This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI10.1109/TAC.2021.3106896, IEEE Transactions on Automatic Control.

Model Evaluation of the Stochastic Boolean Control Networks

Hongwei Chen, Zidong Wang, Bo Shen and Jinling Liang

***Abstract*—This paper investigates the model evaluation problem for the stochastic Boolean control networks (SBCNs). First, an algebraic expression of the SBCN is obtained based on the semi-tensor product method, and a straightforward approach is then proposed to compute the probability that the given observed output sequence is produced by the considered model. Second, two recursive algorithms, namely the forward and the backward algorithms, are designed for model evaluation by resorting to the forward-backward technique. In addition, scaling factors are introduced to deal with the numerical issues arising in the implementation of the developed algorithms. Finally, to illustrate the applicability and effectiveness of the proposed algorithms, a Boolean model of the *lac* operon is employed as an example for numerical simulation.**

***Index Terms*—Stochastic Boolean control networks, model evaluation, forward-backward technique, scaling factor, semi-tensor product.**

I. Introduction

Boolean networks (BNs) were originally introduced by Kauffman in the late 1960s for simulating and analyzing the dynamic behaviors of genes in GRNs [16]. A BN can be described by a directed graph, where the state of a node is restricted to be 0 or 1 (indicating inactive or active, respectively) and is governed by a Boolean function which depends on the states of the other nodes with edges directed to this node. The concept of BN can be extended to Boolean control network (BCN) by introducing certain external stimuli as control input of the network. Recently, the development of algebraic state space representation (ASSR) approach has aroused a significant amount of new research interest in BNs, and several challenging problems have been configured in the ASSR framework based on the semi-tensor product (STP) of matrices. Examples of such problems include, but are not limited to, stability and stabilization [22], [26], disturbance decoupling [7], [25], controllability and observability [24], [32], optimal control [13], [17], and BN synchronization problems [5], [23].

It has been well recognized that the gene regulation processes are inherently stochastic due primarily to the random variation of some key signaling proteins and the probabilistic biochemical reactions [27], [31]. Experimental results have further confirmed that the transitions between states of a GRN do occur in a random fashion [28]. As such, theoretical models without accounting for intrinsic/extrinsic stochastic noises might not be able to accurately reflect the real dynamics of the biological GRNs [12]. On the other hand, the recent work in [6] has shown the benefits of monitoring the stochastic signals/responses in real time for capturing the true cell dynamics behind the statistical averages. Therefore, the stochastic noise should be taken into consideration when simulating the evolution behaviors of the genes in GRNs by BNs [33] and, so far, many formal types of uncertain BNs have been proposed as a generalization of the BNs with examples including the stochastic BNs (SBNs), the probabilistic BNs and the BNs with stochastic perturbations.

The problem of identifying a BN from the time-series gene expression data is extremely important in the field of systems biology since it helps gaining the deep insight into the regulatory mechanisms among genes. For example, the identification of the behaviors of p53-MDM2 negative-feedback GRN plays an important role in the therapy of various types of human cancers including the leukemia and the breast cancer [11]. In the past two decades, identification of the deterministic/stochastic BNs with the available data has been a topic under extensive investigation, and many effective approaches have been proposed in the literature such as algebraic method [18], [29], Best-Fit Extension [19] and Bayesian inference [15], and so on. It is noteworthy that these identification methods for BNs are based on certain developed algorithms, thereby providing *sufficient* conditions only. In [8], a novel method has been proposed for identifying the BNs from the experimental data by resorting to the STP technique. In this approach, only the transition matrix of the network needs to be identified, and this renders certain simplicity as compared to the way of directly identifying the logic equations. However, as the size of the corresponding network increases, dimension of the transition matrix will grow exponentially, which infers that the STP-based approach has an additional cost due to its exponential intrinsic nature. Recently, such an approach has been further utilized to deal with the identification of different types of BNs [1], [10], [21].

The stochastic dynamic behaviors of the GRNs, which have been validated by the aforementioned experimental/theoretical results, make it rather difficult to identify the accurate Boolean model according to the observed data. Actually, due to the limited observations and the exponential growth of the network scales, the precise identification of a large-scale becomes an almost impossible mission [19], [29], not to mention the identification of a stochastic BCN (SBCN) capturing the stochastic behaviors of the gene regulatory processes. On the other hand, when identifying a BN from the given observed sequence by using various well-known identification methods, we might be able to infer different model configurations of the BNs [1]. Consequently, for the given observed output sequence and an identified BN, it is ultimately important to calculate the probability that the observed sequence would be produced by the BN, which is referred to as the *model evaluation* problem.

The model evaluation problem of SBCNs investigated in this paper can be understood as a certain scoring/evaluating problem on how well an identified BN matches the given observed sequence. More specifically, for given model parameters, control input as well as observation sequences, the purpose of the model evaluation problem is to compute the exact probability that the observed sequence can

This work was supported in part by the National Key Research and Development Program of China under Grant 2018AAA0100202, the National Natural Science Foundation of China under Grants 62003083, 61873059, 61873148, 61922024 and 61933007, the Shanghai Science and Technology Program of China under Grant 20JC1414500, the Shanghai Sailing Program of China under Grant 19YF1402400, the Program of Shanghai Academic/Technology Research Leader of China under Grant 20XD1420100, the Royal Society of the UK, and the Alexander von Humboldt Foundation of Germany. *(Corresponding author: Bo Shen.)*

H. Chen and B. Shen are with the College of Information Science and Technology, Donghua University, Shanghai 201620, China, and are also with the Engineering Research Center of Digitalized Textile and Fashion Technology, Ministry of Education, Shanghai 201620, China. (e-mail: hongwei@dhu.edu.cn; bo.shen@dhu.edu.cn)

Z. Wang is with the Department of Computer Science, Brunel University London, Uxbridge, Middlesex UB8 3PH, U.K. (e-mail: Zidong.Wang@brunel.ac.uk)

J. Liang is with the School of Mathematics, Southeast University, Nanjing 210096, China. (e-mail: jinlliang@seu.edu.cn)

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI10.1109/TAC.2021.3106896, IEEE Transactions on Automatic Control.

be produced by the identified model. As a matter of fact, the model evaluation problem and the fault diagnosis problem (see [2] and [3]) have the common objective of estimating certain probability so as to determine whether a given model is appropriate or not. Under the assumption that all the possible system faults are known and can be modeled, a fault diagnosis algorithm has been designed in the interesting paper [2], based on which the fault can be singled out as the candidate with the largest posterior probability for the given observation sequence. Recently, the fault detection and diagnosis problem for BNs has been investigated based on the STP technique and many novel results have been presented in the literature, see e.g. [14], [20].

To the best of the authors' knowledge, the model evaluation problem of BNs is still open and remains challenging. The main motivation of our current investigation is to shorten such a gap by developing some efficient algorithms. *The main technical contributions of this paper can be highlighted in threefold as follows. 1) A new yet effective approach is proposed for model evaluation of BNs by referring to the algebraic representation of the SBCNs. 2) With the help of the forward-backward technique, two recursive STP-based algorithms, namely the forward algorithm and the backward algorithm, are designed to compute the probability for the given observed output sequence to be produced by the model. 3) Scaling factors are introduced to cope with the numerical issues arising in the implementation of the developed algorithm.*

The remainder of this paper is organized as follows. Section II provides some necessary preliminaries on the definition of STP as well as the problem to be addressed. Section III presents the main results on the model evaluation of the SBCNs. Section IV demonstrates the feasibility of the proposed algorithms with a biological example. Section V concludes this paper.

II. Problem Formulation and Preliminaries

A vector is viewed as a column vector except where otherwise clearly stated. The i-th component of a vector x is denoted by $[x]_i$, and $[A]_{i,j}$ means the entry of matrix A in the i-th row and the j-th column. $\mathbb{R}^{m\times n}$ stands for the set of all $m\times n$ real matrices. The superscript "$\top$" represents the transpose for real matrices. δ_m^i means the i-th column of the m-dimensional identity matrix I_m. The delta set $\{\delta_m^i | i=1,2,\dots,m\}$ is simply denoted by Δ_m. A matrix $D\in\mathbb{R}^{m\times n}$ is called a logical matrix if $D=\left[\delta_m^{i_1}\ \delta_m^{i_2}\ \cdots\ \delta_m^{i_n}\right]$ $(i_1,i_2,\dots,i_n\in\{1,2,\dots,m\})$, which can also be represented as $D=\delta_m[i_1,i_2,\dots,i_n]$ for convenience. The set of all $m\times n$ logical matrices is denoted by $\mathcal{L}_{m\times n}$. $B(n,p)$ stands for the binomial distribution with parameters $n\in\mathbb{N}^+$ and $p\in[0,1]$, and the Bernoulli distribution with success probability p is simply denoted by $B(1,p)$. $\mathbb{E}\{\cdot\}$ means the expectation of a stochastic variable "$\cdot$". $N(\mu,\sigma^2)$ means the normal distribution with mean μ and variance σ^2.

