Research Article

Polynomial-Time Algorithm for Controllability Test of a Class of Boolean Biological Networks

Koichi Kobayashi,[1] Jun-Ichi Imura,[2] and Kunihiko Hiraishi[1]

[1] *School of Information Science, Japan Advanced Institute of Science and Technology, Nomi, Ishikawa 923-1292, Japan*
[2] *Graduate School of Information Science and Engineering, Tokyo Institute of Technology, Oh-okayama, Tokyo 152-8552, Japan*

Correspondence should be addressed to Koichi Kobayashi, k-kobaya@jaist.ac.jp

In recent years, Boolean-network-model-based approaches to dynamical analysis of complex biological networks such as gene regulatory networks have been extensively studied. One of the fundamental problems in control theory of such networks is the problem of determining whether a given substance quantity can be arbitrarily controlled by operating the other substance quantities, which we call the controllability problem. This paper proposes a polynomial-time algorithm for solving this problem. Although the algorithm is based on a sufficient condition for controllability, it is easily computable for a wider class of large-scale biological networks compared with the existing approaches. A key to this success in our approach is to give up computing Boolean operations in a rigorous way and to exploit an adjacency matrix of a directed graph induced by a Boolean network. By applying the proposed approach to a neurotransmitter signaling pathway, it is shown that it is effective.

1. Introduction

Various approaches to modeling, analysis, and control synthesis of biological networks such as gene regulatory networks and metabolic networks have been recently developed in the control community as well as the theoretical biology community [1]. In these approaches, it is one of the final goals to develop systematic drug discovery and cancer treatment [2, 3]. Biological networks in general can be expressed by ordinary/partial differential equations with high nonlinearity and high dimensionality. Since such complexities cause difficulties in analysis and control design, various simpler models such as Petri nets, Bayesian networks, Boolean networks, and hybrid systems have been proposed for dealing with complex and large-scale biological networks at the expense of rigorous analysis (see e.g., [4, 5]).

This paper discusses the controllability problem of biological networks. In gene regulatory networks, for example, the controllability problem is defined as the problem of determining whether expressions of genes of interest can be arbitrarily controlled by expressions of a specified set of the other genes. As far as we know, two approaches to the

controllability analysis of such biological networks have been developed so far: a piecewise affine model-based approach and a Boolean network model-based approach. However, the former approach can be applied to only the class of relatively low-dimensional systems [6, 7].

On the other hand, a Boolean network model, where binary state variables are assigned to nodes and the transition rules of the state are given by Boolean functions [8, 9], will be more practical for analysis of *large-scale* biological networks thanks to its bold simplification. Akutsu et al. have recently discussed the controllability problem of Boolean networks with control nodes and controlled nodes and have proven that this problem is NP-hard in a general setting [10]. Furthermore, they have proposed a polynomial-time algorithm for the classes of networks including a tree structure or at most one loop, and an exponential-time algorithm for the other classes. Indeed there is a criticism that a Boolean network model is too simple as a model of biological networks, but for large-scale networks it will be able to provide some indication or clue towards further detailed analysis. Thus various approaches based on this model have been well-studied so far (see e.g., [11–19]).

Motivated by the theoretical results in [10], this paper also focuses on the controllability problem of Boolean networks with control nodes and controlled nodes and proposes a sufficient condition for the Boolean network to be controllable, which can be easily verified by a polynomial-time algorithm. Our standing point is to give up computing complex Boolean operations in a rigorous way and to focus on deriving an easily-checkable sufficient condition for controllability so as to be applied to large-scale networks. The obtained algorithm is based on simple operations on an adjacency matrix of a directed graph induced by a Boolean network. This is a remarkable point of our approach, different from the method in [10], and enables us to apply our approach to a wider class of Boolean networks including nontree structures.

First, after the definition of controllability of Boolean network models with control nodes and controlled nodes is described, a sufficient condition for the controllability is derived in the form of an algorithm. Next, the computational complexity for the algorithm is discussed to show that it is a polynomial-time algorithm. In addition, PC-based numerical experiments show that the obtained algorithm is applicable to a class of Boolean networks with at least 1000 nodes. Finally, as an illustrative example, the proposed algorithm is applied to the Boolean network model of a neurotransmitter signaling pathway [20], which expresses an interaction pathway between the glutamatergic and dopaminergic receptors. Note that the polynomial-time algorithm proposed in [10] cannot be always applied to this problem. This Boolean network model consists of 16 nodes, and the problem of simultaneously controlling two important nodes among them, that is, concentration of exocytosis and phospholipase C, is discussed based on the proposed algorithm. As a result, we show that for example, they can be simultaneously controlled by keeping substance concentration at the other 4 nodes constant with appropriate values.

Notation 1. Let $\mathcal{N}$ denote the set of nonnegative integers and $\{0, 1\}^{m \times n}$ the set of $m \times n$ matrices consisting of elements 0 and 1. We also denote by I_n and $0_{m \times n}$ the $n \times n$ identity matrix and the $m \times n$ zero matrix, respectively. For simplicity of notation, we sometimes use the symbol 0 instead of $0_{m \times n}$ and the symbol I instead of I_n. Let $M^\top$ express the transpose of the matrix M.

2. Boolean Network Models

This section provides a brief review on a Boolean network model [8, 9]. A Boolean network model consists of a set of nodes and a set of regulation rules for nodes, where each node expresses a gene, a molecule, or an event in the genetic network. The state variable ξ_i at node i takes a Boolean value of 0 or 1 representing "inactive" or "active" status of the node, respectively. A regulation rule for each node is given in terms of a Boolean function, and each node state changes synchronously.

As an example, we consider a very simple and interesting Boolean network model of an apoptosis network in Figure 1
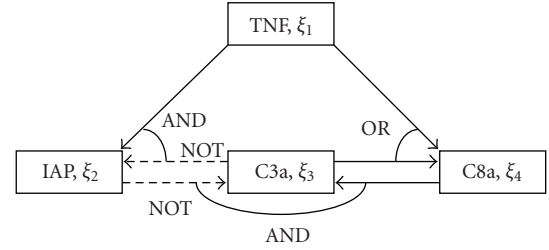

FIGURE 1: Simplified model of an apoptosis network. Activation (solid), Inhibition (broken).

given by

$$
\begin{aligned}
\xi_1(k+1) &= \xi_1(k), \\
\xi_2(k+1) &= \xi_1(k) \wedge \neg\xi_3(k), \\
\xi_3(k+1) &= \neg\xi_2(k) \wedge \xi_4(k), \\
\xi_4(k+1) &= \xi_1(k) \vee \xi_3(k),
\end{aligned}
\tag{1}
$$

where $\neg$, $\wedge$, and $\vee$ denote logical NOT, AND, and OR, respectively, $k \in \mathcal{N}$ denotes the discrete time, the concentration level (high or low) of the tumor necrosis factor (TNF, a stimulus) is denoted by ξ_1, the concentration level of the inhibitor of apoptosis proteins (IAP) by ξ_2, the concentration level of the active caspase 3 (C3a) by ξ_3, and the concentration level of the active caspase 8 (C8a) by ξ_4. Here if the binary variable ξ_i has the value of "1", then the concentration of a certain reactant gets larger than a prescribed threshold (i.e., it is active), otherwise less than that. In addition, logical NOT corresponds to inhibition of gene expressions.

Since the caspase C3a is responsible for cleaving or breaking many other proteins, a high-level of the C3a concentration, that is, $\xi_3 = 1$ implies cell near-death; otherwise, cell survival. As seen in (1), if the concentration of IAP is high ($\xi_2 = 1$) or the concentration of the caspase C8a is low ($\xi_4 = 0$), then the concentration of C3a gets low, that is, $\xi_3 = 0$. On the other hand, ξ_2 and ξ_4 at the next time depend on the value of ξ_3 as well as ξ_1. In this way, some dynamical interactions exist. See [21, 22] for further details.

A general form of a Boolean network model is given by the state equation

$$
\xi(k+1) = f_a(\xi(k)),
\tag{2}
$$

where $\xi(k) = [\xi_1(k)\ \xi_2(k)\ \cdots\ \xi_l(k)]^\top \in \{0, 1\}^l$ is the state vector at time $k \in \mathcal{N}$, and $f_a : \{0, 1\}^l \to \{0, 1\}^l$ is a Boolean function, where logical operators consist of AND ($\wedge$), OR ($\vee$), NOT ($\neg$), and XOR ($\oplus$).

