kinases (cdk) partners [17]. Our model is rooted in the work of Faure *et al.*, who have recently derived and analyzed the Boolean functions of the mammalian cell cycle [17]. The authors have been able to quantitatively reproduce the main known features of the wild-type biological system, as well as the consequences of several types of mutations.

Mammalian cell division is tightly controlled. In a growing mammal, the cell division should coordinate with the overall growth of the organism. This coordination is controlled via extracellular signals. These signals indicate whether a cell should divide or remain in a resting state. The positive signals, or growth factors, instigate the activation of Cyclin D (CycD) in the cell.

The key genes in this model are CycD, retinoblastoma (Rb), and p27. Rb is a tumor-suppressor gene. This gene is expressed in the absence of the cyclins, which inhibit the Rb by phosphorylation. Whenever p27 is present, Rb can also be expressed even in the presence of CycE or CycA. Gene p27 is active in the absence of the cyclins. Whenever p27 is present, it blocks the action of CycE or CycA. Hence, it stops the cell cycle.

The preceding explanation represents the wild-type cell-cycle model. Following one of the proposed mutations in [17], we assume p27 is mutated and its logical rule is always zero (OFF). In this cancerous scenario, p27 can never be activated. This mutation introduces a situation where both CycD and Rb might be

TABLE I
MUTATED BOOLEAN FUNCTIONS OF MAMMALIAN CELL CYCLE

Product	Predictors
$CycD$	Input
Rb	$(\overline{CycD} \wedge \overline{CycE} \wedge \overline{CycA} \wedge \overline{CycB})$
$E2F$	$(\overline{Rb} \wedge \overline{CycA} \wedge \overline{CycB})$
$CycE$	$(E2F \wedge \overline{Rb})$
$CycA$	$(E2F \wedge \overline{Rb} \wedge \overline{Cdc20} \wedge (\overline{Cdh1 \wedge Ubc})) \vee (CycA \wedge \overline{Rb} \wedge \overline{Cdc20} \wedge (\overline{Cdh1 \wedge Ubc}))$
$Cdc20$	$CycB$
$Cdh1$	$(\overline{CycA} \wedge \overline{CycB}) \vee (Cdc20)$
Ubc	$(\overline{Cdh1}) \vee (Cdh1 \wedge Ubc \wedge (Cdc20 \vee CycA \vee CycB))$
$CycB$	$(\overline{Cdc20} \wedge \overline{Cdh1})$

inactive. As a result, in this mutated phenotype, the cell cycles in the absence of any growth factor. In other words, we consider the logical states in which both Rb and CycD are down-regulated as *undesirable states*, when p27 is mutated. Table I summarizes the mutated Boolean functions.

The Boolean functions in Table I are used to construct the embedded-PBN of the cell-cycle's SM-ARN. The defined embedded-PBN maintains the topology of the oriented graph generated by the predictors in Table I. To this end, we assume that the extra-cellular signal to the cell-cycle model is a latent variable. The growth factor is not part of the cell and its value is determined by the surrounding cells. The expression of CycD changes independently of the cell's content and reflects the state of the growth factor. Depending on the expression status of CycD, we obtain two constituent Boolean networks for the embedded-PBN. The first constituent Boolean network is determined from Table I when the value of CycD is equal to zero. Similarly, the second constituent Boolean network is determined by setting the variable of CycD to one. To completely define the embedded-PBN, the switching probability, the perturbation probability, and the probability of selecting each constituent Boolean network have to be specified. We assume that these are known. Here, we set the switching probability and the perturbation probabilities equal to 10^{-2} and 10^{-3}, respectively, and the two constituent Boolean networks are equally likely.

We also have to specify the intertransition interval distributions between the logical states to fully define the cell-cycle's SM-ARN. Although such information is likely to be available in the near future, it is not available today. Here, we simply assume that all intertransition intervals between logical states are exponentially distributed. If $\tau(i,j)$ is the sojourn time in logical state i before transition to state j, then we need the rate of the transition from state i to state j to specify its distribution. We use the gene-expression data to determine the probability, p_{ij}, of the transition from state i to state j in the embedded-PBN. We assume that the rate of the transition from state i to state j is assigned such that

$$P\left\{\tau(i,j) < \min_{\substack{k=1,\ldots,|\mathcal{S}| \\ k \neq j}} \tau(i,k)\right\} = p_{ij}. \tag{37}$$

In other words, the probability of the first transition out of state i to state j is equal to the transition probability p_{ij}. The left side of (37) can be determined for exponentially distributed sojourn times.

