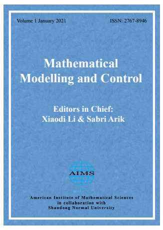
Mathematical Modelling and Control

http://www.aimspress.com/journal/mmc



MMC, 3(3): 256–266
DOI:10.3934/mmc.2023022
Received: 05 September 2022
Revised: 29 March 2023
Accepted: 02 May 2023
Published: 15 September 2023

Research article

Reference trajectory output tracking for Boolean control networks with controls in output

Zejiao Liu[1], **Yu Wang**[1], **Yang Liu**[1,2,*] **and Qihua Ruan**[3]

[1] School of Mathematical Sciences, Zhejiang Normal University, Jinhua 321004, China

[2] Key Laboratory of Intelligent Education Technology and Application of Zhejiang Province, Zhejiang Normal University, Jinhua, 321004, China

[3] College of Mathematics and Finance, Putian University, Putian, 351100, China

* **Correspondence:** Email: liuyang@zjnu.edu.cn.

Abstract: This article investigates the reference trajectory output tracking issue of Boolean control networks (BCNs) that have controls in the output. Firstly, to solve the problem, some necessary and sufficient conditions are obtained. The tracking problem is studied from the perspective of set and matrix calculation. Next, an algorithm for determining whether the output tracking issue is solvable is proposed. Furthermore, the controller design algorithm satisfying the solvability condition is given. Using our methods, we can track some trajectories that cannot be tracked in BCNs without controls in output. In addition, for better application in practice, the corresponding changes in the network transition matrix and output matrix under state, transition, and input constraints are considered. Finally, some examples are presented to illustrate the validity of our results.

Keywords: reference trajectory output tracking; the semi-tensor product of matrices; Boolean control networks

1. Introduction

Since Kauffman introduced a binary network called Boolean network (BN) to study the evolution behavior in genetic regulatory networks (GRNs) [1], it has garnered considerable interest from biologists, systems scientists, and others. In a BN, variables are used to represent genes with values of 1 or 0, where 1 implies that the gene is active, and 0 implies that the gene is inactive. When there are external disturbances regulations, BNs are naturally extended to BCNs. In addition, probabilistic Boolean networks (PBNs) and switching Boolean networks (SBNs) are investigated due to the switching and random phenomena in GRNs [2,3].

In fact, BNs (BCNs) with logical function forms are nonlinear networks. Thus, traditional discrete-time linear system theory cannot be used to investigate research problems. Cheng *et al.* [4] invented the semi-tensor product (STP) of matrices to convert the logical dynamical system into a typical discrete-time linear system. Some methods and ideas in linear system theory can be applied to BNs (BCNs) by using the STP, which further helps scientists analyze and control GRNs for disease intervention [5] or work on feedback shift registers [6] and so on [7–9]. Since then, various properties of BNs have been investigated, such as controllability, observability, stabilization and so on [10–20]. Furthermore, as a helpful tool, the STP method also is used in hybrid dynamic systems and fuzzy systems [21–26].

In the practical GRNs, the state evolution is complicated and the measuring equipment has limitation [27]. Therefore, it is an effective solution for scientists to measure the output and render the system output to track the desired signals to help study the system's dynamics. It is also of great significance in the application of robots and flight control [28–30]. By reviewing the literature, we obtain that the current studies on output tracking issues of BCNs can be mainly divided into the two following categories according to the objectives of output tracking: track a constant reference signal and track a time-varying reference trajectory [27, 31–35]. Methods for designing the state feedback control law were described in [27] to let the system's output track a constant reference signal stably after a finite time. In [31], the switching sequence is found by combining the method of set stabilization, so that the SBNs can track a given a given constant reference signal. [32] constructed an auxiliary system to help the outputs of a BCN track the outputs of a reference system. Since it may cost a lot to control over an infinite horizon, [33] considered tracking a given reference output trajectory. In summary, the output tracking problem is an inevitable and crucial issue for GRNs.

In addition, the output tracking problem has other practical significance. For example, the motion of ships carrying heavy cargo needs to track a desired line reserved before, thereby minimizing financial burden [36]. Furthermore, considering that the output tracking research might help control the spread of future epidemics, we mainly care about tracking a given finite length of time-varying reference output trajectory. Moreover, the idea of adding controls to the output is motivated by the research problem investigated in [37]. In the study of a functional system related to avalanche warning, context-alert of context model combined with terrain temperature, snow height, and accelerometer as the total input variables has a specific effect on the output of functional system [37]. Hence, we add controls to the system output to describe these system models. It is worth highlighting that the authors in [33] studied the output tracking problem of BCNs. Nonetheless, the method in [33] fails to solve the problem in some cases. To our knowledge, no study has been done considering the controls in the output to solve the output tracking problem. The system considered in this paper has controls in both state transition and output models. Therefore, to a certain extent, the method discussed in this paper can realize the output tracking problem that cannot be realized in [33]. In the following, we highlight the main contributions of this paper:

- From the perspectives of set and matrix, for the BCN system with output having controls, we give some equivalent conditions to solve the reference trajectory output tracking issue.
- Based on the obtained theorem, the authors construct an algorithm to determine whether the output tracking issue is solvable.
- If the problem can be solved, we develop an algorithm to obtain a feasible control sequence, allowing the system to track the reference output trajectory in a finite range.

The main structure of this paper is as follows. We introduce the preliminaries in Section 2. The main theorems for the solvability of output tracking issues are obtained in Section 3. Finally, some biological examples are proposed to verify our results in Section 4. A concise conclusion is shown in Section 5.

2. Preliminaries

First, for the convenience of subsequent description, some basic notations are given here.