3. Problem Formulation

In a Boolean network model (2), the state $\xi(k)$ is uniquely determined by giving the initial state $\xi(0) = \xi_0 \in \{0, 1\}^l$, which implies that (2) is an autonomous system and has no control nodes.

On the other hand, this paper will consider the Boolean network model with control (i.e., input) nodes and controlled (i.e., output) nodes to discuss the output-controllability of this model. This model is given by

$$\Sigma \begin{cases} x(k+1) = f(x(k), u(k)), \\ y(k) = Cx(k), \end{cases} \quad (3)$$

where each element of $u \in \{0,1\}^m$ denotes the state of the control node whose value can be arbitrarily given as an external control input in the Boolean network, each element of $x \in \{0,1\}^n$ denotes the state of the node except for the control nodes in the Boolean network, and each element of $y \in \{0,1\}^r$ denotes the state of the node to be controlled as an output in the network. Note here that y does not imply a measured output. Hereafter according to control theory, x, u, and y are called a "state", "control input" and "output", respectively. In addition, $f : \{0,1\}^n \times \{0,1\}^m \rightarrow \{0,1\}^n$ is a Boolean function, and $C \in \{0,1\}^{r \times n}$ is the output matrix satisfying for each element c_{ij} of C

$$\sum_{i=1}^{r} c_{ij} = 1, \quad \forall j, \qquad \sum_{j=1}^{n} c_{ij} = 1, \quad \forall i. \quad (4)$$

Furthermore, the product of C and x in $y = Cx$ expresses a product operation on matrices/vectors of the real number field. Thus the above condition on C guarantees that the output is the state variable itself, that is, for each i there exists j such that $y_i = x_j$ holds. The case of $y = x$ is also included here. This condition on C will not be restrictive in analyzing controllability of biological networks such as gene regulatory networks, since the relation on regulation among genes/molecules will be mainly discussed there.

For the system Σ of (3), the notion of output-controllability is defined as follows.

Definition 1. Suppose that for the system Σ of (3), the finite time $T \in \mathcal{N}$ and the initial state $x(0) = x_0 \in \{0,1\}^n$ are given. Then the system Σ is said to be *T-output-controllable* at x_0 if for every $y_f \in \{0,1\}^r$, there exists a control input sequence $u(k) \in \{0,1\}^m$, $k = 0,1,\dots,T-1$, such that $y(T) = y_f$. Furthermore, the system Σ is said to be *T-output-controllable* if it is *T-output-controllable* at every x_0.

The above notion of controllability comes from the fact that, for example, in control of genetic networks we often would like to determine if expressions of certain gene of interest (corresponding to y) will be able to be inhibited (or activated) by means of appropriately adjusting the expressions of a given set of genes (corresponding to u). It is remarked that we assume that the control time T is explicitly specified in the above definition.

Let us get back to the Boolean network model (1) of an apoptosis network. As discussed in [21, 22], we consider ξ_1(TNF) itself as a control input. So by ignoring the dynamics on ξ_1, that is, $\xi_1(k+1) = \xi_1(k)$, we suppose in (1) that

$x(k) = [\xi_2(k)\ \xi_3(k)\ \xi_4(k)]^\top$ and $u(k) = \xi_1(k)$, which yields (3) of the form

$$x_1(k+1) = \neg x_2(k) \wedge u(k),$$
$$x_2(k+1) = \neg x_1(k) \wedge x_3(k), \quad (5)$$
$$x_3(k+1) = x_2(k) \vee u(k),$$

where $x_i(k)$ denotes the i-th element of $x(k)$. As for the output $y = Cx$, either case of

$$C = I_3, \qquad C = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}, \qquad C = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix},$$

$$C = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \qquad C = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix}, \qquad C = \begin{bmatrix} 0 & 1 & 0 \end{bmatrix},$$

$$C = \begin{bmatrix} 0 & 0 & 1 \end{bmatrix}$$

$$(6)$$

can be treated by assumption. Then let us verify the T-output controllability of the system (5). As discussed in Section 2, $x_2(= \xi_3) = 1$ expresses cell near-death, and $x_2(= \xi_3) = 0$ expresses cell survival. So we would like to know if the system is T-output-controllable with respect to the output $y = x_2$. Suppose that $x_0 = [0\ 0\ 0]^\top$ (i.e., the initial states of IAP, C3a and C8a are all low-level), $C = [0\ 1\ 0]$ (i.e., $y = x_2$), and $T = 2$. Then since $y(2) = 0$ holds independently of u by simple calculation, we see that system (5) is not 2-output-controllable at x_0, which implies that we cannot control the system from the state "cell survival" within 2 time steps no matter how the control value of u is given.

On the other hand, suppose in (1) that $x(k) = [\xi_1(k)\ \xi_2(k)\ \xi_3(k)]^\top$ and $u(k) = \xi_4(k)$. Then we obtain (3) of the form

$$x_1(k+1) = x_1(k),$$
$$x_2(k+1) = x_1(k) \wedge \neg x_3(k), \quad (7)$$
$$x_3(k+1) = \neg x_2(k) \wedge u(k),$$

where $\xi_4(k+1) = \xi_1(k) \vee \xi_3(k)$ is ignored. Suppose that $x_0 = [1\ 0\ 1]^\top$, $T = 2$, and

$$C = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (8)$$

(i.e., $y = [x_2\ x_3]^\top (= [\xi_2\ \xi_3]^\top)$). Then since $x_2(2) = \neg u(0)$ and $x_3(2) = u(1)$ are obtained, we see that system (5) is 2-output-controllable at x_0, for example, (a) $y(2) = [0\ 0]^\top$ for $u(0) = 1$, $u(1) = 0$, (b) $y(2) = [0\ 1]^\top$ for $u(0) = 1$, $u(1) = 1$, (c) $y(2) = [1\ 0]^\top$ for $u(0) = 0$, $u(1) = 0$, and (d) $y(2) = [1\ 1]^\top$ for $u(0) = 0$, $u(1) = 1$. This implies we can *simultaneously* control the value of x_2 and x_3 at $T = 2$. In this way, the proposed controllability enables us to verify the existence of a control input sequence such that the output has the desired value in a given finite time, and the obtained result indicates how to give the value of a control input sequence.

Next, we will explain our basic strategy for deriving the controllability condition. Let us consider a Boolean network expressed as the state equation

$$\xi_1(k+1) = \xi_2(k) \wedge \xi_3(k),$$
$$\xi_2(k+1) = \xi_1(k), \tag{9}$$
$$\xi_3(k+1) = \neg\xi_2(k),$$

which is given by [10]. Although this model is very simple, it provides significant clues to address this problem. For the Boolean network model (9), we can consider three possible specifications, choosing either $\xi_1(k)$, $\xi_2(k)$, or $\xi_3(k)$ to be the control input for the system.

First, suppose that $x(k) = [\xi_1(k)\ \xi_2(k)]^\top$ and $u(k) = \xi_3(k)$, that is, $\xi_3(k)$ itself is the control input. Then it follows that

$$x_1(k+1) = x_2(k) \wedge u(k),$$
$$x_2(k+1) = x_1(k). \tag{10}$$

Note here that $\xi_3(k+1) = \neg\xi_2(k)$ is ignored because we assume that $\xi_3(k)$ itself is the control input. As for the output $y = Cx$, either case of $C = I_2$, $C = [1\ 0]$, $C = [0\ 1]$ can be considered in this case. Consider the controllability of the system (10) with $y = x$ (i.e., $C = I_2$) for $T = 2$. In this example, we will consider whether system (10) is T-output-controllable or not by directly calculating state trajectories of each system. From (10), we have

$$x_1(2) = x_1(0) \wedge u(1),$$
$$x_2(2) = x_2(0) \wedge u(0). \tag{11}$$

So if $x_1(0) = 0$, $x_1(2) \equiv 0$ holds irrespective of the value of $u(1)$, similarly for the case of $x_2(0) = 0$. Therefore, we see that system (10) is not 2-output-controllable. In the same way, we see that system (10) is not T-output-controllable in every case of $C = I_2$, $C = [1\ 0]$, and $C = [0\ 1]$ for $T \geq 2$.