A. Mathematical Preliminaries

Firstly, the definition of STP is presented, which is useful in our later discussion.

Definition 1: [9] For matrices $B_1\in\mathbb{R}^{q\times p}$ and $B_2\in\mathbb{R}^{n\times m}$, their STP is defined as follows:

$$B_1\ltimes B_2\triangleq(B_1\otimes I_{\alpha/p})(B_2\otimes I_{\alpha/n})$$

where α is the least common multiple of p and n.

Remark 1: Hereafter, the STP is simply called "product" and the symbol "$\ltimes$" is omitted if no confusion arises.

Next, with some abuse of notation, the logical variable $X\in\mathcal{B}\triangleq\{1,0\}$ is identified with the vector $x=\delta_2^{2-X}\in\Delta_2$. Throughout the paper, *logical variables are always denoted by upper case letters and their vector forms are denoted by lower ones if no confusion arises*. It is easy to verify that there is a bijective mapping between $\mathcal{B}$ and Δ_2. Moreover, such a mapping can be generalized to the bijection between $\mathcal{B}^n$ and Δ_N with $N=2^n$ by virtue of the STP. Specifically, the vector $X\triangleq(X_1,X_2,\dots,X_n)^\top\in\mathcal{B}^n$ is regarded completely as the column vector $x\triangleq\ltimes_{i=1}^n x_i\in\Delta_N$ with x_i representing the vector form of $X_i\in\mathcal{B}$. With this bijection in mind, an instrumental representation of the logical functions, needed in the sequel to derive the algebraic representation of the SBCNs, is given in the following lemma.

Lemma 1: [9] Let $g(x_1,x_2,\dots,x_n):\ \mathcal{B}^n\to\mathcal{B}$ be a logical function. Then, there exists a unique matrix $M_g\in\mathcal{L}_{2\times N}$, called *the structure matrix of* g, such that

$$g(x_1,x_2,\dots,x_n)=M_g\ltimes_{i=1}^n x_i,\quad x_i\in\Delta_2.$$

Finally, we introduce the concepts of random logical matrix as well as random logical variable.

Definition 2: Let $\mathcal{L}^\kappa_{m\times n}=\{L^{(1)},L^{(2)},\dots,L^{(\kappa)}\}$ with $L^{(1)}$, $L^{(2)}$, …, $L^{(\kappa)}\in\mathcal{L}_{m\times n}$. A logical matrix L is called *a κ-valued random logical matrix* if L takes value in the set $\mathcal{L}^\kappa_{m\times n}$ and $L=L^{(i)}$ with probability $\mathbb{P}_i$ $(i=1,2,\dots,\kappa)$ satisfying $\sum_{i=1}^\kappa\mathbb{P}_i=1$. For the particular case $n=1$, that is, $L\in\{\delta_m^{i_1},\delta_m^{i_2},\dots,\delta_m^{i_\kappa}\}$ with $i_1,i_2,\dots,i_\kappa\in\{1,2,\dots,m\}$, we call it *a κ-valued random logical variable*.

B. SBCNs

The simplest yet basic modeling method for the real regulatory networks is to rely on the discrete logic-based system, which facilitates our study by using the following canonical stochastic, time-invariant Boolean dynamics:

$$X_i(k+1)=f_i\big(X(k),U(k),W(k)\big),\ i=1,2,\dots,n \tag{1a}$$

where vector $X(k)\triangleq(X_1(k),\dots,X_n(k))^\top\in\mathcal{B}^n$ captures the state of each entity in a regulatory network (for example, genes, proteins and small molecules) at discrete time $k\in\mathbb{N}$. The logical function f_i describes the regulatory relationships among genes, which gives the next state of the regulated gene i by combining its regulators' current states. The system is controlled by the input vector $U(k)\triangleq(U_1(k),\dots,U_m(k))^\top\in\mathcal{B}^m$ which is assumed to be deterministic and known. Finally, $W(k)\triangleq(W_1(k),\dots,W_{l_1}(k))^\top\in\mathcal{B}^{l_1}$ is the system noise that reflects the inherent stochastic characteristic of the gene expression process. Without loss of generality, we assume that the system noise obeys the Bernoulli distribution, i.e., $W_i(k)\sim B(1,p_i)$ for $i=1,2,\dots,l_1$.

In the Boolean model, there are two possible states for the abundance of the transcript: high (active gene) and low (inactive gene). Accordingly, the Boolean-value measurement relating to the directly unobservable hidden state variables is described by the following observation equation:

$$Y_j(k)=h_j\big(X(k),V(k)\big),\quad j=1,2,\dots,q \tag{1b}$$

where $Y(k)\triangleq(Y_1(k),\dots,Y_q(k))^\top\in\mathcal{B}^q$ $(q\le n)$ represents the observation at discrete time k; $V(k)\triangleq(V_1(k),\dots,V_{l_2}(k))^\top\in\mathcal{B}^{l_2}$ is the independent observation noise with $V_j(k)\sim B(1,q_j)$ for $j=1,2,\dots,l_2$; and $h_j(\cdot,\cdot)$ is the Boolean function mapping the current state $X(k)$ and the observation noise $V(k)$ into the measurement space. Moreover, the noises $W(k)$ and $V(k)$ are assumed to be mutually independent and white in the sense that the noises at any distinct time instants are independent random variables. We further assume that the noises are independent with the initial state variable.

In what follows, by resorting to Lemma 1, system (1) is converted into an algebraic form, which simplify the network inference problem

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI10.1109/TAC.2021.3106896, IEEE Transactions on Automatic Control.

significantly. For each Boolean function f_i (respectively, h_j), one can find its structure matrix $\mathcal{F}_i$ (respectively, $\mathcal{H}_j$), thereby giving the following component-wise algebraic form of system (1):

$$x_i(k+1) = \mathcal{F}_i w(k)u(k)x(k)$$
$$y_j(k) = \mathcal{H}_j v(k)x(k)$$

where $u(k) = \ltimes_{i=1}^{m} u_i(k) \in \Delta_M$ with $M \triangleq 2^m$; $w(k) = \ltimes_{i=1}^{l_1} w_i(k)$ and $v(k) = \ltimes_{i=1}^{l_2} v_i(k)$ are respectively L_1-valued and L_2-valued random logical variables with $L_1 \triangleq 2^{l_1}$ and $L_2 \triangleq 2^{l_2}$. By setting $y(k) = \ltimes_{j=1}^{q} y_j(k) \in \Delta_Q$ with $Q \triangleq 2^q$, the ASSR of system (1) is further obtained as follows:

$$x(k+1) = \mathcal{F} w(k)u(k)x(k) \tag{2a}$$
$$y(k) = \mathcal{H} v(k)x(k) \tag{2b}$$

where $\mathcal{F} = \mathcal{F}_1 \ltimes_{i=2}^{n} [(I_{L_1 MN} \otimes \mathcal{F}_i)\Phi_{l_1+m+n}] \in \mathcal{L}_{N\times(L_1 MN)}$ and $\mathcal{H} = \mathcal{H}_1 \ltimes_{j=2}^{q} [(I_{L_2 N} \otimes \mathcal{H}_j)]\Phi_{l_2+n} \in \mathcal{L}_{Q\times(L_2 N)}$, in which Φ_{l_1+m+n} and Φ_{l_2+n} are the group power reducing matrices. For more details, please refer to [9].