- $\mathcal{D} := \{0, 1\}$, real numbers are denoted by $\mathbb{R}$ and the set of positive integers is denoted by $\mathbb{Z}_+$.
- $\Delta_q := \{\delta_q^i | 1 \le i \le q\}$, where δ_q^i represents an q-dimensional column vector with the ith element being 1 and others are 0.
- The ith column and ith row of matrix Q are denoted by $\mathrm{Col}_i(Q)$ and $\mathrm{Row}_i(Q)$, respectively.
- We call a matrix $Q \in M_{m\times n}$ logical matrix, if each column vector of it has only one element 1, and the other elements are 0. In this paper, denote the set of $m \times n$-dimensional logical matrices by $\mathcal{L}_{m\times n}$. Besides, $\mathcal{B}_{m\times n}$ denotes the set of $m \times n$ Boolean matrices, with all elements taking value from $\mathcal{D}$.
- For a logical matrix $A = [\delta_m^{i_1}\ \delta_m^{i_2}\ \cdots\ \delta_m^{i_n}]$, its abbreviation is $A = \delta_m[i_1\ i_2\ \cdots\ i_n]$.
- $[\epsilon,\ \xi] := \{\epsilon, \epsilon + 1, \epsilon + 2, \cdots, \xi\}$, where $\epsilon,\ \xi \in \mathbb{Z}_+$.
- $\mathbf{1}_k$ or $\mathbf{0}_k$ is the k-dimensional column vector whose elements are all equal to 1 or 0, respectively.
- If $\alpha(t) = m_1\delta_{2^n}^{i_1} + m_2\delta_{2^n}^{i_2} + \cdots + m_k\delta_{2^n}^{i_k}$, where $m_1, m_2, \cdots, m_k \in \mathbb{Z}_+$, $\Xi(\alpha(t)) := \{\delta_{2^n}^{i_1}, \delta_{2^n}^{i_2}, \cdots, \delta_{2^n}^{i_k}\}$. On the contrary, if one set $\Omega = \{\delta_{2^n}^{i_1}, \delta_{2^n}^{i_2}, \cdots, \delta_{2^n}^{i_k}\}$,

- then $\Psi(\Omega) := \sum_{j=1}^{k} \delta_{2^n}^{i_j}$. To describe concisely, denote $\Psi(\Omega) = \delta_{2^n}^{i_1,i_2,\cdots,i_k}$.
- For two vectors $\rho = [\rho_1\ \rho_2\ \cdots\ \rho_s]^\top \in \mathbb{R}^s$, $\sigma = [\sigma_1\ \sigma_2\ \cdots\ \sigma_s]^\top \in \mathbb{R}^s$. We denote the element-wise multiplication of them by

$$\rho \odot \sigma := [\rho_1\sigma_1\ \rho_2\sigma_2\ \cdots\ \rho_s\sigma_s]^\top.$$

2.1. Semi-tensor product

In this section, we introduce the definition of the STP and how to transform BNs from logical expressions to algebraic forms.

Definition 2.1. *[4] For two matrices $P \in M_{a\times b}$ and $Q \in M_{c\times d}$, the semi-tensor product of them is defined by*

$$P \ltimes Q = (P \otimes I_{l/b})(Q \otimes I_{l/c}),$$

where $l = lcm(b, c)$ is the least common multiple of b and c, and $\otimes$ is the Kronecker product.

When $b = c$, the STP of matrix is consistent with ordinary matrix multiplication. In this article, we will omit "$\ltimes$" without affecting the results to facilitate reading.

Before using the STP to obtain the algebraic expression of BCNs, we show equivalence between Boolean variables and their vector forms. Here, 1 is equivalent to δ_2^1 and 0 is equivalent to δ_2^2.

Lemma 2.1. *[4] For a logical function $g(\mathbf{x}_1, \cdots, \mathbf{x}_n) : \mathcal{D}^n \to \mathcal{D}$, after changing all arguments $\mathbf{x}_i$ to their vector form x_i, i.e., $x_i \in \Delta_2$, $i \in [1, n]$, it has an equivalent algebraic form $g(\mathbf{x}_1, \cdots, \mathbf{x}_n) = L_g x_1 \cdots x_n$, where the structure matrix $L_g \in \mathcal{L}_{2\times 2^n}$ can be uniquely determined by g.*

2.2. Algebraic representation of BCNs

Consider the logical representation of a BCN is abbreviated as

$$\mathbf{x}_i(t + 1) = g_i(\mathbf{u}_1(t), \cdots, \mathbf{u}_m(t), \mathbf{x}_1(t), \cdots, \mathbf{x}_n(t)), \quad (2.1)$$

with its output system being

$$\mathbf{y}_j(t) = h_j(\mathbf{u}_1(t), \cdots, \mathbf{u}_m(t), \mathbf{x}_1(t), \cdots, \mathbf{x}_n(t)), \quad (2.2)$$

where $\mathbf{x}_i(t), \mathbf{u}_k(t), \mathbf{y}_j(t) \in \mathcal{D}$, $i \in [1, n]$, $k \in [1, m]$, $j \in [1, p]$ are logical variables. $g_i : \mathcal{D}^{m+n} \to \mathcal{D}, i \in [1, n]$, $h_j : \mathcal{D}^{m+n} \to \mathcal{D}, j \in [1, p]$ are logical functions. Then, after changing the logical form of variables to the vector form, we define $x(t) = \ltimes_{i=1}^n x_i(t) \in \Delta_{2^n}, u(t) = \ltimes_{k=1}^m u_k(t) \in \Delta_{2^m}$. According to Lemma 2.1, we can transform equation (2.1) into