Secondly, suppose that $x(k) = [\xi_1(k)\ \xi_3(k)]^\top$ and $u(k) = \xi_2(k)$, that is, $\xi_2(k)$ itself is regarded as the control input. Then we obtain

$$x_1(k+1) = u(k) \wedge x_2(k),$$
$$x_2(k+1) = \neg u(k), \tag{12}$$

where $\xi_2(k+1) = \xi_1(k)$ is ignored. Consider the controllability of the system (12) for $T = 2$. From (12) we have

$$x_1(2) = u(1) \wedge (\neg u(0)),$$
$$x_2(2) = \neg u(1). \tag{13}$$

Thus we see that the system is not 2-output-controllable for $C = I_2$, while that the system is T-output-controllable with $T \geq 2$ for both cases of $C = [1\ 0]$ and $C = [0\ 1]$.

Finally, suppose that $x(k) = [\xi_2(k)\ \xi_3(k)]^\top$ and $u(k) = \xi_1(k)$. Then we obtain

$$x_1(k+1) = u(k),$$
$$x_2(k+1) = \neg x_1(k), \tag{14}$$

where $\xi_1(k+1) = \xi_2(k) \wedge \xi_3(k)$ is ignored. Consider the system (14) with $C = I_2$. From (14), we have

$$x_1(2) = u(1),$$
$$x_2(2) = \neg u(0), \tag{15}$$

which implies that system (14) is 2-output-controllable. However, in the case of $T = 1$, we have

$$x_1(1) = u(0),$$
$$x_2(1) = \neg x_1(0), \tag{16}$$

which implies that the system (14) is not 1-output-controllable.

Note that for (11) with $y = x$, we see that the controllability property does not hold due to the fact that $y(T)$ directly depends on $x(0)$. On the other hand, for (13) with $y = x$, $y_1(2)(= x_1(2))$ is adjacent to $u(1)$ and $u(0)$ in the Boolean network, which implies that $y_1(2)$ is arbitrarily given by $u(1)$ and $u(0)$. In a similar way, $y_2(2)(= x_2(2))$ is adjacent to $u(1)$. However, $(y_1(2), y_2(2)) = (1,1)$ cannot be realized by $u(0)$ and $u(1)$ because $y_1(2) = 0$ always holds when $y_2(2) = 1$. These examples are very important in discussing the controllability in a Boolean network, that is, if the Boolean function of $y_i(T)$ includes an initial state $x(0)$, or includes the same input in the outputs at the same time, then the system in question is not T-output-controllable. In the following section, by motivating the above discussion, we will consider to derive a controllability condition.

Remark 1. In the above example, we assume that when some genes are identified as control inputs, the original dynamics of the corresponding genes can be ignored. However, in the case that the corresponding gene has a strong interaction with other genes, this assumption may not be suitable. One of methods for coping with such a case is to add a new gene (node) that works as the control input [10], where it is called an *external control node*. Our approach below can be also applied to this case.

4. Output-Controllability Condition

4.1. Preliminaries. This section presents a sufficient condition for the system (3) to be T-output-controllable in the form of an algorithm.

Consider a simple example given by

$$x_1(k+1) = u(k),$$
$$x_2(k+1) = x_1(k) \wedge (\neg u(k)),$$
$$x_3(k+1) = x_1(k) \wedge x_2(k), \tag{17}$$
$$y_1(k) = x_2(k),$$
$$y_2(k) = x_3(k).$$

This system has the following relation:

$$y_2(2) = x_1(0) \wedge \{u(0) \wedge (\neg u(0))\} = 0. \tag{18}$$

Similarly, we see that $y_2(T) = 0$, $T \geq 2$, hold identically. In Boolean functions, identical equations are in general given by

$$h(a) \wedge (\neg h(a)) \equiv 0, \qquad h(a) \vee (\neg h(a)) \equiv 1, \qquad (19)$$

where $h(\cdot)$ is any Boolean function of a vector of binary variables. Obviously such identities on x_i or u_i affect the controllability in a Boolean network (note that even if $y(T) = x(0) \vee (\neg x(0)) \vee u(0)$, $y(T) \equiv 1$ holds irrespective of $u(0)$).

Let us consider again the Boolean network model (5) of an apoptosis network. If we suppose that $x(0) = x_0 = [0\ 0\ 0]^\top$, $C = [0\ 1\ 0]$ (i.e., $y = x_2$), and $T = 2$, then by a simple calculation, we obtain the following identity:

$$
\begin{aligned}
y(2) &= (x_2(0) \vee \neg u(0)) \wedge (x_2(0) \vee u(0)) \\
&= \neg u(0) \wedge u(0) \qquad (20) \\
&\equiv 0.
\end{aligned}
$$

So in Boolean biological networks, there exists the case that identities are appeared. However, identities may not be appeared in the real biological relevance. The reasons why such identities are appeared are that the state is binarized and that a time-delay of the state is ignored. To overcome the latter point, a temporal Boolean network model $\xi(k + 1) = f_a(\xi(k), \xi(k - 1), \ldots, \xi(k - T))$ has been proposed in [23]. However, identities may appear even in a temporal Boolean network. The output-controllability condition proposed below can be similarly applied to a temporal Boolean network model.

Thus first of all, we will focus on finding such identities in $y(T)$ before discussing a kind of initial condition and a kind of input-independency. This will require the introduction for several symbols.

The following assumption is made.

Assumption 1. The Boolean function f in (3) has no redundant variables.

For example, in the logical function $h(a, b) = a \wedge (b \vee \neg b)$, $h(a, 0) = h(a, 1)$ holds. So b is a redundant variable, and $h(a, b)$ can be rewritten as $h(a) = a$. Any given Boolean function can be changed so as to satisfy Assumption 1: after it is transformed into an appropriate canonical form (e.g., Reed-Muller canonical form (polynomials over the finite field GF(2))), it is easy to eliminate redundant variables by expanding based on four operations over GF(2). Also in the identification of Boolean network models (e.g., see [24]), since the correlations between variables are checked, the Boolean function f in (3) will satisfy Assumption 1 in many cases. By Assumption 1, it is guaranteed that the Boolean function f itself does not include any identities, although $y(T)$ may include some identities. Let p denote the number of the logical NOT appeared in (3), where the logical NOT operators are distinguished when the corresponding terms are different even if the corresponding variables are the same. In addition, consider the fictitious inputs $v_i(k) = 1$, $i = 1, 2, \ldots, p$, which have one-to-one correspondence with the

variables operated by the logical NOT, that is, $\neg x_i$ or $\neg u_i$ in (3). Then the system (3) can be equivalently rewritten as the following system:

$$
\Sigma_v \begin{cases} x(k + 1) = f_v(x(k), u(k), v(k)), \\ y(k) = Cx(k). \end{cases} \qquad (21)
$$

where the Boolean function f_v does not include the logical NOT, and

$$
\begin{aligned}
v(k) &= \begin{bmatrix} v_1(k) & v_2(k) & \cdots & v_p(k) \end{bmatrix}^\top \\
&= \begin{bmatrix} 1 & 1 & \cdots & 1 \end{bmatrix}^\top.
\end{aligned} \qquad (22)
$$

For example, system (17) is rewritten as

$$
\begin{aligned}
x_1(k + 1) &= u(k), \\
x_2(k + 1) &= x_1(k) \wedge (v(k) \oplus u(k)), \\
x_3(k + 1) &= x_1(k) \wedge x_2(k), \qquad (23) \\
y_1(k) &= x_2(k), \\
y_2(k) &= x_3(k),
\end{aligned}
$$

subject to $v(k) = 1$.

Next, consider the adjacency matrix $\Phi \in \{0, 1\}^{(n+m+p) \times (n+m+p)}$ for the directed graph induced by the Boolean network of the system (21). For example, the adjacency matrix for the system (23) is given by

$$
\Phi = \begin{bmatrix} \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \\ 0 \end{bmatrix} & 1 & 1 & 0 & 0 \\ & 0 & 1 & 0 & 0 \\ & 0 & 0 & 0 & 0 \\ & 1 & 0 & 0 & 0 \\ & 1 & 0 & 0 & 0 \end{bmatrix} \begin{matrix} x_1 \\ x_2 \\ x_3 \\ u \\ v \end{matrix}, \qquad (24)
$$
$$
\begin{matrix} x_1 & x_2 & x_3 & u & v \end{matrix}
$$

where if there exists an arc from node i to node j, then the (i, j)-th element of Φ is 1. Hereafter, without loss of generality, the i-th element of $[x^\top\ u^\top\ v^\top]^\top$ is assigned to node i in the directed graph, where $i \in \{1, 2, \ldots, n + m + p\}$. In the case of (24), x_1, x_2, x_3, u, and v are assigned to nodes 1, 2, 3, 4, and 5, respectively. Then in Figure 2, which shows a temporal/spatial network of the system (17), we say that for example, there exists a path between $x_2(2)$ and $u(0)$.