According to Table I, the cell-cycle's SM-ARN consists of nine genes: CycD, Rb, E2F, CycE, CycA, Cdc20, Cdh1, UbcH10, and CycB. The above order of genes is used in the binary representation of the logical states, with CycD as the most significant bit and CycB as the least significant bit. This order of genes in the logical states facilitates the presentation of our results and does not affect the computed control policy.

Preventing the logical states with simultaneously down-regulated CycD and Rb as our objective, we apply the intervention method described in Section III to the constructed SM-ARN. Here we only consider a single control, u. If the control is high, $u = 1$, then the state of the control gene is reversed; if $u = 0$, then the state of the control gene remains unchanged. The control gene can be any one of the the genes in the model except CycD.

We assume that the reward of the logical states with down-regulated Rb and CycD is lower than those for the states in which these two genes are not simultaneously down-regulated. We also consider the cost of applying a control action, which reduces the reward of each logical state. We postulate the following rate-of-reward function:

$$r(i,u) = \begin{cases} 6, & \text{if } u = 0 \text{ and} \\ & \quad (\mathrm{CycD}, \mathrm{Rb}) \neq (0,0) \text{ for logical state } i \\ 1, & \text{if } u = 0 \text{ and} \\ & \quad (\mathrm{CycD}, \mathrm{Rb}) = (0,0) \text{ for logical state } i \\ 5, & \text{if } u = 1 \text{ and} \\ & \quad (\mathrm{CycD}, \mathrm{Rb}) \neq (0,0) \text{ for logical state } i \\ 0, & \text{if } u = 1 \text{ and} \\ & \quad (\mathrm{CycD}, \mathrm{Rb}) = (0,0) \text{ for logical state } i. \end{cases} \tag{38}$$

We select an arbitrary rate of reward; however, the reward and control cost are selected so that applying the control to prevent the undesirable logical states is preferable in comparison to not applying control and remaining in an undesirable state. In practice, the reward values have to capture the benefits and costs of the intervention and the relative preference of the states. They have to be set in conjunction with physicians according to their clinical judgement. Although this is not feasible within the domain of current medical practice, we do believe that such an approach will become increasingly mainstream once engineering approaches are demonstrated to yield significant benefits in translational medicine.

Assuming the preceding rate-of-reward function, we compute a control policy for the SM-ARN of the cell cycle. Fig. 5 depicts the fraction of time that the SM-ARN spends in each logical state when there is no intervention. Per Fig. 5, the aggregated fraction of time that the cell-cycle model spends in the logical states with simultaneously down-regulated CycD and Rb is significant. In the long run, the model spends 49% of its time in the undesirable states.

We define ΔP to be the percentage of the change in the fraction of time that the SM-ARN spends in the logical states with simultaneously down-regulated CycD and Rb before and after the intervention. As a performance measure, ΔP indicates the percentage of the reduction in the fraction of time that the model spends in undesirable logical states in the long run.

If we assume that we can alter the expression level of any gene in the network as a therapeutic method, then it is natural to ask which gene should be used as a control gene to alter the behavior of the model. To this end, we find an intervention policy for each of the genes in the network using the intervention method

TABLE II
ΔP FOR THE INTERVENTION STRATEGY BASED ON VARIOUS CONTROL GENES

Control Gene	Rb	E2F	CycE	CycA	Cdc20	Cdh1	UbcH10	CycB
ΔP	94.2%	89.1%	71.1%	62.1%	63.5%	68.4%	59.7%	75.2%

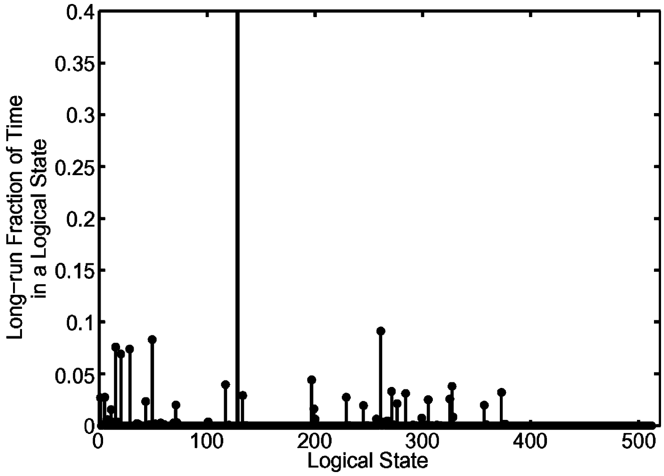

Fig. 5. Fraction of time that the SM-ARN of mammalian cell cycle spends in each logical state prior to intervention. The vertical line separates the undesirable logical states from the desirable logical states.