$$x_i(t + 1) = M_i u(t)x(t), \quad (2.3)$$

where structure matrices are specified as $M_i \in \mathcal{L}_{2\times 2^{m+n}}, i \in [1, n]$. Likewise, define $y(t) = \ltimes_{j=1}^p y_j(t)$. Then the outputs are

$$y_j(t + 1) = N_j u(t)x(t), \quad (2.4)$$

where $N_j \in \mathcal{L}_{2\times 2^{m+n}}$, $j \in [1, p]$. Therefore, we can determine the algebraic form of BCN (2.1) and its output (2.2) as

$$\begin{cases} x(t + 1) = L \ltimes u(t) \ltimes x(t), \\ y(t) = H \ltimes u(t) \ltimes x(t), \end{cases} \quad (2.5)$$

where L, which can be obtained by $L = M_1 * M_2 * \cdots * M_n$, is referred to as the network transition matrix. Similarly, H is called an output matrix which can be expressed as $H = N_1 * N_2 * \cdots * N_p$, where "$*$" is the Khatri-Rao product of matrices.

Remark 2.1. *Without the loss of generality, in the equations* (2.1) *and* (2.2)*, the evolution of the state and the output, respectively, in the BCN model we give, is related to the same control sequence. If the state and output are affected by different control variables, we can still combine all the control variables with a dummy matrix [4]. Then, it will make the control sequences the same and larger dimensional than before for the convenience of subsequent studies.*

3. Main Results

3.1. Problem formulation

Based on the above STP work, we can analyze the algebraic form of BCNs to study the output tracking issue. For simplicity, denote a control sequence u^t by $u^t = \{u(0), u(1), \cdots, u(t)\}$. Thus, the output of the system (2.5) under u^t is shown as $y(t, x(0), u^t)$. Then, there is $y(t, x(0), u^t) = Hu(t)Lu(t-1)Lu(t-2)\cdots Lu(0)x(0)$. Since we are investigating the problem of reference trajectory output tracking, there needs to be an assigned reference output trajectory y^P:

$$y^o(1) = \delta_{2^p}^{y_1}, y^o(2) = \delta_{2^p}^{y_2}, \cdots, y^o(P) = \delta_{2^p}^{y_P}.$$

Next, we will give the definition of reference trajectory output tracking and the solvable conditions for the output tracking issue will be discovered using the properties of BCNs.

Definition 3.1. *[33] Consider BCN* (2.5) *with a given initial state* $x(0)$ *and a reference output trajectory* y^P*. The (reference trajectory) output tracking issue is solvable, if we can find a control sequence* u^P *such that* $y(t; x(0), u^t) = y^o(t)$, $\forall t \in [1, P]$.

We first regard $u(t)x(t)$ in (2.5) as a control-state pair. For convenience of subsequent description, let $u(t) = \delta_{2^m}^{k_t}$, $x(t) = \delta_{2^n}^{i_t}$. Then $u(t)x(t) = \delta_{2^{m+n}}^{j_t}$, where $j_t = 2^n(k_t - 1) + i_t$. For the output $y^o(t) = \delta_{2^p}^{y_t}$, the set of $u(t)x(t)$ that can realize output tracking is

$$\Omega(t) = \{u(t)x(t) = \delta_{2^{m+n}}^{j_t} | Col_{j_t}(H) = \delta_{2^p}^{y_t}\}, \forall\, t \in [1, P].$$

We construct sets $X(t)$, $\Lambda(t)$ and $\Gamma(t)$ alternatively as shown below to address the output tracking issue, where $t \in [1, P]$.

First, the state set obtained from state $x(0)$ through one step is $X(1) = \{x(1)|x(1) = Lu(0)x(0), u(0) \in \Delta_{2^m}\}$. Define the set $\Lambda(t) = \{\tilde{u}(t)x(t), \tilde{u}(t) \in \Delta_{2^m}, x(t) \in X(t)\}$, $\forall t \in [1, P]$. $\Gamma(t)$ represents the intersection of $\Lambda(t)$ and $\Omega(t)$, i.e.,

$$\Gamma(t) = \Lambda(t) \cap \Omega(t), \forall\, t \in [1, P]. \tag{3.1}$$

It is essential to mention that for $t \in [2, P]$

$$X(t) = \{x(t)|x(t) = Lu(t-1)x(t-1), u(t-1)x(t-1) \in \Gamma(t-1)\}. \tag{3.2}$$

Remark 3.1. $\Omega(t)$ *is the control-state pairs set whose elements can produce the desired output according to H.* $X(t)$ *is the state set under control sequence which meets the condition of tracking, from the perspective of network transition matrix L of dynamical equation. To unify the dimensions of the two sets, we define the set* $\Lambda(t)$.

3.2. The solvability of the output tracking issue

Following the definition of sets $X(t)$, $\Lambda(t)$ and $\Gamma(t)$, a theorem can be obtained to determine the solvability of the output tracking issue from the perspective of the set.

Theorem 3.1. *Given an initial state* $x(0) = \delta_{2^n}^{i_0}$*, a positive integer P and a reference output trajectory* y^P*, the output tracking issue of BCN* (2.5) *is solvable if and only if*