Using the adjacency matrix Φ, we also compute the matrix $\Phi^t C_0^\top C^\top$, $t = 1, 2, \ldots, T$, where

$$
C_0 = \begin{bmatrix} I_n & 0_{n \times (m+p)} \end{bmatrix} \in \{0, 1\}^{n \times (n+m+p)}. \qquad (25)
$$

In the case of the system (17), we have

$$
\Phi C_0^\top C^\top = \begin{bmatrix} 1 & 1 \\ 0 & 1 \\ 0 & 0 \\ 1 & 0 \\ 1 & 0 \end{bmatrix} \begin{matrix} x_1 \\ x_2 \\ x_3 \\ u \\ v \end{matrix}, \qquad (26)
$$
$$
\begin{matrix} y_1 & y_2 \end{matrix}
$$

where $C = [0_{2 \times 1}\ I_2]$.

For the system (21), $\Phi^t C_0^\top C^\top$ expresses whether there exist paths between $y(T)$ and $x(T-t)$, $y(T)$ and $u(T-t)$, or $y(T)$ and $v(T-t)$ for any given T. In the case of (26), we see that $y_2(2)(= x_3(2))$ is adjacent to $x_1(1)$ and $x_2(1)$. In other words, $\Phi^t C_0^\top C^\top$ expresses which elements of $x(T-t)$, $u(T-t)$, and $v(T-t)$ are variables of a Boolean function representing $y_i(T)$. However, note here that from $\Phi^t C_0^\top C^\top$, we cannot specify an explicit form of the Boolean function in question.

Furthermore, the following symbol is used:

$$\begin{bmatrix} X^t \\ U^t \\ V^t \end{bmatrix} = \Phi^t C_0^\top C^\top, \tag{27}$$

where $X^t \in \mathcal{N}^{n\times r}$, $U^t \in \mathcal{N}^{m\times r}$, and $V^t \in \mathcal{N}^{p\times r}$. Let also X_{i_x,j_x}^t, U_{i_u,j_u}^t, and V_{i_v,j_v}^t denote each element of X^t, U^t, V^t, respectively. If $X_{i_x,j_x}^t \geq 1$ holds, then there exist X_{i_x,j_x}^t paths between $y_{j_x}(T)$ and $x_{i_x}(T-t)$. For the state x_{i_x}, $i_x = 1, 2, \ldots, n$, of the system (3), let $\mathcal{P}_x$ express the index set of elements of x_{i_x} operated by the logical NOT as $\neg x_{i_x}$. In a similar way, for the control input u_{i_u}, $i_u = 1, 2, \ldots, m$, of the system (3), let $\mathcal{P}_u$ express the index set of elements of u_{i_u} operated by the logical NOT as $\neg u_{i_u}$. Here, $p = |\mathcal{P}_x| + |\mathcal{P}_u|$ holds. In addition, there is a one-to-one correspondence between each element of $\mathcal{P}_x$, $\mathcal{P}_u$ and the index i_v of v. Let $\nu(i_x)$ and $\nu(i_u)$ express the index i_v of v corresponding to $i_x \in \mathcal{P}_x$ and $i_u \in \mathcal{P}_u$, respectively. In the case of the system (23), $\mathcal{P}_x = \varnothing$, $\mathcal{P}_u = \{1\}$ hold, and for $i_u = 1, \nu(i_u) = 1$ holds.

Finally, we define the following matrices:

$$\begin{aligned} X_0 &= C_0 \Phi^T C_0^\top C^\top \in \mathcal{N}^{n\times r}, \\ U &= \overline{B}\,\overline{\Phi} C_0^\top C^\top \in \mathcal{N}^{mT\times r}, \end{aligned} \tag{28}$$

where

$$\overline{B} = \begin{bmatrix} B & & 0 \\ & \ddots & \\ 0 & & B \end{bmatrix} \in \{0,1\}^{mT\times(n+m+p)T},$$

$$B = \begin{bmatrix} 0_{m\times n} & I_m & 0_{m\times p} \end{bmatrix} \in \{0,1\}^{m\times(n+m+p)}, \tag{29}$$

$$\overline{\Phi} = \begin{bmatrix} \Phi \\ \Phi^2 \\ \vdots \\ \Phi^T \end{bmatrix} \in \mathcal{N}^{(n+m+p)T\times(n+m+p)}.$$

4.2. Proposed Algorithm.

Now we are in a position to propose a T-output-controllability test algorithm. Since this kind of problem is NP-hard [10], we pay our attention on deriving a sufficient condition for the controllability. Although this sufficient condition is given in the form of an algorithm, it is somewhat complex. Thus before describing an algorithm, we describe the outline of the algorithm.

First, we consider a necessary condition for $y(T)$ to include identical equations. From Figure 2 of the example (23), we see that $y_2(2)(= x_3(2))$ in (18), which has no identities, has two paths from $u(0)$, and that $v(0)$ is
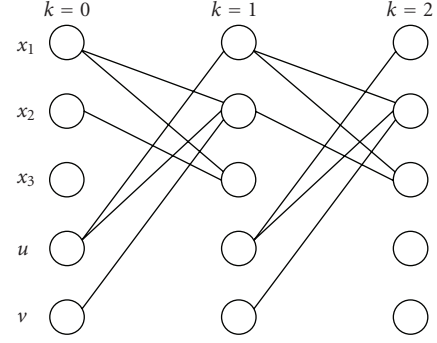

FIGURE 2: Temporal/spatial network of the system (23).

connected to some node on the paths. In this way, if some identical equation exists in $y_j(T)$, there always exist more than 2 paths from $y_j(T)$ to some state and also the logical-NOT operations exist on the paths, which is a necessary condition and not necessarily a sufficient condition. Since it will spend huge time to rigorously specify the existence of identities for a large network, we consider here to exclude the cases satisfying the above necessary condition, that is, we do not determine here the controllability in such cases.

Next, for the system that includes no identical equations, we use a kind of input-independency to determine the controllability. For example, consider the case that neither identity on u nor x exists in $y(T)$ and that $y(T)$ is expressed by

$$\begin{aligned} y_1(T) &= h_1(u_1(0), u_2(3)), \\ y_2(T) &= h_2(u_1(1), u_2(1), u_2(2)) \end{aligned} \tag{30}$$

as a result of recursive calculation (see Section 6 for such an example), where h_1, h_2 are some Boolean functions. This system is obviously T-output controllable because each $y_j(T)$ is expressed by different $u_i(k)$ and no x_0 exists in $y_j(T)$. From the viewpoint of adjacency relation, this implies that there exists no path between $x(0)$ and $y(T)$, there exists at least one path from each $y_j(T)$ to some $u_i(k)$, and each $u_i(k)$ has a path with only one $y_j(T)$ or has no path to any $y_j(T)$. This can be easily found from the adjacency matrix, although it is a sufficient condition for the controllability. This is a rough story of our approach.

The proposed algorithm is given as follows.

Algorithm 1 (T-output-controllability test algorithm).

Part A: Check of the Existence of Identical Equations.

Step 1. Set $t = 1$. Compute X^1, U^1, and V^1.

Step 2. If $T = 1$, go to Step 6. Otherwise set $t = t + 1$. Compute X^t, U^t, and V^t.

Step 3. If there exists (i_x, j_x) such that $X_{i_x,j_x}^t \geq 2$ or (i_u, j_u) such that $U_{i_u,j_u}^t \geq 2$, denote them by (i_x^*, j_x^*) or (i_u^*, j_u^*), respectively, and go to Step 4. Otherwise, go to Step 2 if $t < T$ and go to Step 6 if t = T.

Step 4. If there exists i_x^* such that $i_x^* \in \mathcal{P}_x$ or i_u^* such that $i_u^* \in \mathcal{P}_u$, and $V_{\nu(i_x^*),j_x}^t \geq 1$ or $V_{\nu(i_u^*),j_u}^t \geq 1$ holds, go to Step 8. Otherwise, go to Step 5.

Step 5.

Substep 5.1. Set $j = 1$.

Substep 5.2. If any element of j_x^*-th column or j_u^*-th column in V^j is greater than or equal to 1, go to Step 8. Otherwise, go to Substep 5.3.

Substep 5.3. If $j \leq t - 1$, set $j = j + 1$ and go to Substep 5.2, or else go to Step 2.

Part B: Check of the Independence of Each $y(T)$.