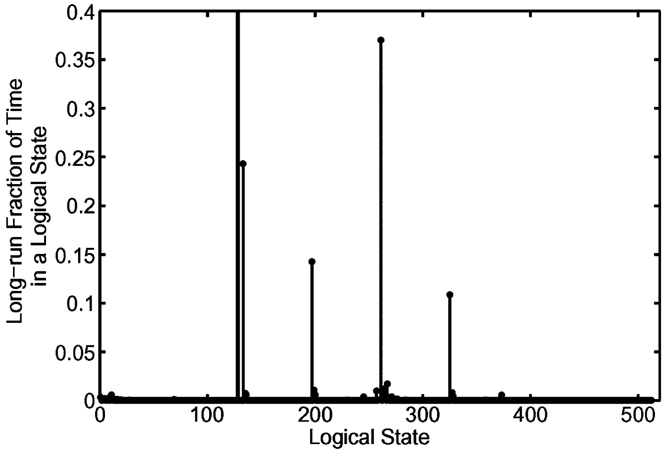

Fig. 6. Fraction of time that the SM-ARN of mammalian cell cycle spends in logical states after intervention using Rb as the control gene. The vertical line separates the undesirable logical states from the desirable logical states.

explained in Section III. Table II lists the value of ΔP corresponding to each gene in the network. Among all the genes, Rb and E2F have the best performance.

After the intervention in the SM-ARN based on the control policy designed for Rb, the fraction of time that the model spends in the undesirable logical states is significantly altered. Directly using Rb as the control gene, we can reduce the fraction of time that the model spends in the undesirable states to less than 2%.

If the direct control of Rb is not feasible, then one can use E2F as the control gene. In this case, the system spends slightly more time in the undesirable states, but still less than 4.5%. The difference between the performances of Rb and E2F is somewhat insignificant. Fig. 7 depicts the percentage of time that the SM-ARN spends in each logical state after intervention when the control strategy is designed based on E2F.

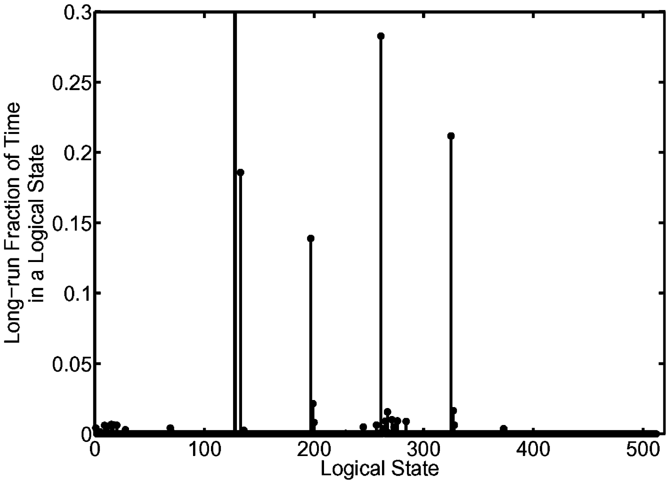

Fig. 7. Fraction of time that the SM-ARN of mammalian cell cycle spends in logical states after intervention using E2F as the control gene. The vertical line separates the undesirable logical states from the desirable logical states.

V. Conclusion

We have formulated the design of optimal intervention strategies for two proposed asynchronous regulatory network models with discrete state spaces. The DA-PBN model extends the benefits of context-sensitive PBNs by adding the ability to cope with temporal context as well as structural context. Since asynchronism at the node level has practical limitations, we introduce the SM-ARN model, in which the asynchronism is at the logical-state level. Empirically measurable timing information of biological systems can be directly incorporated into the SM-ARN model to determine the time-delay distributions between transitions from one logical state to another logical state. Using the SM-ARN model, we have modeled the dynamics of a mammalian cell cycle regulatory network. The proposed intervention method for the SM-ARN is then used to design a strategy to influence the dynamics of the SM-ARN constructed for the mammalian cell cycle. The goal of the intervention is to reduce the long run likelihood of undesirable cell growth. The presented numerical studies strongly suggest that our intervention method effectively alters the dynamics of the cell cycle model.

References

[1] A. Datta and E. R. Dougherty, *Introduction to Genomic Signal Processing With Control*. Boca Raton, FL: CRC, 2006.

[2] I. Shmulevich and E. R. Dougherty, *Genomic Signal Processing*. Princeton, NJ: Princeton Univ. Press, 2007.