C. Problem Formulation

Given served competing SBCNs in the form of (1), a fundamental problem that must be solved is to choose the proper model, which best matches the known observations among these competing ones, so that it is chosen/utilized in the real-world applications. For convenience, the complete parameters set of model (2) is denoted by $\theta = (\mathcal{F}, \mathcal{H}, \rho, \pi)$, where $\rho \triangleq \{p_1 \ldots, p_{l_1}, q_1, \ldots, q_{l_2}\}$ represents the noise parameters set, and vector $\pi = (\pi_1, \pi_2, \ldots, \pi_N)^\top$ describes the initial distribution of the state with $[\pi]_{i_0} \triangleq \Pr\{x(0) = \delta_N^{i_0}\}$. The maximum-likelihood estimation of θ is given by

$$\hat{\theta} = \arg\max_{\theta\in\Theta} \Pr\{y_{0:K}|\theta, u_{0:K-1}\} \tag{3}$$

where $\Theta = \{\theta_1, \theta_2, \ldots, \theta_\ell\}$ and θ_i stands for the parameter of the i-th candidate model, in which $i = 1, 2, \ldots, \ell$ and ℓ is the number of the total candidate models for the regulatory network; $y_{0:K} \triangleq (y(0), y(1), \ldots, y(K)) = (\delta_Q^{j_0}, \delta_Q^{j_1}, \ldots, \delta_Q^{j_K})$ represents the entire observed/known output sequence with K being the terminal time instant; and $u_{0:K-1} \triangleq (u(0), u(1), \ldots, u(K-1)) = (\delta_M^{\mu_0}, \delta_M^{\mu_1}, \ldots, \delta_M^{\mu_{K-1}})$ indicates the given control input sequence.

Now, we are in a position to compute the probability of the observation sequence $y_{0:K}$ for the given parameter θ and the control input sequence $u_{0:K-1}$. More specifically, given $u_{0:K-1}$ and θ, due to the randomness characteristic of the system/observation noises as well as the stochastic initial state, the corresponding output sequence cannot be uniquely determined which, of course, includes the known observation sequence $y_{0:K}$ as a special case.

What we are interested is the proportion/probability of $y_{0:K}$ in all these possible observations, which can be formulated as follows:

Problem 1 (The Model Evaluation Problem): Given the observation sequence $y_{0:K}$, the model parameter θ and the corresponding control sequence $u_{0:K-1}$, this problem asks for computing the exact probability that the observed sequence is produced just by that model, i.e., $\Pr\{y_{0:K}|\theta, u_{0:K-1}\}$.

Remark 2: From the models' point of view, Problem 1 can also be regarded as the scoring problem as how suitable a given model matches an observed output sequence, which is extremely useful in the process of the system modeling. For instance, if we consider the case in which we are attempting to choose among several competing models, the solution to this problem allows us to single out the optimum model which best matches the known observations.

III. Main Results

In this section, the above-formulated problem will be solved. Firstly, a straightforward approach is proposed in subsection III-A to compute the probability of the observation sequence $y_{0:K}$ for the given parameter θ and the control input sequence $u_{0:K-1}$. Then, in subsection III-B, two recursive STP-based algorithms are further designed for model evaluation by resorting to the forward-backward technique, which will greatly decrease the computational complexity. Finally, scaling factors are introduced in subsection III-C to deal with the numerical precision issues arising in the implementation of the developed algorithms.

A. A Straightforward Approach

Based on the state space representation of the SBCN, we can obtain a simple algebraic expression for $\Pr\{y_{0:K}|\theta, u_{0:K-1}\}$. For this purpose, we first give an auxiliary result whose proof is straightforward based on the ASSR formulation and hence omitted here for brevity.

Lemma 2: Let (2) be the ASSR of system (1).

i) For the given control input $u(k) = \delta_M^{\mu_k}$, the one-step transition probability from $x(k) = \delta_N^{i_k}$ to $x(k+1) = \delta_N^{i_{k+1}}$ is

$$\begin{aligned}&\Pr\{x(k+1) = \delta_N^{i_{k+1}}|x(k) = \delta_N^{i_k}, \theta, u(k) = \delta_M^{\mu_k}\}\\ &= \left[F \ltimes \delta_M^{\mu_k}\right]_{i_{k+1}, i_k}\end{aligned} \tag{4}$$

where $F \triangleq \mathcal{F} \ltimes (\ltimes_{i=1}^{l_1} \begin{bmatrix} p_i & 1-p_i \end{bmatrix}^\top)$.

ii) For each of the N possible states, the corresponding emission probability[1] can be computed as

$$\Pr\{y(k) = \delta_Q^{j_k} \Big| x(k) = \delta_N^{i_k}, \theta\} = \left[H\right]_{j_k, i_k} \tag{5}$$

where $H \triangleq \mathcal{H} \ltimes (\ltimes_{i=1}^{l_2} \begin{bmatrix} q_i & 1-q_i \end{bmatrix}^\top)$.

Remark 3: For the given $x(k) \in \Delta_N$, there is a set of emission probabilities governing the distribution of the observation $y(k)$. The size/cardinal of this set is determined by the nature of the observed variable $y(k)$. In this paper, it is discrete with Q possible values.

Theorem 1: For the given SBCN (2) with corresponding control sequence $u_{0:K-1}$, the probability of the observation sequence $y_{0:K}$ can be calculated as

$$\begin{aligned}&\Pr\{y_{0:K}|\theta, u_{0:K-1}\}\\ &= \sum_{i_0, i_1, \ldots, i_K} \pi_{i_0} \prod_{k=1}^{K} \left[F \ltimes \delta_M^{\mu_{k-1}}\right]_{i_k, i_{k-1}} \prod_{k=0}^{K} \left[H\right]_{j_k, i_k}\end{aligned} \tag{6}$$

where $i_0, i_1, \ldots, i_K \in \{1, 2, \ldots, N\}$.

Proof: Consider an arbitrary fixed state sequence of length K as $x_{0:K} \triangleq (x(0), x(1), \ldots, x(K)) = (\delta_N^{i_0}, \delta_N^{i_1}, \ldots, \delta_N^{i_K})$. According to i) of Lemma 2, the probability of such a state sequence can be derived as follows:

$$\begin{aligned}&\Pr\{x_{0:K}|\theta, u_{0:K-1}\}\\ &= \Pr\{x(K)|x_{0:K-1}, \theta, u_{0:K-1}\} \cdot \Pr\{x_{0:K-1}|\theta, u_{0:K-1}\}\\ &= \cdots\\ &= \pi_{i_0} \prod_{k=1}^{K} \left[F \ltimes \delta_M^{\mu_{k-1}}\right]_{i_k, i_{k-1}}.\end{aligned}$$

Based on the statistical independence of the observations, we can use ii) of Lemma 2 to obtain $\Pr\{y_{0:K}|x_{0:K}, \theta\} = \prod_{k=0}^{K} [H]_{j_k, i_k}$. The probability that $y_{0:K}$ and $x_{0:K}$ occur simultaneously, namely, the joint probability of $y_{0:K}$ and $x_{0:K}$, can be computed as

$$\Pr\{y_{0:K}, x_{0:K}|\theta, u_{0:K-1}\}$$

[1]Emission probability is the probability that a particular measurable state can be observed, provided that system (2) is in one of the hidden states.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI10.1109/TAC.2021.3106896, IEEE Transactions on Automatic Control.

$$= \Pr\{y_{0:K}|x_{0:K},\theta\}\cdot\Pr\{x_{0:K}|\theta,u_{0:K-1}\}$$
$$= \pi_{i_0}\prod_{k=1}^{K}\left[F\ltimes\delta_M^{\mu_{k-1}}\right]_{i_k,i_{k-1}}\prod_{k=0}^{K}\left[H\right]_{j_k,i_k}. \quad (7)$$

For the given model parameter θ and the corresponding control sequence $u_{0:K-1}$, calculating the probability of $y_{0:K}$ involves summing the joint probability in (7) across all possible state sequences $x_{0:K}$, then one gets (6). The proof is complete. ∎

Explication of the computation in (6) of Theorem 1 can be given as follows. Initially, at time $k=0$, the initial state of system (2) is $x(0)=\delta_N^{i_0}$ with probability π_{i_0}, and simultaneously emit the observation $y(0)=\delta_Q^{j_0}$ with probability $[H]_{j_0,i_0}$. Then the clock changes from 0 to time 1. Subsequently, the control input $u(0)=\delta_M^{\mu_0}$ steers the SBCN (2) from state $x(0)$ to state $x(1)=\delta_N^{i_1}$ with probability $[F\ltimes\delta_M^{\mu_0}]_{i_1,i_0}$, and at the same time generates the observation $y(1)=\delta_Q^{j_1}$ with probability $[H]_{j_1,i_1}$. This recurrence process continues in such a manner until the last transition at time $K-1$ from state $x(K-1)=\delta_N^{i_{K-1}}$ to state $x(K)=\delta_N^{i_K}$ (driven by $u(K-1)=\delta_M^{i_{K-1}}$) with probability $[F\ltimes\delta_M^{u_{K-1}}]_{i_K,i_{K-1}}$, and emitting the observation $y(K)=\delta_Q^{j_K}$ with probability $[H]_{j_K,i_K}$.