Step 6. If the following conditions hold for the matrices X_0 and U in (28), system (3) is T-output-controllable, or else if only condition (i) does not hold, then go to Step 7. Otherwise go to Step 8.

   (i) $X_0 = 0_{n \times r}$ holds;

   (ii) each column vector of U is a nonzero vector;

   (iii) each row vector of U is a zero vector, or has only one element with a nonzero value.

Step 7. Suppose $x(0) = x_0$ for a given constant vector $x_0 \in \{0,1\}^n$. Let $\mathcal{L}(x_0) \subseteq \{1, 2, \ldots, n\}$ denote the index set of elements of $x(1) = f_v(x_0, u(0), v(0))$ that are constant for any $u(0)$ ($v(0) = 1$). Then if the following condition holds, system (3) is T-output-controllable at x_0. Otherwise, go to Step 8.

   (iv) For $X^{T-1}(= C_0 \Phi^{T-1} C_0^\top C^\top)$, there exists no $l \in \mathcal{L}(x_0)$ satisfying $X_{l,j_x}^{T-1} \geq 1$.

Step 8. This algorithm cannot determine whether the system (3) is T-output-controllable or not (at x_0).

The above algorithm allows us to determine the T-output-controllability of the system as follows.

First, noting that the identical equations have the form in (19), and $x(T)$ is obtained recursively from (21), we see that the identical equations appeared in $x(T)$ always have the form

$$(V_1 \oplus w(k)) \wedge (V_2 \oplus w(k))(\equiv 0), \tag{31}$$

$$(V_1 \oplus w(k)) \vee (V_2 \oplus w(k))(\equiv 1), \tag{32}$$

where $w(k)$ denotes either variable of $x(k)$ or $u(k)$,

$$V_1 = \bigoplus_{(i,j) \in \mathcal{I}_1} v_i(k+j), \qquad V_2 = \bigoplus_{(i,j) \in \mathcal{I}_2} v_i(k+j), \qquad v_i = 1,$$

$$V_1 \oplus w(k) = w(k), \qquad V_2 \oplus w(k) = \neg w(k), \tag{33}$$

and $\mathcal{I}_1$, $\mathcal{I}_2$ are some subsets of the index set $\{(i,j)|i = 1, 2, \ldots, p; \; j = 0, 1, \ldots, T - 1\}$. Then using the forms of (31)

and (32), the following lemma on Part A of Algorithm 1 is obtained.

Lemma 1. *In Step 6, $y(T)$ includes neither identities of* (31) *nor identities of* (32).

Proof. In Step 3, from $X_{i_x,j_x}^t \geq 2$ for some t, $i_x = i_x^*$, and $j_x = j_x^*$, we see that more than 2 paths from $y_{j_x}(T)$ to $x_{i_x}(T - t)$ exist, which is necessary for the identity on $x_{i_x}(T - t)$ to exist (similarly for the case of $U_{i_u,j_u}^t \geq 2$). Thus we next focus on the existence of logical NOT (i.e., v_i) in these paths in Step 4 and Step 5.

Consider the case that the logical NOT (i.e., v_i) corresponding to $x_{i_x^*}(T - t)$ or $u_{i_u^*}(T - t)$ obtained in Step 3 exists in (21), in other words, either $i_x^* \in \mathcal{P}_x$ or $i_u^* \in \mathcal{P}_u$ holds. Then the condition $V_{\nu(i_x^*),j_x}^t \geq 1$ implies that the term $v_{\nu(i_x^*)}(T - t) \oplus x_{i_x^*}(T - t)$ is included in the paths in question, which is a necessary condition for the existence of the identity in $y_{j_x^*}(T)$. Thus we exclude this case (Step 4). (similarly for the case $u_{i_u^*}(T - t)$).

In the other case, from (31), (32), for $v(T - j)$, some $j \in \{1, 2, \ldots, t - 1\}$, to exist in the paths in question is necessary for the existence of identities. If any element of the j_x^*-column or the j_u^*-column of V^j is greater than or equal to 1, some element of $v(T - j)$ exists in the paths in question. Thus we exclude this case (Substep 5.2). Therefore, it follows that $y(T)$ includes no identities in Step 6. □

From Lemma 1, we see that the case that $y(T)$ includes the identities that have the form of (31) or (32) is excluded from the viewpoint of a necessary condition for the identity to exist in $y(T)$. Thus we obtain the following theorem.

Theorem 1. *For a given T, the following statements hold.*

   (i) *the system* (3) *is T-output-controllable if conditions (i), (ii), and (iii) in Step 6 hold subject to Part A of Algorithm 1,*

   (ii) *for a given $x_0 \in \{0,1\}^n$, the system* (3) *is T-output-controllable at x_0 if condition (iv) in Step 7 holds subject to Part A and Step 6.*

Proof. First, the statement (i) is proven for the system satisfying the condition that $y(T)$ includes neither identities of (31) nor identities of (32). From Lemma 1, this condition is satisfied in Step 6. Then condition (i) in Step 6 implies that there exists no path between each element of $x(0)$ and each element of $y(T)$, since the (i, j)-th element of X_0 expresses if a path from $x_i(0)$ to $y_j(T)$ exists or not. On the other hand, note that $(mh + i, j)$-th element of U expresses if a path from $u_i(T - h - 1)$ to $y_j(T)$ exists or not ($h = 0, 1, \ldots, T - 1$). Thus condition (ii) in Step 6 implies that there exists at least one path from each element of $y(T)$ to some $u_i(k)$.

Furthermore, condition (iii) in Step 6 means that the input $u_i(k)$ for each $i \in \{1, 2, \ldots, m\}$ and $k \in \{0, 1, \ldots, T - 1\}$ has a path connected to only one element of $y(T)$ or has no path to any element of $y(T)$. From these conditions, it follows that each $u_i(k)$ affects at most one $y_j(T)$ and not the other $y_h(T)$, $h \neq j$. Hence the value of $y_j(T)$ can be

independently specified by the corresponding $u_i(k)$, which implies that system (21) is T-output-controllable.

Next, the statement (ii) is proven. Since condition (i) in Step 6 does not hold, in this case, there exists a path between some element of $x(0)$ and some element of $y(T)$. On the other hand, condition (iv) in Step 7 guarantees that there exists no path between constant elements of $x(1) = f_v(x_0, u(0), v(0))$ and elements of $y(T)$. Thus $y(T)$ is not affected by the value of x_0. Therefore, from (ii)–(iv), it follows that system (21) is T-output-controllable at x_0. This completes the proof. □

As an example, consider system (17) again. Suppose $T = 2$. The matrices X^1, U^1, V^1 of Step 1 are given by (26), and X^2, U^2, V^2 of Step 2 are

$$\Phi^2 C_0^\top C^\top = \begin{bmatrix} 0 & 1 \\ 0 & 0 \\ 0 & 0 \\ \hline 1 & 2 \\ \hline 0 & 1 \end{bmatrix}. \tag{34}$$

In Step 3, from $U_{1,2}^2 = 2$, we obtain $(i_u^*, j_u^*) = (1,2)$ and $v(i_u^*) = 1$. In Step 4, from $\mathcal{P}_x = \varnothing$ and $\mathcal{P}_u = \{1\}$, we have $i_u^* \in \mathcal{P}_u$ and $V_{1,2}^2 = 1$. So go to Step 8, that is, it is impossible to determine if system (17) is 2-output-controllable. In fact, from (18), $y_2(2)$ includes the identity $u(0) \wedge (\neg u(0)) = 0$. Thus we see that there exists an identical equation.

Let us also consider the case of $y = x_2$, $C = [0\ 1\ 0]$ in the system (17). Then for $T = 2$, we have

$$\begin{aligned} \Phi C_0^\top C^\top &= \begin{bmatrix} 1 & 0 & 0 \mid 1 \mid 1 \end{bmatrix}^\top, \\ \Phi^2 C_0^\top C^\top &= \begin{bmatrix} 0 & 0 & 0 \mid 1 \mid 0 \end{bmatrix}^\top. \end{aligned} \tag{35}$$

From Step 1 → Step 2 → Step 3 → Step 6, we can see that the system (17) is 2-output-controllable. In fact, by simple calculation, the Boolean function of $y(T)(= x_2(2))$ is derived as $y(T) = u(0) \wedge \neg u(1)$.