[3] I. Shmulevich, E. R. Dougherty, S. Kim, and W. Zhang, “Probabilistic Boolean networks: A rule-based uncertainty model for gene regulatory networks,” *Bioinformatics*, vol. 18, no. 2, pp. 261–274, 2002.

[4] I. Shmulevich, E. R. Dougherty, and W. Zhang, "Gene perturbation and intervention in probabilistic Boolean networks," *Bioinformatics*, vol. 18, no. 10, pp. 1319–1331, 2002.

[5] A. Datta, A. Choudhary, M. L. Bittner, and E. R. Dougherty, "External control in markovian genetic regulatory networks," *Mach. Learn.*, vol. 52, no. 1/2, pp. 169–191, 2003.

[6] R. Pal, A. Datta, and E. R. Dougherty, "Optimal infinite-horizon control for probabilistic Boolean networks," *IEEE Trans. Signal Process.*, vol. 54, no. 6, pp. 2375–2387, Jun. 2006.

[7] S. Huang, "Gene expression profiling, genetic networks, and cellular states: An integrating concept for tumorigenesis and drug discovery," *J. Mol. Med.*, vol. 77, no. 6, pp. 469–480, 1999.

[8] R. Thomas, *Kinetic Logic: A Boolean Approach to the Analysis of Complex Regulatory Systems*. New York: Springer-Verlag, 1979.

[9] I. Harvey and T. Bossomaier, "Time out of joint: Attractors in asynchronous random Boolean networks," in *Proc. 4th Eur. Conf. Artificial Life*, Jul. 1997, pp. 67–75.

[10] X. Deng, H. Geng, and M. T. Matache, "Dynamics of asynchronous random Boolean networks with asynchrony generated by stochastic processes," *BioSystems*, vol. 88, no. 1–2, pp. 16–34, 2007.

[11] F. Greil and B. Drossel, "The dynamics of critical Kauffman networks under asynchronous stochastic update," *Phys. Rev. Lett.*, vol. 95, no. 4(048701), 2005.

[12] C. Gershenson, "Classification of random Boolean networks," in *Proc. 8th Int. Conf. Artificial Life (Artificial Life VIII)*, Sydney, Australia, Dec. 2002, pp. 1–8.

[13] E. A. Di Paolo, "Rhythmic and non-rhythmic attractors in asynchronous random Boolean networks," *BioSystems*, vol. 59, no. 3, pp. 185–195, 2001.

[14] E. A. Di Paolo, "Searching for rhythms in asynchronous Boolean networks," in *Proc. 7th Int. Conf. Artificial Life (Artificial Life VII)*, Portland, OR, Aug. 2000, pp. 1–6.

[15] D. Cornforth, D. G. Green, D. Newth, and M. R. Kirley, "Ordered asynchronous processes in natural and artificial systems," in *Proc. 5th Australia-Japan Joint Workshop on Intelligent and Evolutionary Systems*, New Zealand, Nov. 2001, pp. 105–112.

[16] M. Chaves, E. D. Sontag, and R. Albert, "Methods of robustness analysis for Boolean models of gene control networks," *IEE Proc. Systems Biology*, vol. 235, pp. 154–167, 2006.

[17] A. Faure, A. Naldi, C. Chaouiya, and D. Theiffry, "Dynamical analysis of a generic Boolean model for the control of the mammalian cell cycle," *Bioinformatics*, vol. 22, no. 14, pp. e124–e131, 2006.

[18] B. Mesot and C. Teuscher, "Critical values in asynchronous random Boolean networks," in *Proc. 7th Eur. Conf. Artificial Life (ECAL03)*, Sep. 2003, pp. 367–377, MIT Press.

[19] M. Chaves, R. Albert, and E. D. Sontag, "Robustness and fragility of Boolean models for genetic regulatory networks," *J. Theoret. Biol.*, vol. 235, pp. 431–449, 2005.

[20] S. Kim, H. Li, E. R. Dougherty, N. W. Cao, Y. D. Chen, M. L. Bittner, and E. B. Suh, "Can Markov chain models mimic biological regulation?," *J. Biol. Syst.*, vol. 10, no. 4, pp. 337–357, 2002.

[21] M. Brun, E. R. Dougherty, and I. Shmulevich, "Steady-state probabilities for attractors in probabilistic Boolean networks," *Signal Process.*, vol. 85, no. 10, pp. 1993–2013, 2005.

[22] J. R. Norris, *Markov Chains*. Cambridge, U.K.: Cambridge Univ. Press, 1998.

[23] D. P. Bertsekas, *Dynamic Programming and Optimal Control*. New York: Athena, 2001.

[24] R. G. Gallager, *Discrete Stochastic Processes*. Norwell, MA: Kluwer, 1995.