$$\Gamma(P) \neq \emptyset.$$

Proof. (Sufficiency) For any $t' \in [1, P - 1]$, if $\Gamma(t') = \emptyset$, by calculation it has $X(t' + 1) = \{x(t' + 1)|x(t' + 1) = Lu(t')x(t'), u(t')x(t') \in \Gamma(t')\} = \emptyset$. Similarly, it can be obtained that for any $t \in [t' + 1, P]$, $X(t) = \{x(t)|x(t) = Lu(t-1)x(t-1)\} = \emptyset$ always holds. It shows that $\Lambda(t) = \emptyset$ and $\Gamma(t) = \emptyset$, which is in contradiction to $\Gamma(P) \neq \emptyset$. Therefore, the condition $\Gamma(P) \neq \emptyset$ implies that $\Gamma(t) \neq \emptyset$, $\forall t \in [1, P]$. Then, we can choose a $u(P)x(P) \in \Gamma(P)$ and denote them by $u(P) = \delta_{2^m}^{k_P}$, $x(P) = \delta_{2^n}^{i_P}$, respectively. Since $x(P) \in X(P)$, there exists $u(P - 1)x(P - 1) \in \Gamma(P - 1)$, where $u(P - 1) = \delta_{2^m}^{k_{P-1}}$, $x(P) = \delta_{2^n}^{i_{P-1}}$, such that $Lu(P - 1)x(P - 1) = x(P) = \delta_{2^n}^{i_P}$ holds. Similarly, we can find a series of control-state pairs $\{u(1)x(1), u(2)x(2), \cdots, u(P)x(P)\}$, which satisfies $u(t)x(t) \in \Gamma(t)$ and $Lu(t)x(t) = x(t + 1)$, $\forall t \in [1, P - 1]$. In addition, for $x(0) = \delta_{2^n}^{i_0}$, we only need to determine $u(0) = \delta_{2^m}^{k_0}$ so that $Lu(0)x(0) = x(1)$. Hence, there exists a control sequence $u^P = \{u(0) = \delta_{2^m}^{k_0}, u(1) = \delta_{2^m}^{k_1}, \cdots, u(P) = \delta_{2^m}^{k_P}\}$ such that for any $t \in [1, P]$, $u(t)x(t) \in \Gamma(t)$, then $y(t; x(0), u^t) = y^o(t)$ holds. Thus, the solvable goal can be achieved.

(Necessity) If $\Gamma(P) = \emptyset$, based on (3.1), we have $\Lambda(P) \cap \Omega(P) = \emptyset$. Then, for any $x(P) \in X(P)$, $\tilde{u}(P)x(P)$ constructed by $\tilde{u}(P) \in \Delta_{2^m}$ does not belong to $\Omega(P)$. Therefore, $y(P) = H\tilde{u}(P)x(P) \neq y^o(P)$, which breaks the requirement that the output tracking issue of BCN (2.5) is solvable. □

Theorem 3.1 shows that $\forall t \in [1, P]$, $\Gamma(t) \neq \emptyset$ means that there exists $u(t)x(t)$ satisfying both dynamical update condition and output tracking condition. It determines that the output tracking goal of BCN (2.5) can be achieved from the perspective of the set. Next, we consider giving another theorem to realize the goal through vector expression from the perspective of matrix calculation.

Denote

$$w(t) = \Psi(\Omega(t)) \in \mathcal{B}_{2^{m+n} \times 1}, \tag{3.3}$$

which is the vector form of the set $\Omega(t)$ including all the control-state pairs that can produce output $y^o(t)$. Define

$$\gamma(t) = w(t) \odot (\mathbf{1}_{2^m} \ltimes L\gamma(t-1)), t \in [1, P], \qquad (3.4)$$

with $\gamma(0) := \mathbf{1}_{2^m} \ltimes x(0)$. Then, we have the following necessary and sufficient criterion for output tracking using vector expression.

Theorem 3.2. *Given an initial state* $x(0) = \delta_{2^n}^{i_0}$ *and a positive integer P, the output tracking issue of BCN* (2.5) *is solvable if and only if* $\gamma(P) \neq \mathbf{0}_{2^{m+n}}$.

Proof. For sufficiency, if there is $t_1 \in [1, P-1]$, such that $\gamma(t_1) = \mathbf{0}_{2^{m+n}}$, then by (3.4), the equation $\gamma(t) = \mathbf{0}_{2^{m+n}}, t \in [t_1+1, P]$ holds, which is in contradiction with the condition $\gamma(P) \neq \mathbf{0}_{2^{m+n}}$. Thus, it concludes that $\gamma(t) \neq \mathbf{0}_{2^{m+n}}, t \in [1, P]$. For $\gamma(P) \neq \mathbf{0}_{2^{m+n}}$, we can take $u(P)x(P) = \delta_{2^{m+n}}^{j_P} \in \Xi(\gamma(P))$, and denote them by $u(P) = \delta_{2^m}^{k_P}, x(P) = \delta_{2^n}^{i_P}$, respectively. According to the construction of $\gamma(t)$ in (3.4), it implies that $y(P) = Hu(P)x(P) = y^o(P)$. Then, we can choose $u(t-1)x(t-1) \in \Xi(\gamma(t-1))$ satisfying that $Lu(t-1)x(t-1) = x(t) = \delta_{2^n}^{i_t}$, where $u(t-1) = \delta_{2^m}^{k_{t-1}}$, $x(t-1) = \delta_{2^n}^{i_{t-1}}$, $t = P, P-1, \cdots, 2$. It also holds that $y(t) = Hu(t)x(t) = y^o(t)$. For $t = 1$, $u(0)$ is determined by satisfying $Lu(0)x(0) = x(1) = \delta_{2^n}^{i_1}$. Hence, there is a $u^P = \{u(0) = \delta_{2^m}^{k_0}, u(1) = \delta_{2^m}^{k_1}, \cdots, u(P) = \delta_{2^m}^{k_P}\}$ so that $\forall t \in [1, P], y(t; x(0), u^t) = y^o(t)$ holds, which means that there is one way that the output tracking goal of BCN (2.5) can be resolved. As for necessity, it is removed here since it is analogous to Theorem 3.1. □

Theorem 3.1 gives the necessary and sufficient conditions for the solvability of output tracking problem from the set viewpoint, which helps us to understand. In Theorem 3.2, the criterion is given by vector method, which is convenient for practical calculation and judgment. Then, Algorithm 1 is given below to illustrate the general steps for determining whether the output tracking issue can be solved.