As for identical equations, the proposed algorithm excludes the case of $\neg h(a) \wedge \neg h(a)$ as well as (19). This is a weak point of this algorithm. Furthermore, consider the following system:

$$\begin{aligned} x_1(k+1) &= x_2(k) \wedge u_2(k) \oplus u_1(k), \\ x_2(k+1) &= x_1(k) \oplus u_2(k), \\ y(k) &= x(k). \end{aligned} \tag{36}$$

This system is T-output-controllable for $T = 1$. However, the proposed algorithm cannot determine whether this system is 1-output-controllable or not; thus there exists a class of systems such that the proposed algorithm cannot determine the controllability. Needless to say, it will not be so easy to cope with various cases stated above due to high nonlinearity of Boolean functions.

While the proposed algorithm includes such disadvantages, one of the main advantages of the algorithm is that the computational complexity of the above algorithm is very small. This will be discussed in the following section.

5. Computational Complexity Analysis

In this section, we discuss the computational complexity of the algorithm proposed in the previous section.

First, let us recall the definition of the symbols used here. The number of the state, the control input, and the output in (3) are denoted by n, m, and r, respectively. The number of the logical NOT appeared in (3) is expressed by p. In addition, $T \in \mathcal{N}$ expresses the control time. Then the following result is obtained.

Lemma 2. *The computational complexity of the proposed algorithm is $O((n + m + p)^3(T - 1) + (n + m + p)nrT)$ for $T \geq 2$, $n, m, p, r \geq 1$.*

Proof. The computation of the proposed algorithm consists of (a) checking each condition of Part A, and (b) checking whether conditions (i) to (iv) hold or not.

First, (b) is considered. The computational complexity to compute Φ^2 is $O((n + m + p)^3)$. So the computational complexity to compute Φ^T and $\overline{\Phi}$ is given by both $O((n + m + p)^3(T - 1))$. Further, the computational complexity to compute the product of $\overline{\Phi}$ and $C_0^\top C^\top$ is $O((n + m + p)nrT)$. So by simple calculation, the computational complexity of U is obtained as $O((n + m + p)^3(T - 1) + (n + m + p)nrT)$. The computational complexity of generating X_0 is obviously less than the case of U. Therefore, the computational complexity to compute X_0 and U is $O((n + m + p)^3(T-1) + (n+m+p)nrT)$, which also includes the computational complexity to check conditions (i) to (iv) in Steps 6 and 7 for given X_0 and U.

Next, (a) is considered. The matrices X^t, U^t, V^t are obtained directly from $\Phi C_0^\top C^\top$, and the computational complexity of Step 5 is $O(prT)$. As a result, since the computational complexity of each checking in Part A is $O((n + m + p)^2(T - 1)) + O(prT)$, the computational complexity of Part A is less than $O((n + m + p)^3(T - 1) + (n + m + p)nrT)$.

Therefore, the computational complexity of the proposed algorithm is given by $O((n + m + p)^3(T-1)+(n+m+p)nrT)$. □

From Lemma 2, we see that the proposed algorithm is a polynomial-time algorithm. Furthermore, the computational time for performing the proposed algorithm is evaluated by numerical experiments, where the total computational time in Part B is measured because from the proof of Lemma 2 we see that the computational complexity of Part B is dominant. So the adjacency matrices to be evaluated are generated randomly for each $l(= n + m)$, where $n = m = l/2$, $p = 0$ are given. The results are shown in Table 1, where MATLAB on the computer with the Intel Core 2 Duo CPU 3.0 GHz and the 2 GB memory is used. In Table 1, the worst computational time implies the worst value among 100 cases randomly selected for each l. From Table 1, we see that the proposed algorithm can be applied to relatively large-scale Boolean network models.

6. Application to Neurotransmitter Signaling Pathway

In this section, the proposed algorithm is applied to a Boolean network model of interaction pathway between the glutamatergic and dopaminergic receptors in Figure 3, which has been proposed in [20]. In this pathway, exocytosis, by which a cell directs the contents of secretory vesicles out of the cell membrane, is regulated, depending on the value of neurotransmitters such as dopamine and glutamate. Then it is important from the viewpoint of synaptic plasticity to consider whether exocytosis can be controlled by regulating other elements. In the Boolean network model of Figure 3, the dopamine (neurotransmitter, ξ_2) is synthesized by tyrosine hydroxylase (ξ_1) and catabolized by COMT (ξ_3). The dopamine binds to the dopamine receptor 1 (DRD1, ξ_4) and the dopamine receptor 2 (DRD2, ξ_5). DRD1 stimulates adenylate cyclase (ξ_6) to activate protein kinase A (ξ_7), which activates DARPP32 (ξ_{11}). DARPP32 inhibits protein phosphatase 1 (ξ_{12}). By inhibition of protein phosphatase 1, activation of protein kinase A, and presence of the glutamate (ξ_{13}), the glutamate receptor (ξ_{14}) is activated to elevate the concentration of the intracellular calcium (ξ_9). On the other hand, DRD2 inactivates adenylate cyclase and activates phospholipase C (ξ_8) in order to elevate the concentration of the intracellular calcium. The intracellular calcium activates calcineurin (ξ_{10}), which inhibits DARPP32. Also, the intracellular calcium activates packaging proteins (ξ_{15}) and finally exocytosis (ξ_{16}). The process of exocytosis of the glutamate receptor expresses one of events in synaptic plasticity, that is, if exocytosis is activated, then the neurotransmitter is secreted out of the cell membrane. In this model, the concentration of the above reactants is expressed by a binary variable ξ_i, that is, $\xi_i = 1$ if it is high, otherwise $\xi_i = 0$. Then the state equations of this system are given as

$$
\begin{aligned}
\xi_1(k+1) &= \xi_1(k), \\
\xi_2(k+1) &= \xi_1(k) \wedge \neg\xi_3(k), \\
\xi_3(k+1) &= \xi_2(k), \\
\xi_4(k+1) &= \xi_2(k), \\
\xi_5(k+1) &= \xi_2(k), \\
\xi_6(k+1) &= \xi_4(k) \wedge \neg\xi_5(k), \\
\xi_7(k+1) &= \xi_6(k), \\
\xi_8(k+1) &= \xi_5(k), \\
\xi_9(k+1) &= \xi_8(k) \vee \xi_{14}(k), \\
\xi_{10}(k+1) &= \xi_9(k), \\
\xi_{11}(k+1) &= \neg\xi_{10}(k) \wedge \xi_7(k), \\
\xi_{12}(k+1) &= \neg\xi_{11}(k), \\
\xi_{13}(k+1) &= \xi_{13}(k), \\
\xi_{14}(k+1) &= \xi_7(k) \wedge \neg\xi_{12}(k) \wedge \xi_{13}(k), \\
\xi_{15}(k+1) &= \xi_9(k), \\
\xi_{16}(k+1) &= \xi_{15}(k).
\end{aligned}
\tag{37}
$$

Table 1: Computational time of the proposed algorithm ($T = 10$).

l	$n(= m)$	Worst comp. time [sec]
100	50	0.1
200	100	0.5
300	150	1.5
400	200	3.3
500	250	6.3
600	300	10.2
700	350	15.9
800	400	23.1
900	450	32.4
1000	500	43.6

From Figure 3, we see that this Boolean network includes at least four loops, for example, the loop of ξ_2, ξ_4, ξ_6, and ξ_5, the loop of ξ_{11}, ξ_{12}, ξ_{14}, ξ_9, and ξ_{10}, and so forth. In synaptic plasticity, it is required that the binary value of ξ_{16} expressing exocytosis can be arbitrarily controlled. Furthermore, phospholipase C (ξ_8) is a kind of enzymes that cleaves phospholipids and as a result protein kinase C as well as calcium (ξ_9) are activated. The former, protein kinase C, which works outside of the network in Figure 3, is one of key enzymes in signal transduction pathways. Thus since phospholipase C affects the other significant network, it will be important to simultaneously control the value of ξ_8 and the value of ξ_{16}. Therefore ξ_8 and ξ_{16} are regarded as the output, that is, $y = [\xi_8 \ \xi_{16}]^\top$. In addition, we assume that ξ_8 and ξ_{16} cannot be directly controlled.

For a fixed dimension of u and the fixed output $y = [\xi_8 \ \xi_{16}]^\top$, all combinations of ξ_i, $i = 1, 2, \ldots, 7, 9, \ldots, 15$, are considered as the control inputs, which we call the input-combinations. Then for a given T, the proposed algorithm is applied to the system of the form (3) obtained for each input-combination of ξ_i. It is remarked that depending on the choice of the kind of control inputs, there exist several cases to which the polynomial-time algorithm proposed in [10] cannot be applied due to the graph-structure constraints. Furthermore, it is also remarked that even for fixed control inputs, the controllability problem is NP-hard. So the problem of finding efficient control inputs that make the system controllable is further harder than this problem.