Babak Faryabi (S'03) received the M.Sc. degree in electrical and computer engineering from the Sharif University of Technology, Iran, in 1997. He is currently pursuing the Ph.D. degree at Texas A&M University, College Station.

Prior to joining the Texas A&M University, he was a Research Assistant at the University of Toronto, Toronto, ON, Canada. He was the recipient of the Edward S. Rogers Scholarship from the Department of Electrical and Computer Engineering, University of Toronto, from 2003 to 2005. His research interests lie in the general area of systems biology. Specifically, he is interested in the applications of Merkovian decision processes and their approximations in the modeling and control of gene regulatory networks.

Dr. Faryabi is a member of the International Society for Computational Biology.

Jean-François Chamberland (S'98–M'04) received the B.Eng. degree from McGill University, Montreal, QC, Canada, in 1998, the M.Sc. degree from Cornell University, Ithaca, NY, in 2000, and the Ph.D. degree from the University of Illinois at Urbana-Champaign, Urbana, in 2004, all in electrical engineering.

He joined Texas A&M University, College Station, in 2004, where he is currently an Assistant Professor in the Department of Electrical and Computer Engineering. His research interests include communication systems, queueing theory, detection and estimation, and genomic signal processing.

Dr. Chamberland was the recipient of a Young Author Best Paper Award from the IEEE Signal Processing Society in 2006.

Golnaz Vahedi (S'02) received the B.Sc. degree in electrical engineering from the Sharif University of Technology, Iran, in 2001, and the M.Sc. degree in electrical and computer engineering from University of Alberta, Edmonton, AB, Canada, in 2004. She is currently pursuing the Ph.D. degree in electrical and computer engineering at Texas A&M University, College Station.

Her research interests includes systems biology, genomic signal processing, and inference and control of genetic regulatory networks.

Aniruddha Datta (S'87–M'91–SM'97) received the B.Tech. degree in electrical engineering from the Indian Institute of Technology, Kharagpur, in 1985, the M.S.E.E. degree from Southern Illinois University, Carbondale, in 1987, and the M.S. degree in applied mathematics and the Ph.D. degree from the University of Southern California, Los Angeles, in 1991.

In August 1991, he joined the Department of Electrical Engineering at Texas A&M University, College Station, where he is currently a Professor and holder of the J. W. Runyon, Jr. '35 Professorship II. His areas of interest include adaptive control, robust control, PID control, and genomic signal processing. From 2001 to 2003, he worked on Cancer research as a postdoctoral trainee under a National Cancer Institute (NCI) Training Grant. He is the author of the book *Adaptive Internal Model Control* (Springer-Verlag, 1998), a coauthor of the book *Structure and Synthesis of PID Controllers* (Springer-Verlag, 2000), a coauthor of the book *PID Controllers for Time Delay Systems* (Birkhauser, 2004), and a coauthor of the book *Introduction to Genomic Signal Processing with Control* (CRC, 2007).

Dr. Datta served as an Associate Editor of the IEEE TRANSACTIONS ON AUTOMATIC CONTROL from 2001–2003 and the IEEE TRANSACTIONS ON SYSTEMS, MAN AND CYBERNETICS—PART B, CYBERNETICS from 2005–2006.

Edward R. Dougherty (M'05) received the M.S. degree in computer science from Stevens Institute of Technology, Hoboken, NJ, and the Ph.D. degree in mathematics from Rutgers University, Piscataway, NJ.

He is a Professor in the Department of Electrical Engineering at Texas A&M University, College Station, where he holds the Robert M. Kennedy Chair and is Director of the Genomic Signal Processing Laboratory. He is also the Director of the Computational Biology Division of the Translational Genomics Research Institute, Phoenix, AZ. At Texas A&M, he received the Association of Former Students Distinguished Achievement Award in Research, was named Fellow of the Texas Engineering Experiment Station, and was named Halliburton Professor of the Dwight Look College of Engineering. He is the author of 14 books, editor of five others, and author of more than 200 journal papers. He has contributed extensively to the statistical design of nonlinear operators for image processing and the consequent application of pattern recognition theory to nonlinear image processing. His research in genomic signal processing is aimed at diagnosis and prognosis based on genetic signatures and using gene regulatory networks to develop therapies based on the disruption or mitigation of aberrant gene function contributing to the pathology of a disease.

Dr. Dougherty was awarded the Doctor Honoris Causa by the Tampere University of Technology, Finland. He is a fellow of SPIE, has received the SPIE President's Award, and served as the Editor of the SPIE/IS&T *Journal of Electronic Imaging*.