In what follows, we analyze the number of calculations required by Theorem 1 to get the probability of $y_{0:K}$ generated by the SBCN (2) with known control input sequence $u_{0:K-1}$. Firstly, for the STP of $F\in\mathcal{L}_{N\times(MN)}$ and $u(k)\in\mathcal{L}_{M\times 1}$, one totally needs $N^2\times M$ times of multiplications and $N^2(M-1)$ times of additions. Then, each term in the sum of (6) can be obtained after $N^2(2M-1)+2K+1$ times of calculations. Note that there are N possible states which can be reached at every time instant $k=0,1,\ldots,K$. Therefore, there are totally N^{K+1} possible choices for the state sequence $x_{0:K}$. The computational complexity of Theorem 1 requires the order of $O(M\cdot N^{K+3})$, which is computationally infeasible, even for small values of M, N and K. In the next subsection, a more efficient procedure is to be proposed for computing the quantity $\Pr\{y_{0:K}|\theta,u_{0:K-1}\}$, which will heavily reduce the computational complexity.

Algorithm 1 Forward Algorithm

Input: observations $y_{0:K}=\{\delta_Q^{j_0},\ldots,\delta_Q^{j_K}\}$, the control sequence $u_{0:K-1}=\{\delta_M^{\mu_0},\ldots,\delta_M^{\mu_{K-1}}\}$, matrices F and H, and the initial distribution vector π

Output: the probability $\Pr\{y_{0:K}|\theta,u_{0:K-1}\}$

procedure FORWARD ALGORITHM
  Initialization:
  for $i_0\leftarrow 1$ **to** N **do**
    $P_0^f(i_0)\leftarrow\pi_{i_0}\cdot\left[H\right]_{j_0,i_0}$
  end for
  Recursion:
  for $k\leftarrow 0$ **to** $K-1$ **do**
    for $i_{k+1}\leftarrow 1$ **to** N **do**
      Compute $P_{k+1}^f(i_{k+1})$ via (10)
    end for
  end for
  Termination:
  $\Pr\{y_{0:K}|\theta,u_{0:K-1}\}\leftarrow\sum_{i_K=1}^{N}P_K^f(i_K)$
  return $\Pr\{y_{0:K}|\theta,u_{0:K-1}\}$
end procedure

B. The Forward-Backward Algorithm

Let $P_k^f(i_k)$ be the joint probability of all the observations up to time instant k and the state $x(k)=\delta_N^{i_k}$, that is,

$$P_k^f(i_k)\triangleq\Pr\{y_{0:k},x(k)=\delta_N^{i_k}|\theta,u_{0:k-1}\}. \quad (8)$$

Based on Lemma 2, the following result provides a recursive STP-based algorithm to compute the probability of the observed sequence produced by the model.

Theorem 2: Consider the SBCN (2) with observations $y_{0:K}$ and control input sequence $u_{0:K-1}$. The probability of the observation sequence $y_{0:K}$ can be calculated as follows:

$$\Pr\{y_{0:K}|\theta,u_{0:K-1}\}=\sum_{i_K=1}^{N}P_K^f(i_K) \quad (9)$$

in which $P_K^f(i_K)$ can be calculated recursively as

$$P_{k+1}^f(i_{k+1})=\left[H\right]_{j_{k+1},i_{k+1}}\cdot\Big[\sum_{i_k=1}^{N}\left[F\ltimes\delta_M^{\mu_k}\right]_{i_{k+1},i_k}\cdot P_k^f(i_k)\Big] \quad (10)$$

with $k=0,\ldots,K-1$ and $P_0^f(i_0)=\pi_{i_0}\cdot[H]_{j_0,i_0}$, $i_0\in\{1,2,\ldots,N\}$.

Proof: According to the definition of $P_K^f(i_K)$, it is easy to verify that

$$\Pr\{y_{0:K}|\theta,u_{0:K-1}\}=\sum_{i_K=1}^{N}\Pr\{y_{0:K},x(K)=\delta_N^{i_K}|\theta,u_{0:K-1}\}$$
$$=\sum_{i_K=1}^{N}P_K^f(i_K)$$

which gives the objective computation as the sum of the terminal forward variable $P_K^f(i_K)$.

Next, we show the validity of the recursion relationships in (10) that allow $P_K^f(i_K)$ to be calculated efficiently. From the probability product rule and the total probability formula, together with the conditional independence properties, the forward variable $P_{k+1}^f(i_{k+1})$ can be expressed in terms of $P_k^f(i_k)$ as in (11). In order to implement this recursion, one needs a starting condition that is given by

$$P_0^f(i_0)=\Pr\{y(0),x(0)=\delta_N^{i_0}|\theta\}=\pi_{i_0}\cdot\left[H\right]_{j_0,i_0}.$$

The proof is complete. ∎

Remark 4: The forward algorithm[2] presented in Theorem 2 to compute $\Pr\{y_{0:K}|\theta,u_{0:K-1}\}$ is factorized/programmed in Algorithm 1, from which the relating computational complexity can be easily computed in terms of bit operations. From (10), to calculate the entry $P_{k+1}^f(i_{k+1})$ from the entries $P_k^f(i_k)$, one needs $N^2(2M-1)+2N$ calculations. Besides, to find all the N entries at time instant $k+1$ from those at k requires $N[N^2(2M-1)+2N]$ calculations. Note that the forward algorithm begins with $P_0^f(i_0)$ $(i_0=1,2,\ldots,N)$ and calculates $P_1^f(i_1)$, $P_2^f(i_2)$, …, $P_K^f(i_K)$ iteratively. The total number of calculations dedicated to Algorithm 1 is then

$$KN\left[N^2(2M-1)+2N\right]+N$$

which is in the order of $O(MKN^3)$ rather than the $O(M\cdot N^{K+3})$ as required by the direct calculation.

Remark 5: Theorem 2 provides a recursive STP-based method for calculating the probability of the observation sequence. Actually, the forward algorithm is a sort of dynamic programming algorithm, which utilizes a table to deposit the intermediate values as they build up the probability of the whole observation sequence. Algorithm 1 calculates the probability of $y_{0:K}$ by taking a sum over the probabilities of all possible states sequences that could generate the observation sequence, which can be calculated efficiently by folding each of these sequences into a single forward trellis.

[2]The forward algorithm is designed on the basis of algebraic form of the SBCNs that can be solved by using the MATLAB toolbox established by Cheng (http://lsc.amss.ac.cn/ dcheng/stp/STP.zip).

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI10.1109/TAC.2021.3106896, IEEE Transactions on Automatic Control.

Remark 6: Calculation of the forward probability (outlined in Algorithm 1) is based on the trellis diagram, which is further illustrated in Fig. 1. The value of each $P^f_{k+1}(i_{k+1})$ is computed by summing up all the previous values $P^f_k(i_k)$ ($i_k = 1, 2, \ldots, N$) weighted by transition probabilities $[F \ltimes \delta^{\mu_k}_M]_{i_{k+1}, i_k}$ and multiplied by the observation probability $[H]_{j_{k+1}, i_{k+1}}$. The key point is that, since there are N states at each time slot, all the possible state sequences $x_{0:K}$ will re-merge into these N nodes, no matter how long the observation sequence is. At time slot k, one needs to compute values of $P^f_k(i_k)$, $i_k = 1, 2, \ldots, N$, where each computation involves only N previous values of $P^f_{k-1}(i_{k-1})$. It is worth emphasizing that the structure of the trellis diagram shown in Fig. 1 illustrates the efficiency of the forward computation algorithm very well.