3.3. Tracking control design

After we investigate whether the output tracking issue for a reference trajectory in a given finite time can be solved, we consider finding a control sequence satisfying the condition that BCN (2.5) is solvable, such that the outputs can achieve the given reference output trajectory.

Algorithm 1 Determine the solvability of output tracking issue.

Input: $x(0) = \delta_{2^n}^{i_0}$, P, and the reference output trajectory y^P with a known BCN (2.5)

```
Initialize γ(0) = 1_{2^m} ⋉ δ_{2^n}^{i_0}, t = 1.
while t = 1, ··· , P do
    Calculate the corresponding Ω(t) and w(t) based on y^o(t) = δ_{2^p}^{y_t}.
    Calculate γ(t) = w(t) ⊙ (1_{2^m} ⋉ Lγ(t − 1)).
    if γ(t) = 0_{2^{m+n}} then
        the output tracking issue is unsolvable.
    else
        t ← t + 1.
    end if
else
    the output tracking issue of BCN (2.5) is solvable.
end while
```

The basic idea to solve the problem is to calculate $u(t)$ and $x(t)$ backward by $\gamma(t), t \in [1, P]$. Obviously, the control-state pairs that can realize output tracking may be not unique at each time, and our goal is to find a feasible control sequence so that the output trajectory can track the reference output trajectory. Algorithm 2 gives the design method of the controllers.

Remark 3.2. *It is worth to note that Algorithm 2 is based on the Algorithm 1. We need to calculate all* $\gamma(t)$, $t \in [1, P]$ *and determine that* $\gamma(P) \neq \mathbf{0}_{2^{m+n}}$. *Therefore, Step 3 in Algorithm 2 is a way to select the feasible control u and x. Finally, the control sequence that can achieve the output tracking is determined based on the* $\gamma(t)$, $t \in [1, P]$, *calculated in Algorithm 1.*

Furthermore, it has been proved that solving control problems of BCNs using the STP is an NP-hard problem and causes exponential computational burden. In particular, the worst-case computational complexity involved in determining the problem's solvability (Algorithm 1) is $O(2^n P)$, *and designing the control sequence (Algorithm 2) is* $O(2^{n+m} P)$.

Algorithm 2 Get a feasible control sequence $\{u^{*}(0), u^{*}(1), \cdots, u^{*}(P)\}$, such that BCN (2.5) realize the output tracking.

Input: $x(0) = \delta_{2^n}^{i_0}$, P, and the reference output trajectory y^P with a known BCN (2.5)

Determine that whether the problem can be solved according to Algorithm 1.
if it is solvable **then**
  Randomly choose $u^{*}(P)x^{*}(P) = \delta_{2^{m+n}}^{j_P} \in \Xi(\gamma(P))$, and denote them by $u^{*}(P) = \delta_{2^m}^{k_P}, x^{*}(P) = \delta_{2^n}^{i_P}$.
  while $t = P, P-1, \cdots, 1$ **do**
    Choose $u^{*}(t-1)x^{*}(t-1) \in \Xi(\gamma(t-1))$ satisfying that $Lu^{*}(t-1)x^{*}(t-1) = x^{*}(t) = \delta_{2^n}^{i_t}$. Denote $u^{*}(t-1) = \delta_{2^m}^{k_{t-1}}$, $x^{*}(t-1) = \delta_{2^n}^{i_{t-1}}$.
    $t \leftarrow t-1$
  end while
  Select $u^{*}(0) = \delta_{2^m}^{k_0}$, which satisfies $Lu^{*}(0)x(0) = x^{*}(1)$.
else end

Proposition 3.1. *Given* $x(0) = \delta_{2^n}^{i_0}$, *under the control sequence* $\{u^{*}(0), u^{*}(1), \cdots, u^{*}(P)\}$ *obtained in Algorithm 2, the state trajectory* $\{x^{*}(1) = \delta_{2^n}^{i_1}, x^{*}(2) = \delta_{2^n}^{i_2}, \cdots, x^{*}(P) = \delta_{2^n}^{i_P}\}$ *will produce the desired reference output trajectory* y^P.

Proof. From Algorithm 2, we can find that

$$x^{*}(t) = \begin{cases} Lu^{*}(0)x(0) & t = 1, \\ Lu^{*}(t-1)x^{*}(t-1) & t \in [2, P]. \end{cases} \tag{3.5}$$

For any $t \in [1, P]$, there is

$$\begin{aligned} y(t) &= Hu^{*}(t)x^{*}(t) \\ &= Hu^{*}(t)Lu^{*}(t-1)x^{*}(t-1) \\ &= Hu^{*}(t)Lu^{*}(t-1)Lu^{*}(t-2)\cdots Lu^{*}(1)Lu^{*}(0)x(0). \end{aligned}$$

Since $u^{*}(t)x^{*}(t) \in \Xi(\gamma(t))$ holds for all $t \in [1, P]$, it implies that $u^{*}(t)x^{*}(t) \in \Omega(t)$. Therefore, it can be obtained that $y(t) = y^o(t), \forall t \in [1, P]$. □

In daily life, constraints are everywhere. Taking the treatment of patients by doctors as an example, the state of patients or cells can be regarded as the state of the system, and the prescription drugs issued by doctors can be regarded as the control of the system. The purpose of doctors' treatment of patients is to hope that patients will enter a continuous healthy state, so some unhealthy and sick states should be avoided [38]. In some cases, the type of drugs will be determined according to the state of the patient. For example, when the patient is in the state of drinking alcohol, the doctor will not consider taking cephalosporin anti-inflammatory drugs, which is the transition constraint considered here [39]. In addition, if some drugs were developed a long time ago, and with the increase of medical level, we find the use of these drugs may have undesirable side effects on the human body, then doctors will not consider prescribing these types of drugs. For example, furazolidone as an effective drug, is usually used to treat gastrointestinal diseases such as dysentery and enteritis caused by bacteria and protozoa. However, scientists found that it has potentially carcinogenic properties [40]. Hence, doctors won't use compound preparation containing furazolidone now, and it can be one of the examples for the input control constraints. Therefore, in controllers' design, we should also consider changes in state, transition and input constraints.