By applying our algorithm to the case of each input-combination of ξ_i and each fixed T, we obtain, for example, the following results. In the case of $\dim u(k) = 2$ and $T = 5$, we can find that among $_{14}C_2 (= 92)$ input-combinations, there exist at least 6 input-combinations of ξ_i that make the system 5-output-controllable. In this way, since the proposed algorithm for each input-combination is very efficient, for example, the computation time via the proposed algorithm is about 10 [sec] for Boolean networks with 600 nodes (see Table 1) and $T = 10$, it enables us to verify the controllability condition for a certain number of input-combinations within a practical time; for example, about 3 [hours] will be required for 1000 input-combinations of a Boolean network with 600 nodes.
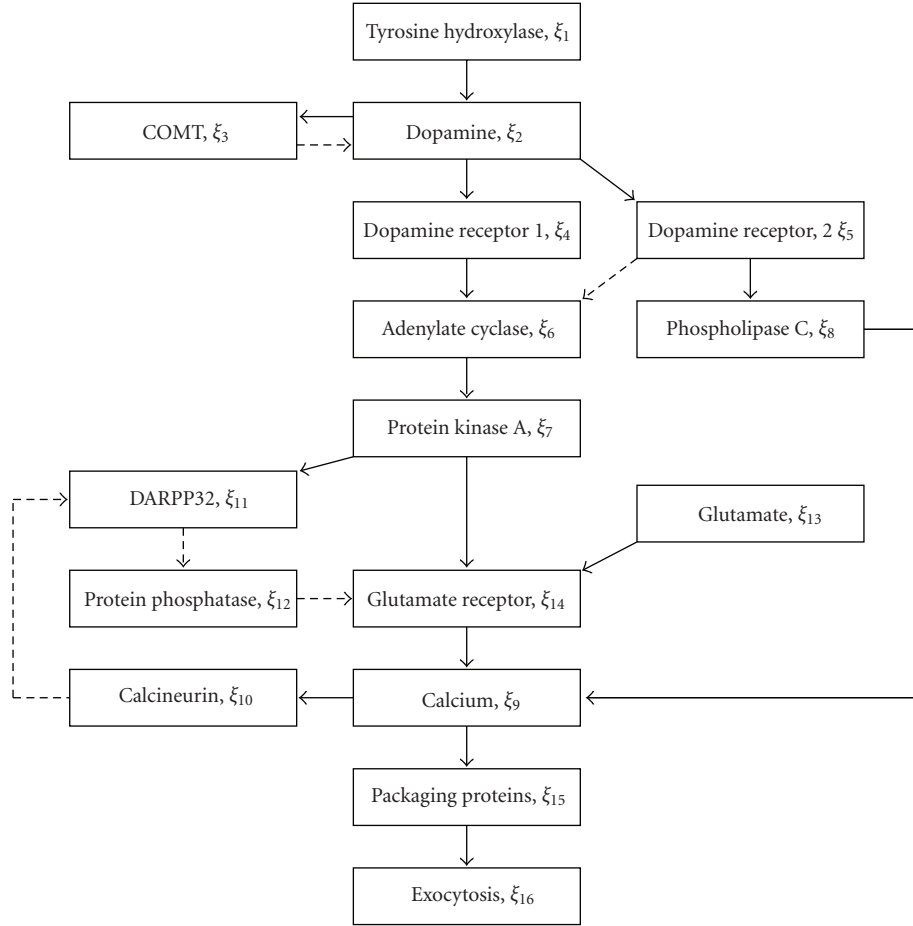
FIGURE 3: Simplified model of the interaction pathway between the glutamatergic and dopaminergic receptors. Activation (solid), Inhibition (broken).

In the case of $\dim u(k) = 4$ and $T = 6$, we can also find controllable control inputs among $_{14}C_4 (= 1001)$ input-combinations. For example, we obtain as one of combinations of $x(k) \in \{0, 1\}^{12}$ and $u(k) \in \{0, 1\}^4$ that make the system 6-output-controllable

$$x(k) = \begin{bmatrix} \xi_1(k) & \xi_3(k) & \xi_4(k) & \xi_5(k) & \xi_6(k) & \xi_8(k) \end{bmatrix}$$
$$\begin{matrix} \xi_9(k) & \xi_{11}(k) & \xi_{12}(k) & \xi_{14}(k) & \xi_{15}(k) & \xi_{16}(k) \end{matrix}\Big]^{\top},$$
$$(38)$$

$$u(k) = \begin{bmatrix} \xi_2(k) & \xi_7(k) & \xi_{10}(k) & \xi_{13}(k) \end{bmatrix}^{\top}. \qquad (39)$$

It is remarked that the polynomial-time algorithm proposed in [10] cannot be applied to the system with the state (38) and the input (39) because the network includes the two loops, that is, the loop of ξ_2, ξ_4, ξ_6, and ξ_5, and the loop of ξ_7, ξ_{11}, ξ_{12}, and ξ_{14}. Furthermore, based on the above result the Boolean function of $y(6)$ can be derived as

$$y_1(6) = u_1(4), \qquad (40)$$

$$y_2(6) = u_1(1) \lor (u_2(2) \land \neg u_3(0) \land u_4(2)) \qquad (41)$$

which implies that the value of $y(6)$ can be freely given by control inputs, for example, (a) $y(6) = [0 \ 0]^{\top}$ for $u_1(4) = 0$, $u_1(1) = 0$, $u_2(2) = 0$, $u_3(0) = 1$, $u_4(2) = 0$, and (b) $y(6) = [1 \ 1]^{\top}$ for $u_1(4) = 1$, $u_1(1) = 1$, $u_2(2) = 0$, $u_3(0) = 1$, $u_4(2) = 0$.

Finally, we discuss the control input sequence realizing the desired output values. One of criticisms in control of Boolean networks is to assume that the value of the control input can be arbitrarily given at each time. In many biological systems, this assumption is not always satisfied, and input constraints are frequently imposed. One of input constraints is that the value of the control input is given as a constant within a certain sufficiently long time period. Although it is one of future works to explicitly deal with such an input constraint, based on the proposed algorithm, we may also find a constant-valued sequence of control inputs for which the desired values of outputs are obtained. For example, in (40) and (41), let us consider to find a control input sequence satisfying $y_1(6) = 0$ and $y_2(6) = 1$. Since $u_1(4) = 0$, $u_1(1) = 0$, $u_2(2) = 1$, $u_3(0) = 0$, and $u_4(2) = 1$ are obtained as one of solutions, it is remarked that the following control inputs are given as any binary value: $u_1(0)$, $u_1(2)$, $u_1(3)$, $u_1(5)$, and $u_2(0)$, $u_2(1)$, $u_2(3)$, $u_2(4)$, $u_2(5)$, and $u_3(1)$, $u_3(2)$,

$u_3(3)$, $u_3(4)$, $u_3(5)$, and $u_4(0)$, $u_4(1)$, $u_4(3)$, $u_4(4)$, $u_4(5)$. This allows us to give the values of the control input sequences as a constant, that is, $u_1(k) = 0$, $u_2(k) = 1$, $u_3(k) = 0$, $u_4(k) = 1$, $k = 0, 1, \ldots, 5$. Thus the proposed algorithm helps us to find a practically useful control input sequence. Furthermore, this kind of degree of freedom in control inputs may be used for the optimal control problem. Once we can determine control input variables by our algorithm, we can use a tool for finding optimal control input sequences, which have been developed in hybrid control theory, for example, [25, 26]. It is expected that such an analysis will provide one of guidelines in experimental approaches to the control problem of biological networks.

7. Conclusion

In this paper, the controllability analysis for biological networks expressed by a Boolean network model with control nodes (inputs) and controlled nodes (outputs) has been discussed. First, a sufficient condition for the Boolean network model to be output-controllable has been derived by exploiting an adjacency matrix of its network graph. The obtained condition, which is given in the form of an algorithm, can be checked in polynomial time with respect to the state/input dimensions and the control time period; it will be one of the powerful tools that can provide some clues for finding effective control inputs to control a large-scale biological network. Next, by PC-based numerical experiments, it has been shown that the proposed method is applicable to large-scale Boolean networks with at least 1000 nodes. Finally, the proposed method has been applied to the Boolean network model expressing a neurotransmitter signaling pathway, and has shown that it is controllable with respect to both exocytosis and phospholipase C when appropriate control inputs are used.