In a similar manner, we can consider a backward variable $P^b_k(i_k)$ defined as

$$P^b_k(i_k) \triangleq \Pr\{y_{k+1:K} \big| x(k) = \delta^{i_k}_N, \theta, u_{k:K-1}\}. \tag{12}$$

With the help of Lemma 2 and the Bayes theorem, we immediately have the following result.

Theorem 3: Consider SBCN (2) with noisy observations $y_{0:K}$ and control sequence $u_{0:K-1}$. Let $P^b_k(i_k)$ be the probability of the partial observation sequence from $k+1$ to the end, given state $x(k) = \delta^{i_k}_N$ and the model θ. Then, the probability of the noisy observation sequence $y_{0:K}$ can be calculated by

$$\Pr\{y_{0:K} | \theta, u_{0:K-1}\} = \sum_{i_0=1}^{N} \big[H\big]_{j_0, i_0} \cdot P^b_0(i_0) \cdot \pi_{i_0} \tag{13}$$

where $P^b_0(i_0)$ can be calculated recursively as

$$P^b_k(i_k) = \sum_{i_{k+1}=1}^{N} \big[H\big]_{j_{k+1}, i_{k+1}} \cdot P^b_{k+1}(i_{k+1}) \cdot \big[F \ltimes \delta^{\mu_k}_M\big]_{i_{k+1}, i_k} \tag{14}$$

in which $k = K-1, K-2, \ldots, 0$ and $P^b_K(i_K) \equiv 1$, $i_K = 1, 2, \ldots, N$.

Proof: It follows from (12) that

$$\begin{aligned}\Pr\{y_{0:K} | \theta, u_{0:K-1}\} &= \sum_{i_0=1}^{N} \Pr\{y_{0:K}, x(0) = \delta^{i_0}_N | \theta, u_{0:K-1}\} \\ &= \sum_{i_0=1}^{N} \big[H\big]_{j_0, i_0} \cdot P^b_0(i_0) \cdot \pi_{i_0}.\end{aligned}$$

By utilizing the conditional independence properties, the recursion in (15) can be similarly obtained for the quantities $P^b_k(i_k)$. In addition, one needs an initial condition for this recursion, that is, a value for $P^b_K(i_K)$. Based on the Bayes theorem, one has (16). By setting $k = K$ in the above equation and replacing $P^f_k(i_k)$ with its definition, one can obtain that $P^b_K(i_K) = (\Pr\{y_{0:K}, x(K) = \delta^{i_K}_N | \theta, u_{0:K-1}\})/P^f_K(i_K) = 1$ which implies that $P^b_K(i_K) = 1$ for all settings of i_K. The proof is complete. ∎

Remark 7: The pseudocode for computing is displayed in Algorithm 2. It is interesting to find that Algorithm 2 involves a backward message propagating process that evaluates $P^b_k(i_k)$ in terms of $P^b_{k+1}(i_{k+1})$. At each step, the backward algorithm absorbs the effect of observation $y(k+1)$ through the emission probability $[H]_{j_{k+1}, i_{k+1}}$, multiplied by the transition probability $[F \ltimes \delta^{\mu_k}_M]_{i_{k+1}, i_k}$, and then marginalizes $x(k+1)$.

Remark 8: The model-based fault detection and diagnosis problem has been investigated for stochastic Boolean dynamical systems in a recent work [2], and an effective diagnosis algorithm has been proposed to select the fault as the candidate with the largest posterior probability for the given observation sequence. It is worth mentioning that the calculation in the fault diagnosis step is on the basis of a bank of Boolean Kalman filter [4] running in the parallel (i.e., one for each candidate model), which implies that the performance of the proposed algorithm is closely correlated to that of the Boolean Kalman filter. Thus, the fault diagnosis algorithm proposed in [2] might not be appropriate in dealing with certain special/extreme cases, such as system (1) with larger noises p_i ($i = 1, 2, \ldots, l_1$) and q_j ($j = 1, 2, \ldots, l_2$). Here, the parameters p_i and q_j determine the intensity of the system noise and the observation noise, respectively.

Algorithm 2 Backward Algorithm

Input: observations $y_{0:K} = \{\delta^{j_0}_Q, \ldots, \delta^{j_K}_Q\}$, the control sequence $u_{0:K-1} = \{\delta^{\mu_0}_M, \ldots, \delta^{\mu_{K-1}}_M\}$, matrices F and H, and the initial distribution vector π

Output: the probability $\Pr\{y_{0:K} | \theta, u_{0:K-1}\}$

procedure BACKWARD ALGORITHM
  Initialization:
  for $i_K \leftarrow 1$ **to** N **do**
    $P^b_K(i_K) \leftarrow 1$
  end for
  Recursion:
  for $k \leftarrow K-1$ **to** 0 **do**
    for $i_k \leftarrow 1$ **to** N **do**
      Compute $P^b_k(i_k)$ via (14)
    end for
  end for
  Termination:
  $\Pr\{y_{0:K} | \theta, u_{0:K-1}\} \leftarrow \sum_{i_0=1}^{N} \big[H\big]_{j_0, i_0} \cdot P^b_0(i_0) \cdot \pi_{i_0}$
  return $\Pr\{y_{0:K} | \theta, u_{0:K-1}\}$
end procedure

C. Scaling Factors

Although computers have become so much capable, numerical issues still arise in the implementation of the developed algorithms, especially for the interminable state sequence. From the '**for**' loop of lines 8–10 in Algorithm 1, we note that, at each time instant k, the current value $P^f_{k+1}(i_{k+1})$ is calculated from the previous value $P^f_k(i_k)$ by multiplying with the transition probability $[F \ltimes \delta^{\mu_k}_M]_{i_{k+1}, i_k}$ and the observation probability $[H]_{j_{k+1}, i_{k+1}}$. Since these probability quantities are always prominently less than 1, the value of

$$\begin{aligned}P^f_{k+1}(i_{k+1}) &= \Pr\Big\{y(k+1) = \delta^{j_{k+1}}_Q \Big| y_{0:k}, x(k+1) = \delta^{i_{k+1}}_N, \theta, u_{0:k}\Big\} \cdot \Pr\Big\{y_{0:k}, x(k+1) = \delta^{i_{k+1}}_N \Big| \theta, u_{0:k}\Big\} \\ &= \Pr\Big\{y(k+1) = \delta^{j_{k+1}}_Q \Big| x(k+1) = \delta^{i_{k+1}}_N, \theta\Big\} \cdot \Big(\sum_{i_k=1}^{N} \Pr\Big\{y_{0:k}, x(k) = \delta^{i_k}_N, x(k+1) = \delta^{i_{k+1}}_N \Big| \theta, u_{0:k}\Big\}\Big) \\ &= \big[H\big]_{j_{k+1}, i_{k+1}} \cdot \Big(\sum_{i_k=1}^{N} \big[F \ltimes \delta^{\mu_k}_M\big]_{i_{k+1}, i_k} \cdot P^f_k(i_k)\Big)\end{aligned} \tag{11}$$

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI10.1109/TAC.2021.3106896, IEEE Transactions on Automatic Control.

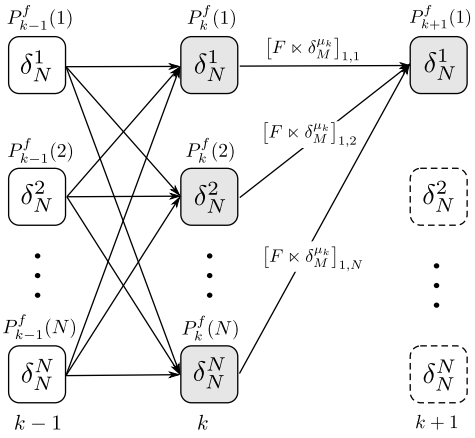

Fig. 1. Computation illustration of the forward probability in terms of a trellis diagram of states δ_N^i, $i = 1, 2, \ldots, N$, where $P_k^f(i_k)$ represents the probability of being in the state $\delta_N^{i_k}$ after knowing the first k observations, given the model parameter θ and the corresponding control input sequence $u_{0:k-1}$.

$P_{k+1}^f(i_{k+1})$ might close to 0 exponentially when the '**for**' loop (i.e., lines 7–12 in Algorithm 1) runs along the time sequence. For moderate lengths of the state sequence (say 100 or so), the computation of $P_{k+1}^f(i_{k+1})$ will quickly overlap the capability of the modern computer, even though the double-precision floating-point format is utilized.