(1) State constraint

When solving the output tracking issue of the system and designing the controllers, if a state $x(t) = \delta_{2^n}^{i}$ should be avoided, then the ith row of matrix L should be

$$Row_i(L) = \mathbf{0}_{2^{m+n}}^{\top}.$$

At this point, we do not consider any transition from other states to $x(t) = \delta_{2^n}^{i}$.

(2) Transition constraint

In a BCN, a transition constraint forbids the control input $u(t)$ for a state $x(t)$. For example, if the transition from the state $x(t) = \delta_{2^n}^{i}$ using control input $u(t) = \delta_{2^m}^{k}$ is forbidden, then we have

$$Col_{2^n(k-1)+i}(L) = \mathbf{0}_{2^n}, \text{ and } Col_{2^n(k-1)+i}(H) = \mathbf{0}_{2^p}.$$

(3) Input constraint

The input constraint means there are some controls that we cannot take. Suppose that $u(t) = \delta_{2^m}^{k}$ has to be forbidden, then the k-th block of L and H will be null matrices, that is

$$L_k := L \ltimes \delta_{2^m}^k = \mathbf{0}_{2^n\times 2^n}, \text{ and } H_k := H \ltimes \delta_{2^m}^k = \mathbf{0}_{2^p\times 2^n}.$$

4. Example

Example 4.1. *Consider BCN in [33]:*

$$\begin{cases} x_1(t+1) = \neg u_1(t) \wedge (x_2(t) \vee x_3(t)), \\ x_2(t+1) = \neg u_1(t) \wedge u_2(t) \wedge x_1(t), \\ x_3(t+1) = \neg u_1(t) \wedge (u_2(t) \vee (u_3(t) \wedge x_1(t))), \\ y_1(t) = (u_1(t) \wedge u_3(t)) \vee (\neg u_1(t) \wedge u_2(t)) \vee x_1(t), \\ y_2(t) = (\neg u_1(t) \vee u_2(t)) \vee (u_2(t) \wedge u_3(t)) \wedge x_2(t). \end{cases} \tag{4.1}$$

This is a simplified BCN model of lac operon in Escherichia coli. Applying the STP method, the following algebraic form can be obtained:

$$\begin{cases} x(t+1) = Lu(t)x(t), \\ y(t) = Hu(t)x(t), \end{cases} \tag{4.2}$$

where $x(t) = \ltimes_{i=1}^3 x_i(t) \in \Delta_8$, $u(t) = \ltimes_{k=1}^3 u_k(t) \in \Delta_8$, $y(t) = \ltimes_{j=1}^2 y_j(t) \in \Delta_4$. *Besides, the network transition matrix and the output matrix are*

$$\begin{aligned} L = \delta_8[&8\ 8\ 8\ 8\ 8\ 8\ 8\ 8 \mid 8\ 8\ 8\ 8\ 8\ 8\ 8\ 8 \\ &8\ 8\ 8\ 8\ 8\ 8\ 8\ 8 \mid 8\ 8\ 8\ 8\ 8\ 8\ 8\ 8 \\ &1\ 1\ 1\ 5\ 3\ 3\ 3\ 7 \mid 1\ 1\ 1\ 5\ 3\ 3\ 3\ 7 \\ &3\ 3\ 3\ 7\ 4\ 4\ 4\ 8 \mid 4\ 4\ 4\ 8\ 4\ 4\ 4\ 8], \end{aligned}$$

$$\begin{aligned} H = \delta_4[&1\ 1\ 1\ 1\ 1\ 1\ 1\ 1 \mid 1\ 1\ 1\ 1\ 3\ 3\ 3\ 3 \\ &2\ 2\ 2\ 2\ 2\ 2\ 2\ 2 \mid 2\ 2\ 2\ 2\ 4\ 4\ 4\ 4 \\ &1\ 1\ 1\ 1\ 1\ 1\ 1\ 1 \mid 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1 \\ &1\ 1\ 1\ 1\ 3\ 3\ 3\ 3 \mid 1\ 1\ 1\ 1\ 3\ 3\ 3\ 3]. \end{aligned}$$

At the beginning, we assume that $x(0) = \delta_8^2$, *and give two reference output trajectories as:*

Table 1. Output trajectory 1.

t	1	2	3	4
y_1^o	1	0	0	0
y_2^o	0	1	0	0
y^o	δ_4^2	δ_4^3	δ_4^4	δ_4^4

Table 2. Output trajectory 2.

t	1	2	3	4
y_1^o	0	1	0	0
y_2^o	1	0	0	0
y^o	δ_4^3	δ_4^2	δ_4^4	δ_4^4

To better simulate the GRNs, we assume that there are three types of constraints in (4.2)*:*

(1) State constraint: State δ_8^1 *should be avoided, which means that* $Row_1(L) = \mathbf{0}_{64}^\top$.

(2) Transition constraint: For state $x(t) = \delta_8^6$, *it is prohibited to use control input* $u(t) = \delta_8^3$. *Then, the two matrices become* $L_3 = \delta_8[8\ 8\ 8\ 8\ 8\ 0\ 8\ 8]$, $H_3 = \delta_4[2\ 2\ 2\ 2\ 2\ 0\ 2\ 2]$. *With a slight abuse of notation, we use* δ_i^0 *here to represent* $\mathbf{0}_i$.