There are many interesting open problems to be addressed in the future. It is one of the most important issues to characterize a class of Boolean networks to which our algorithm can be applied as a necessary and sufficient condition. In addition, extensions to the case of systems with input constraints and uncertainty are also one of the significant topics.

Acknowledgment

References

[1] M. Khammash, C. J. Tomlin, and M. Vidyasagar, Eds., "Joint special issue on systems biology," *IEEE Transactions on Automatic Control & IEEE Transactions on Circuits and Systems I*, 2008.

[2] H. Kitano, "Computational systems biology," *Nature*, vol. 420, no. 6912, pp. 206–210, 2002.

[3] H. Kitano, "Cancer as a robust system: implications for anticancer therapy," *Nature Reviews Cancer*, vol. 4, no. 3, pp. 227–235, 2004.

[4] G. Ferrari-Trecate and J. Lygeros, "Workshop on Hybrid Systems Biology," in *Proceedings of the 45th IEEE Conference on Decision and Control/Workshop Hybrid Systems Biology*, 2006.

[5] H. De Jong, "Modeling and simulation of genetic regulatory systems: a literature review," *Journal of Computational Biology*, vol. 9, no. 1, pp. 67–103, 2002.

[6] S.-I. Azuma, E. Yanagisawa, and J.-I. Imura, "Controllability analysis of biosystems based on piecewise-affine systems approach," *IEEE Transactions on Automatic Control*, vol. 53, no. 1, pp. 139–152, 2008.

[7] C. Belta, J. Schug, T. Dang et al., "Stability and reachability analysis of a hybrid model of luminescence in the marine bacterium *Vibrio fischeri*," in *Proceedings of the 40th IEEE Conference on Decision and Control (CDC '01)*, pp. 869–874, 2001.

[8] S. A. Kauffman, "Metabolic stability and epigenesis in randomly constructed genetic nets," *Journal of Theoretical Biology*, vol. 22, no. 3, pp. 437–467, 1969.

[9] S. A. Kauffman, *The Origins of Order: Self-Organization and Selection in Evolution*, Oxford University Press, Oxford, UK, 1993.

[10] T. Akutsu, M. Hayashida, W.-K. Ching, and M. K. Ng, "Control of Boolean networks: hardness results and algorithms for tree structured networks," *Journal of Theoretical Biology*, vol. 244, no. 4, pp. 670–679, 2007.

[11] M. Aldana, "Boolean dynamics of networks with scale-free topology," *Physica D*, vol. 185, no. 1, pp. 45–66, 2003.

[12] A. Datta, A. Choudhary, M. L. Bittner, and E. R. Dougherty, "External control in Markovian genetic regulatory networks," *Machine Learning*, vol. 52, no. 1-2, pp. 169–191, 2003.

[13] A. Datta, A. Choudhary, M. L. Bittner, and E. R. Dougherty, "External control in Markovian genetic regulatory networks: the imperfect information case," *Bioinformatics*, vol. 20, no. 6, pp. 924–930, 2004.

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

[15] C. J. Langmead and S. K. Jha, "Symbolic approaches to finding control strategies in Boolean networks," in *Proceedings of the 6th Asia-Pacific Bioinformatics Conference*, pp. 307–319, 2008.

[16] S. Martin, Z. Zhang, A. Martino, and J.-L. Faulon, "Boolean dynamics of genetic regulatory networks inferred from microarray time series data," *Bioinformatics*, vol. 23, no. 7, pp. 866–874, 2007.

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

[18] 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.

[19] I. Shmulevich and W. Zhang, "Binary analysis and optimization-based normalization of gene expression data," *Bioinformatics*, vol. 18, no. 4, pp. 555–565, 2002.

[20] S. Gupta, S. S. Bisht, R. Kukreti, S. Jain, and S. K. Brahmachari, "Boolean network analysis of a neurotransmitter signaling pathway," *Journal of Theoretical Biology*, vol. 244, no. 3, pp. 463–469, 2007.

[21] M. Chaves, "Methods for qualitative analysis of genetic networks," in *Proceedings of the European Control Conference*, pp. 671–676, 2009.

[22] L. Tournier and M. Chaves, "Uncovering operational interactions in genetic networks using asynchronous Boolean dynamics," *Journal of Theoretical Biology*, vol. 260, no. 2, pp. 196–209, 2009.

[23] A. Silvescu and V. Honavar, "Temporal Boolean network models of genetic networks and their inference from gene expression time series," *Complex Systems*, vol. 13, pp. 54–71, 2001.

[24] T. Akutsu, S. Miyano, and S. Kuhara, "Identification of genetic networks from a small number of gene expression patterns under the boolean network model," in *Proceedings of the Pacific Symposium on Biocomputing*, pp. 17–28, 1999.

[25] A. Bemporad and M. Morari, "Control of systems integrating logic, dynamics, and constraints," *Automatica*, vol. 35, no. 3, pp. 407–427, 1999.

[26] H. Nakayama, H. Tanaka, and T. Ushio, "The formulation of the control of an expression pattern in a gene network by propositional calculus," *Journal of Theoretical Biology*, vol. 240, no. 3, pp. 443–450, 2006.

EUSIPCO 2011

19th European Signal Processing Conference
August 29- September 2, 2011, Barcelona (Spain)

Fotografia © Turisme de Barcelona / J. Trullàs

Preliminary call for papers

The 2011 European Signal Processing Conference (EUSIPCO-2011) is the nineteenth in a series of conferences promoted by the European Association for Signal Processing (EURASIP, www.eurasip.org). This year edition will take place in Barcelona, capital city of Catalonia (Spain), and will be jointly organized by the Centre Tecnològic de Telecomunicacions de Catalunya (CTTC) and the Universitat Politècnica de Catalunya (UPC).

EUSIPCO-2011 will focus on key aspects of signal processing theory and applications as listed below. Acceptance of submissions will be based on quality, relevance and originality. Accepted papers will be published in the EUSIPCO proceedings and presented during the conference. Paper submissions, proposals for tutorials and proposals for special sessions are invited in, but not limited to, the following areas of interest.

Areas of Interest

- Audio and electro-acoustics.
- Design, implementation, and applications of signal processing systems.
- Multimedia signal processing and coding.
- Image and multidimensional signal processing.
- Signal detection and estimation.
- Sensor array and multi-channel signal processing.
- Sensor fusion in networked systems.
- Signal processing for communications.
- Medical imaging and image analysis.
- Non-stationary, non-linear and non-Gaussian signal processing.

Submissions

Procedures to submit a paper and proposals for special sessions and tutorials will be detailed at www.eusipco2011.org. Submitted papers must be camera-ready, no more than 5 pages long, and conforming to the standard specified on the EUSIPCO 2011 web site. First authors who are registered students can participate in the best student paper competition.

Important Deadlines:

Proposals for special sessions	15 Dec 2010
Proposals for tutorials	18 Feb 2011
Electronic submission of full papers	**21 Feb 2011**
Notification of acceptance	23 May 2011
Submission of camera-ready papers	6 Jun 2011

Webpage: www.eusipco2011.org

Organizing Committee

Honorary Chair
Miguel A. Lagunas (CTTC)

General Chair
Ana I. Pérez-Neira (UPC)

General Vice-Chair
Carles Antón-Haro (CTTC)

Technical Program Chair
Xavier Mestre (CTTC)

Technical Program Co-Chairs
Javier Hernando (UPC)
Montserrat Pardàs (UPC)

Plenary Talks
Ferran Marqués (UPC)
Yonina Eldar (Technion)

Special Sessions
Ignacio Santamaría (Unversidad de Cantabria)
Mats Bengtsson (KTH)

Finances
Montserrat Nájar (UPC)

Tutorials
Daniel P. Palomar
(Hong Kong UST)
Beatrice Pesquet-Popescu (ENST)

Publicity
Stephan Pfletschinger (CTTC)
Mònica Navarro (CTTC)

Publications
Antonio Pascual (UPC)
Carles Fernández (CTTC)

Industrial Liaison & Exhibits
Angeliki Alexiou
(University of Piraeus)
Albert Sitjà (CTTC)

International Liaison
Ju Liu (Shandong University-China)
Jinhong Yuan (UNSW-Australia)
Tamas Sziranyi (SZTAKI -Hungary)
Rich Stern (CMU-USA)
Ricardo L. de Queiroz (UNB-Brazil)

EURASIP
European Association for Signal Procesing

CTTC
Centre Tecnològic de Telecomunicacions de Catalunya

UPC