In the following, we will discuss the re-scaled versions of $P_{k+1}^f(i_{k+1})$ and $P_k^b(i_k)$ aiming at increasing the numerical precision of the developed algorithms, thereby overcome the numerical difficulties in the implementations. In doing so, a normalized version of $P_k^f(i_k)$ is put forward as follows:

$$\begin{aligned}\hat{P}_k^f(i_k) &\triangleq \Pr\{x(k) = \delta_N^{i_k} | y_{0:k}, \theta, u_{0:k-1}\} \\ &= \frac{P_k^f(i_k)}{\Pr\{y_{0:k}|\theta, u_{0:k-1}\}}\end{aligned} \tag{17}$$

which will remain within the significant precision of the computers since it is the i_k-th entry of certain probability distribution vector over N possible states, i.e., $\sum_{i_k=1}^N \hat{P}_k^f(i_k) = 1$.

For the purpose of associating the scaled probability with the original forward one, we define the following scaling factors:

$$\xi_k \triangleq \Pr\{y(k)|y_{0:k-1}, \theta, u_{0:k-1}\} \tag{18}$$

where $\xi_0 \triangleq \Pr\{y(0)|\theta\}$, which can be computed as

$$\begin{aligned}\xi_0 &= \sum_{i_0=1}^N \Pr\Big\{y(0)|x(0) = \delta_N^{i_0}, \theta\Big\} \cdot \Pr\Big\{x(0) = \delta_N^{i_0}|\theta\Big\} \\ &= \sum_{i_0=1}^N \pi_{i_0} \cdot \big[H\big]_{j_0,i_0}.\end{aligned}$$

Then, according to the probability product rule, we arrive at

$$\Pr\{y_{0:k}|\theta, u_{0:k-1}\} = \prod_{\lambda=0}^k \xi_\lambda \tag{19}$$

which, together with (17), yields $P_k^f(i_k) = (\prod_{\lambda=0}^k \xi_\lambda) \cdot \hat{P}_k^f(i_k)$. It is easy to derive from (10) that

$$\begin{aligned}&\xi_{k+1} \cdot \hat{P}_{k+1}^f(i_{k+1}) \\ &= \big[H\big]_{j_{k+1},i_{k+1}} \cdot \Big(\sum_{i_k=1}^N \big[F \ltimes \delta_M^{\mu_k}\big]_{i_{k+1},i_k} \cdot \hat{P}_k^f(i_k)\Big)\end{aligned} \tag{20}$$

where $i_{k+1} = 1, 2, \ldots, N$. Note that $\sum_{i_{k+1}=1}^N \hat{P}_{k+1}^f(i_{k+1}) = 1$. The above equation yields

$$\xi_{k+1} = \sum_{i_{k+1}=1}^N \big[H\big]_{j_{k+1},i_{k+1}} \cdot \Big(\sum_{i_k=1}^N \big[F \ltimes \delta_M^{\mu_k}\big]_{i_{k+1},i_k} \cdot \hat{P}_k^f(i_k)\Big).$$

In light of (20), the recursion result (10) for $P_k^f(i_k)$ and $P_{k+1}^f(i_{k+1})$ can be finally converted into the following recursion for the normalized variables:

$$\begin{aligned}\hat{P}_{k+1}^f(i_{k+1}) =& \frac{1}{\xi_{k+1}} \cdot \big[H\big]_{j_{k+1},i_{k+1}} \\ &\cdot \Big(\sum_{i_k=1}^N \big[F \ltimes \delta_M^{\mu_k}\big]_{i_{k+1},i_k} \cdot \hat{P}_k^f(i_k)\Big)\end{aligned} \tag{21}$$

In order to implement this recursion, one needs a starting condition that is given by

$$\hat{P}_0^f(i_0) = \frac{P_0^f(i_0)}{\Pr\{y(0)|\theta\}} = \frac{\pi_{i_0} \cdot \big[H\big]_{j_0,i_0}}{\sum_{i_0=1}^N \pi_{i_0} \cdot \big[H\big]_{j_0,i_0}}.$$

Remark 9: It is worth pointing out that, at each step of the propagating phase concerning the forward message which is used to calculate $\hat{P}_{k+1}^f(i_{k+1})$, one needs to compute and store ξ_{k+1}, which is easy to be implemented since it is the coefficient that normalizes the right-hand side of (20) to give $\hat{P}_{k+1}^f(i_{k+1})$.

Now, we introduce the re-scaled variables $\hat{P}_k^b(i_k)$ by using $P_k^b(i_k) = \Big(\prod_{\lambda=k+1}^K \xi_\lambda\Big) \cdot \hat{P}_k^b(i_k)$. It follows from (12) and (19) that

$$\begin{aligned}\hat{P}_k^b(i_k) &= \frac{P_k^b(i_k)}{\big(\prod_{\lambda=0}^K \xi_\lambda\big)}\Big(\prod_{\lambda=0}^k \xi_\lambda\Big) \\ &= \frac{\Pr\{y_{k+1:K}|x(k) = \delta_N^{i_k}, \theta, u_{k:K-1}\}}{\Pr\{y_{k+1:K}|y_{0:k}, \theta, u_{k:K-1}\}}\end{aligned}$$

which will again be expected to perform numerically well because the quantity $\hat{P}_k^b(i_k)$ is just the ratio of two conditional probabilities. Then, the recursion equality (14) for $P_k^b(i_k)$ and $P_{k+1}^b(i_{k+1})$ can be converted into the one as follows:

$$\begin{aligned}&\hat{P}_k^b(i_k) \\ &= \frac{1}{\xi_{k+1}} \cdot \sum_{i_{k+1}=1}^N \big[H\big]_{j_{k+1},i_{k+1}} \cdot \hat{P}_{k+1}^b(i_{k+1}) \cdot \big[F \ltimes \delta_M^{\mu_k}\big]_{i_{k+1},i_k}.\end{aligned}$$

When applying this recursion relation, we make use of the scaling factor ξ_{k+1} that is previously computed in the $P_{k+1}^f(i_{k+1})$ phase.

Remark 10: It is easy to verify that the scaling technique makes the calculation of $\hat{P}_k^f(i_k)$ and $\hat{P}_k^b(i_k)$ within the dynamic range of the computer for $k = 1, \ldots, K$. It should be pointed out that the value of $\Pr\{y_{0:K}|\theta, u_{0:K-1}\}$ can be obtained by (19) rather than by summing up the $\hat{P}_k^f(i_k)$ ($\hat{P}_k^b(i_k)$) terms since they are re-scaled. Noting that scaling factor ξ_k ($k = 1, \ldots, K$) is often significantly less than 1, the calculation of $\Pr\{y_{0:K}|\theta, u_{0:K-1}\}$ via (19) would exceed the dynamic range of the computer. Therefore, the logarithm of $\Pr\{y_{0:K}|\theta, u_{0:K-1}\}$ is utilized to cope with the model evaluation problem, which can be calculated from the scaling factors as $\log \Pr\{y_{0:K}|\theta, u_{0:K-1}\} = \sum_{\lambda=0}^K \log \xi_\lambda$.

Remark 11: In the interesting paper [20], an observer has been designed to evaluate the probabilistic distribution of the state vectors. Based on this observer, a novel fault detection scheme has been proposed for the mix-valued probabilistic BNs. It is worth pointing out that the algorithm proposed in [20] cannot be further utilized to cope with the fault diagnosis problems. If all the possible types of system faults are known which are modeled respectively by $\theta_1, \theta_2, \ldots, \theta_\ell$, the forward/backward algorithms developed in this

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI10.1109/TAC.2021.3106896, IEEE Transactions on Automatic Control.

note then could be used to deal with the fault diagnosis problem for BNs described by (3).

IV. Numerical Simulation

In this section, we employ a GRN known as the *lac* operon in *Escherichia coli* as a numerical example, which is of vital importance as it can provide numerous insights into understanding the sugar metabolism. By encoding the interaction type (that is, activation or inhibition) in Boolean representation, a reduced Boolean model has been established in [30] to simulate the dynamic behavior and the interaction of the *lac* gene as follows:

$$\begin{cases} M(k+1) = \neg G_e(k) \wedge (L(k) \vee L_m(k)) \\ L(k+1) = M(k) \wedge L_e(k) \wedge \neg G_e(k) \\ L_m(k+1) = ((L_{em}(k) \wedge M(k) \vee L_e(k)) \wedge \neg G_e(k) \end{cases} \tag{22}$$

where M, L, and L_m indicate the *lac* mRNA, the lactose in high concentration, and the lactose in medium concentration, respectively; G_e stands for the glucose; L_{em} and L_e represent, respectively, the medium concentration and the high concentration of extra cellular lactose.