(3) Input constraint: Considering the specification about the concentration of extra cellular lactose, the control input $u = \ltimes_{k=1}^3 u_k = \delta_8^2$ *and* $u = \ltimes_{k=1}^3 u_k = \delta_8^6$ *are forbidden. Then, we have* $L_2 = L_6 = \mathbf{0}_{8\times 8}$, $H_2 = H_6 = \mathbf{0}_{4\times 8}$.

Under the three types of constraints, the new matrices are expressed as

$$\begin{aligned} \hat{L} = \delta_8[&8\ 8\ 8\ 8\ 8\ 8\ 8\ 8 \mid 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0 \\ &8\ 8\ 8\ 8\ 8\ 0\ 8\ 8 \mid 8\ 8\ 8\ 8\ 8\ 8\ 8\ 8 \\ &0\ 0\ 0\ 5\ 3\ 3\ 3\ 7 \mid 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0 \\ &3\ 3\ 3\ 7\ 4\ 4\ 4\ 8 \mid 4\ 4\ 4\ 8\ 4\ 4\ 4\ 8], \end{aligned}$$

$$\begin{aligned} \hat{H} = \delta_4[&1\ 1\ 1\ 1\ 1\ 1\ 1\ 1 \mid 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0 \\ &2\ 2\ 2\ 2\ 2\ 0\ 2\ 2 \mid 2\ 2\ 2\ 2\ 4\ 4\ 4\ 4 \\ &1\ 1\ 1\ 1\ 1\ 1\ 1\ 1 \mid 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0 \\ &1\ 1\ 1\ 1\ 3\ 3\ 3\ 3 \mid 1\ 1\ 1\ 1\ 3\ 3\ 3\ 3]. \end{aligned}$$

The following is to determine whether the output tracking issue can be solved for the given output trajectories 1 and 2. If it is solvable, we can give the corresponding controllers' design. Initialize $\gamma(0) = \mathbf{1}_8 \ltimes \delta_8^2$, *and it has* $\gamma(0) = \delta_{64}^{2,10,18,26,34,42,50,58}$ *by calculation.*

We first consider whether the system can track the specified reference output trajectory 1.

(1) For $y^o(1) = \delta_4^2$, *according to* $\hat{H}$, *we can get* $\Omega(1) = \{\delta_{64}^{17}, \delta_{64}^{18}, \delta_{64}^{19}, \delta_{64}^{20}, \delta_{64}^{21}, \delta_{64}^{23}, \delta_{64}^{24}, \delta_{64}^{25}, \delta_{64}^{26}, \delta_{64}^{27}, \delta_{64}^{28}\}$ *and* $w(1) = \delta_{64}^{17,18,19,20,21,23,24,25,26,27,28}$. *Then, we have*

$$\gamma(1) = w(1) \odot (\mathbf{1}_{2^m} \ltimes \hat{L}\gamma(0)) = \delta_{64}^{19,20,27,28} + 3\delta_{64}^{24}.$$

(2) For $y^o(2) = \delta_4^3$, we have $\Omega(2) = \{\delta_{64}^{53}, \delta_{64}^{54}, \delta_{64}^{55}, \delta_{64}^{56}, \delta_{64}^{61}, \delta_{64}^{62}, \delta_{64}^{63}, \delta_{64}^{64}\}$ and $w(2) = \delta_{64}^{53,54,55,56,61,62,63,64}$. Then, it holds that

$$\gamma(2) = w(2) \odot (\mathbf{1}_{2^m} \ltimes \hat{L}\gamma(1)) = 7\delta_{64}^{56,64}.$$

(3) For $y^o(3) = y^o(4) = \delta_4^4$, it holds that $\Omega(3) = \Omega(4) = \{\delta_{64}^{29}, \delta_{64}^{30}, \delta_{64}^{31}, \delta_{64}^{32}\}$, and $w(3) = w(4) = \delta_{64}^{29,30,31,32}$. Furthermore, we have

$$\gamma(3) = w(3) \odot (\mathbf{1}_{2^m} \ltimes \hat{L}\gamma(2)) = 14\delta_{64}^{32},$$

$$\gamma(4) = w(4) \odot (\mathbf{1}_{2^m} \ltimes \hat{L}\gamma(3)) = 14\delta_{64}^{32}.$$

It can be concluded that the output tracking issue of the system is solvable for reference output trajectory 1 since $\gamma(4) \neq \mathbf{0}_8$. Then, we can use the method in Algorithm 2 to find a feasible control sequence to realize output tracking for the output trajectory 1. The specific process is as follows. First, since $P = 4$ and $\gamma(4) = 14\delta_{64}^{32}$, we can only get that $u^*(4)x^*(4) = \delta_{64}^{32} \in \Xi(\gamma(4))$. Hence, by decomposition, we have $u^*(4) = \delta_8^4$ and $x^*(4) = \delta_8^8$. Next, we need to choose $u^*(3)x^*(3) \in \Xi(\gamma(3))$, satisfying $\hat{L}u^*(3)x^*(3) = x^*(4) = \delta_8^8$. Then, we have $u^*(3)x^*(3) = \delta_{64}^{32}$ with $u^*(3) = \delta_8^4$ and $x^*(3) = \delta_8^8$. Similarly, we can obtain a control input sequence that meets the output tracking condition represented by $\{u^*(0) = \delta_8^1, u^*(1) = \delta_8^3, u^*(2) = \delta_8^7, u^*(3) = \delta_8^4, u^*(4) = \delta_8^4\}$, with corresponding state sequence denoted by $\{x^*(0) = \delta_8^2, x^*(1) = \delta_8^8, x^*(2) = \delta_8^8, x^*(3) = \delta_8^8, x^*(4) = \delta_8^8\}$. Note that this is only a feasible situation we have proposed, and it is not unique.