Remark 12: As is well known, the *lac* operon exhibits bistability. In order to show that model (22) also exhibits the same dynamics, stochasticity in the uptake of the inducer is considered by introducing a random variable $\mathcal{L}_e \sim N(\mu, \sigma)$. Then, concentration of the extracellular lactose represented by (L_e, L_{em}) is a function of $\mathcal{L}_e$ as follows:

$$(L_e, L_{em}) = \begin{cases} (0,0), & \text{if } \mathcal{L}_e < 1 \\ (0,1), & \text{if } 1 \le \mathcal{L}_e < 1 \\ (1,1), & \text{if } 2 \le \mathcal{L}_e \end{cases}$$

which implies that $w(k) = \ltimes_{i=1}^{2} w_i(k)$ ($w_1(k)$ and $w_2(k)$ are the vector form of the logical variables $L_e(k)$ and $L_{em}(k)$, respectively) is a random logical variable defined in Definition 2, that is, $w(k)$ takes values in the set $\{\delta_4^1, \delta_4^3, \delta_4^4\}$ and $w(k) = \delta_4^i$ with certain probability $\mathbb{P}_i$ $(i = 1, 3, 4)$. In this case, system (22) becomes an SBCN as in (2) with $u(k)$ serving as a control input. For notational convenience, let $\mathbb{P} = (\mathbb{P}_1, \mathbb{P}_3, \mathbb{P}_4)$.

Let x_1, x_2, x_3 and u be the vector form of the logical variables M, L, L_m and G_e, respectively. By resorting to the STP technique and setting $x(k) = \ltimes_{i=1}^{3} x_i(k)$, $w(k) = \ltimes_{i=1}^{2} w_i(k)$, the BN (22) can be converted into the following algebraic form:

$$x(k+1) = \mathcal{F} w(k) u(k) x(k) \tag{23a}$$

where

$$\begin{aligned} \mathcal{F} = \delta_8[&8,8,8,8,8,8,8,8,1,1,1,5,3,3,3,7,8,8,8,8,8,8,\\ &8,8,8,2,2,2,6,4,4,4,8,8,8,8,8,8,8,8,8,3,3,\\ &3,7,4,4,4,8,8,8,8,8,8,8,8,8,4,4,4,8,4,4,4,8]. \end{aligned}$$

The observation equation in this example is given as

$$Y(k) = X(k) \oplus V(k) \tag{23b}$$

where symbol "$\oplus$" indicates the modulo-2 addition. Here, the measurement noise $V(k) = (V_1(k), V_2(k), V_3(k))$ is assumed to be white with $V_j(k) \sim B(1, q_j)$ for $j = 1, 2, 3$, and it is mutually independent with the system noise and the initial state. By resorting to the STP technique, we have

$$\begin{aligned} \mathcal{H} = \delta_8[&8,7,6,5,4,3,2,1,7,8,5,6,3,4,1,2,6,5,8,7,2,\\ &1,4,3,5,6,7,8,1,2,3,4,4,3,2,1,8,7,6,5,3,4,\\ &1,2,7,8,5,6,2,1,4,3,6,5,8,7,1,2,3,4,5,6,7,8]. \end{aligned}$$

In this example, the complete parameters set of system (23) is denoted by $\theta = (\mathcal{F}, \mathcal{H}, \mathbb{P}, q, \pi)$, where $q = (q_1, q_2, q_3)$. We take $\mathbb{P} = (0.1, 0.1, 0.8)$, $q = (0.01, 0.01, 0.01)$ and $\pi = (0.05, 0, 0, 0.05, 0, 0, 0, 0.9)$. Without loss of generality, the control input sequence $u_{0:49}$ is set to be $(\delta_2^2, \delta_2^2, \ldots, \delta_2^2)$, which means that the glucose is low ($G_e(k) = 0$) at time instants $k = 0, \ldots, 49$. Then, the transition matrix F and the observation matrix H in Algorithm 1 can be achieved. Fig. 2 displays the entire observed output sequence $y_{0:50}$. It can be observed that the system spends a significant amount of time in the equilibrium state $(0, 0, 0)$. The probability that $y_{0:50}$ is produced by the above addressed model can be calculated by Algorithm 1 as $\Pr\{y_{0:50}|\theta, u_{0:49}\} = 1.0927 \times 10^{-7}$. Here, the total number of calculations dedicated to Algorithm 1 is 390408 (as opposed to 1.2114×10^{49} required by the direct calculation method).

As a comparison, let $\theta' = (\mathcal{F}, \mathcal{H}, \mathbb{P}', q', \pi)$ be the complete parameters set of another identified BN with $\mathbb{P}' = (0.15, 0.05, 0.8)$ and $q' = (0.015, 0.015, 0.015)$. By Algorithm 1, we can obtain the probability that the observed sequence $y_{0:50}$ is produced by this BN with parameter θ' as follows:

$$\Pr\{y_{0:50}|\theta', u_{0:49}\} = 2.4955 \times 10^{-8} < \Pr\{y_{0:50}|\theta, u_{0:49}\}$$

which implies that the Boolean model with parameter θ matches the given observed output sequence $y_{0:50}$ better than the one with θ'.

$$\begin{aligned} P_k^b(i_k) &= \sum_{i_{k+1}=1}^{N} \Pr\left\{y_{k+1:K} \middle| x(k+1) = \delta_N^{i_{k+1}}, \theta, u_{k:K-1}\right\} \cdot \Pr\left\{x(k+1) = \delta_N^{i_{k+1}} \middle| x(k) = \delta_N^{i_k}, \theta, u_{k:K-1}\right\} \\ &= \sum_{i_{k+1}=1}^{N} \Pr\left\{y(k+1) = \delta_Q^{j_{k+1}} \middle| y_{k+2:K}, x(k+1) = \delta_N^{i_{k+1}}, \theta, u_{k+1:K-1}\right\} \cdot P_{k+1}^b(i_{k+1}) \cdot \left[F \ltimes \delta_M^{\mu_k}\right]_{i_{k+1}, i_k} \\ &= \sum_{i_{k+1}=1}^{N} \left[H\right]_{j_{k+1}, i_{k+1}} \cdot P_{k+1}^b(i_{k+1}) \cdot \left[F \ltimes \delta_M^{\mu_k}\right]_{i_{k+1}, i_k} \end{aligned} \tag{15}$$

$$\begin{aligned} P_k^f(i_k) P_k^b(i_k) &= \Pr\left\{y_{0:k} | x(k) = \delta_N^{i_k}, \theta, u_{0:k-1}\right\} \cdot \Pr\left\{x(k) = \delta_N^{i_k} | \theta, u_{0:k-1}\right\} \cdot \Pr\left\{y_{k+1:K} | x(k) = \delta_N^{i_k}, \theta, u_{k:K-1}\right\} \\ &= \Pr\left\{y_{0:K} | x(k) = \delta_N^{i_k}, \theta, u_{0:K-1}\right\} \cdot \Pr\left\{x(k) = \delta_N^{i_k} | \theta, u_{0:k-1}\right\} \\ &= \Pr\left\{y_{0:K}, x(k) = \delta_N^{i_k} | \theta, u_{0:K-1}\right\} \end{aligned} \tag{16}$$

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI10.1109/TAC.2021.3106896, IEEE Transactions on Automatic Control.

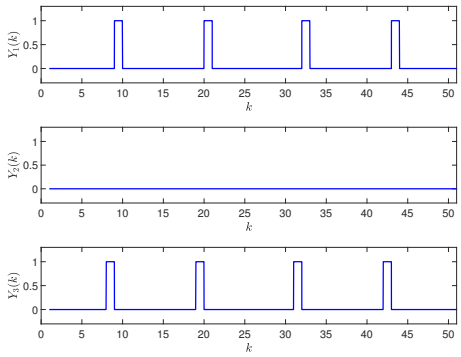

Fig. 2. The observation value of the measurement output in (23b).

V. Conclusion

In this paper, model evaluation of the SBCNs has been investigated based on the STP of matrices. By converting the SBCN into an algebraic form with the dynamics similar to that of a hidden Markov model, a straightforward approach has been proposed to compute the probability that the given observed sequence is produced by the model. In order to reduce the computational complexity, two recursive STP-based algorithms have been designed for model evaluation by resorting to the forward-backward technique. Moreover, scaling factors have also been introduced to deal with the numerical issues arising in the implementation of the developed algorithms. Finally, a Boolean model of the *lac* operon has been employed to illustrate the applicability and effectiveness of the proposed algorithms. In the future, we intend to utilize our proposed forward/backward algorithms to deal with the fault diagnosis problem for BNs as well as other complex systems [34], [35].

References

[1] T. Akutsu and A. A. Melkman, "Identification of the structure of a probabilistic Boolean network from samples including frequencies of outcomes," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 30, no. 8, pp. 2383–2396, Aug. 2019.

[2] A. Bahadorinejad and U. M. Braga-Neto, "Optimal fault detection and diagnosis in transcriptional circuits using next-generation sequencing," *IEEE-ACM Trans. Comput. Biol. Bioinform.*, vol. 15, no. 2, pp. 516–525, Mar.-Apr. 2018.

[3] A. Bahadorinejad and U. M. Braga-Neto, "Adaptive particle filtering for fault detection in partially observed Boolean dynamical systems," *IEEE-ACM Trans. Comput. Biol. Bioinform.*, vol. 17, no. 4, pp. 1105–1114, Jul.-Aug. 2020.

[4] U. M. Braga-Neto, "Optimal state estimation for Boolean dynamical systems," in *Proc. 45th Auu. Asilomar Conf. Signals, Syst. Comput.*, Pacific Grove, CA, USA, 2011, pp. 1050–1054.

[5] H. Chen and J. Liang, "Local Synchronization of interconnected Boolean networks with stochastic disturbances," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 31, no. 2, pp. 452–463, Feb. 2020.

[6] T. Chen and U. M. Braga-Neto, "Maximum-likelihood estimation of the discrete coefficient of determination in stochastic Boolean systems," *IEEE Trans. Signal Process.*, vol. 61, no. 15, pp. 3880–3894, Aug. 2013.

[7] D. Cheng, "Disturbance decoupling of Boolean control networks," *IEEE Trans. Autom. Control*, vol. 56, no. 1, pp. 2–10, Jan. 2011.

[8] D. Cheng, H. Qi, and Z. Li, "Model construction of Boolean network via observed data," *IEEE Trans. Neural Netw.*, vol. 22, no. 4, pp. 525–536, Apr. 2011.

[9] D. Cheng, H. Qi, and Z. Li, *Analysis and Control of Boolean Networks: A Semi-tensor Product Approach*. London, U.K.: Springer-Verlag, 2011.

[10] D. Cheng and Y. Zhao, "Identification of Boolean control networks," *Automatica*, vol. 47, no. 4, pp. 702–710, Apr. 2011.

[11] M. Choi, J. Shi, S. H. Jung, X. Chen, and K.-H. Cho, "Attractor landscape analysis reveals feedback loops in the p53 network that control the cellular response to DNA damage," *Sci. Signal.*, vol. 5, no. 251, p. ra83, Nov. 2012.

[12] H. De Jong, "Modeling and simulation of genetic regulatory systems: a literature review," *J. Comput. Biol.*, vol. 9, no. 1, pp. 67–103, 2002.

[13] E. Fornasini and M. E. Valcher, "Optimal control of Boolean control networks," *IEEE Trans. Autom. Control*, vol. 59, no. 5, pp. 1258–1270, May 2014.

[14] E. Fornasini and M. E. Valcher, "Fault detection analysis of Boolean control networks," *IEEE Trans. Autom. Control*, vol. 60, no. 10, pp. 2734–2739, Oct. 2015.

[15] S. Han, R. K. W. Wong, T. C. M. Lee, L. Shen, S.-Y. R. Li, and X. Fan, "A full Bayesian approach for Boolean genetic network inference," *PLoS One*, vol. 9, no. 12, p. e115806, Dec. 2014.

[16] S. A. Kauffman, "Metabolic stability and epigenesis in randomly constructed genetic nets," *J. Theor. Biol.*, vol. 22, no. 3, pp. 437–467, 1969.

[17] D. Laschov and M. Margaliot, "Minimum-time control of Boolean networks," *SIAM J. Control Optim.*, vol. 5, no. 4, pp. 2869–2892, 2013.

[18] R. Laubenbacher and B. Stigler, "A computational algebra approach to the reverse engineering of gene regulatory networks," *J. Theor. Biol.*, vol. 229, no. 4, pp. 523–537, Aug. 2004.

[19] H. Lähdesmäki, I. Shmulevich, and O. Yli-Harja, "On learning gene regulatory networks under the Boolean network model," *Mach. Learn.*, vol. 52, no. 1-2, pp. 147–167, 2003.

[20] T. Leifeld, Z. Zhang, and P. Zhang, "Fault detection for probabilistic Boolean networks," in *European Control Conference*, Aalborg, Denmark, 2016, pp. 740–745.

[21] T. Leifeld, Z. Zhang, and P. Zhang, "Identification of Boolean network models from time series data incorporating prior knowledge," *Front. Physiol.*, vol. 9, p. 695, Jun. 2018.

[22] F. Li and Y. Tang, "Pinning control design for the stabilization of Boolean networks," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 27, no. 7, pp. 1585–1590, Jul. 2016.

[23] R. Li and T. Chu, "Complete synchronization of Boolean networks," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 23, no. 5, pp. 840–846, May 2012.

[24] J. Liang, H. Chen, and J. Lam, "An improved criterion for controllability of Boolean control networks," *IEEE Trans. Autom. Control*, vol. 62, no. 11, pp. 6012–6018, Nov. 2017.

[25] M. Yang, R. Li, and T. Chu, "Controller design for disturbance decoupling of Boolean control networks," *Automatica*, vol. 49, no. 1, pp. 273–277, Jan. 2013.

[26] M. Meng, L. Liu, and G. Feng, "Stability and l_1 gain analysis of Boolean networks with Markovian jump parameters," *IEEE Trans. Autom. Control*, vol. 62, no. 8, pp. 4222–4228, Aug. 2017.

[27] D. Murrugarra, A. Veliz-Cuba, B. Aguilar, S. Arat, and R. Laubenbacher, "Modeling stochasticity and variability in gene regulatory networks," *EURASIP J. Bioinform. Syst. Biol.*, vol. 2012, no. 1, p. 5, Jun. 2012.

[28] 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.

[29] A. Veliz-Cuba, "An algebraic approach to reverse engineering finite dynamical systems arising from biology," *SIAM J. Appl. Dyn. Syst.*, vol. 11, no. 1, pp. 31–48, 2012.

[30] A. Veliz-Cuba and B. Stigler, "Boolean models can explain bistability in the *lac* operon," *J. Comput. Biol.*, vol. 18, no. 6, pp. 783–794, Jul. 2011.

[31] Z. Wang, H. Wu, J. Liang, and X. Liu, "On modeling and state estimation for genetic regulatory networks with polytopic uncertainties," *IEEE Trans. Nanobiosci.*, vol. 12, no. 1, pp. 13–20, Mar. 2013.

[32] K. Zhang and L. Zhang, "Observability of Boolean control networks: a unified approach based on finite automata," *IEEE Trans. Autom. Control*, vol. 61, no. 9, pp. 2733–2738, Sep. 2016.

[33] P. Zhu, J. Liang, and J. Han, "Gene perturbation and intervention in context-sensitive stochastic Boolean networks," *BMC Syst. Biol.*, vol. 8, p. 60, May 2014.

[34] K. Zhu, J. Hu, Y. Liu, N. D. Alotaibi and F. E. Alsaadi, "On ℓ_2-ℓ_∞ output-feedback control scheduled by stochastic communication protocol for two-dimensional switched systems," *International Journal of Systems Science*, in press, DOI: 10.1080/00207721.2021.1914768.

[35] L. Zou, Z. Wang, J. Hu, Y. Liu and X. Liu, "Communication-protocol-based analysis and synthesis of networked systems: Progress, prospects and challenges," *International Journal of Systems Science*, in press, DOI: 10.1080/00207721.2021.1917721.