Analogously, for reference output trajectory 2, we can obtain that

$$\gamma(1) = 3\delta_{64}^{56,64},\ \gamma(2) = 6\delta_{64}^{24},\ \gamma(3) = 6\delta_{64}^{32},\ \gamma(4) = 6\delta_{64}^{32}.$$

It determines that the system can track reference output trajectory 2 since $\gamma(4) \neq \mathbf{0}_8$. Furthermore, the input sequence that meets the output tracking condition can be $\{u^*(0) = \delta_8^1, u^*(1) = \delta_8^7, u^*(2) = \delta_8^3, u^*(3) = \delta_8^4, u^*(4) = \delta_8^4\}$.

In [33], there does not have any control in the output system. The system can realize output tracking under output trajectory 1, but fails under output trajectory 2. As one of the innovations of this paper, we add control input to the output system. We can find in this example that it makes the system realize output tracking under two reference output trajectories.

Example 4.2. *Consider a modified lactose operon model in Escherichia coli with 5 nodes [41]:*

$$\begin{cases} x_1(t+1) = x_3(t) \wedge u_1(t), \\ x_2(t+1) = x_1(t), \\ x_3(t+1) = x_3(t) \vee (x_4(t) \wedge x_2(t)), \\ x_4(t+1) = x_5(t) \vee (x_4(t) \wedge \neg x_2(t)), \\ x_5(t+1) = x_1(t) \vee u_2(t), \\ y_1(t) = x_1(t) \vee u_1(t), \\ y_2(t) = x_2(t) \vee u_2(t). \end{cases} \tag{4.3}$$

Based on the STP method, we can transfer the model to the form as (2.5) with $x(t) = \ltimes_{i=1}^5 x_i(t) \in \Delta_{32}$, $u(t) = \ltimes_{k=1}^2 u_k(t) \in \Delta_4$, $y(t) = \ltimes_{j=1}^2 y_j(t) \in \Delta_4$, where $L = \delta_{32}[1\ 3\ 1\ 3\ 17\ 19\ 21\ 23\ \cdots\ 26\ 26\ 26\ 28\ 30\ 30\ 30\ 32]$ and $H = \delta_4[1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ \cdots\ 4\ 4\ 4\ 4\ 4\ 4\ 4\ 4]$. Next, we suppose that there are some constraints:

(1) State constraint: State δ_{32}^{32} is avoided and $Row_{32}(L) = \mathbf{0}_{128}^\top$.

(2) Transition constraint: For state $x(t) = \delta_{32}^3$, the control input $u(t) = \delta_4^1$ is limited, so we will have $Col_3(L) = \mathbf{0}_{32}$, and $Col_3(H) = \mathbf{0}_4$.

(3) Input constraint: Suppose that the control input $u = \ltimes_{k=1}^2 u_k = \delta_4^2$ is forbidden. Finally, the matrices are changed to $\hat{L} = \delta_{32}[1\ 3\ 0\ 3\ 17\ 19\ 21\ 23\ \cdots\ 26\ 26\ 26\ 28\ 30\ 30\ 30\ 0]$ and $\hat{H} = \delta_4[1\ 1\ 0\ 1\ 1\ 1\ 1\ 1\ \cdots\ 4\ 4\ 4\ 4\ 4\ 4\ 4\ 4]$.

Here, we are going to check under the condition that the initial state is $x(0) = \delta_{32}^{20}$, whether the output tracking issue can be solvable with the given output trajectory $y^o(1) = \delta_4^1, y^o(2) = \delta_4^2, y^o(3) = \delta_4^1$. First, according to Algorithm 1, it has $\gamma(0) = \mathbf{1}_4 \ltimes \delta_{32}^{20} = \delta_{128}^{20,52,84,116}$. Then, calculated by the while loop, we can obtain $\gamma(1) = \delta_{128}^{11,27,28,43,59,60,75,91,92,107,123,124}$, $\gamma(2) = \delta_{128}^{105,107}$, $\gamma(3) = 2\delta_{128}^{17}$. It shows that the output tracking issue of BCN (4.3) is solvable. Hence, we can use Algorithm 2 to get a feasible control sequence such that BCN (4.3) achieves output tracking.

First of all, because of $\gamma(P) = \gamma(3) = 2\delta_{128}^{17}$, it can be divided to $u^*(3)x^*(3) = \delta_4^1 \ltimes \delta_{32}^{17}$. Then, we aim to choose $u^*(2)x^*(2) \in \Xi(\gamma(2))$ such that $\hat{L}u^*(2)x^*(2) = x^*(3) = \delta_{32}^{17}$. By calculation, it obtains $u^*(2) = \delta_4^4, x^*(2) = \delta_{32}^9$. Furthermore, we can get $u^*(1) = \delta_4^1, x^*(1) = \delta_{32}^{27}$ and

$u^*(0) = \delta_4^3$. Therefore, using the control sequence $\{u^*(0) = \delta_4^3, u^*(1) = \delta_4^1, u^*(2) = \delta_4^4, u^*(3) = \delta_4^1\}$, the output tracking issue based on the given reference trajectory can be solved.

5. Conclusions

By using the STP, for the BCNs with controls in output, the reference trajectory output tracking issue is investigated. Some results to solve the output tracking issue are proposed and an algorithm is designed to judge the solvability. Moreover, the controllers' design algorithm is given by calculation when we have determined that the output tracking issue is solvable. To better apply the results to daily life, the corresponding changes in the network transition matrix and output matrix under some constraints simulated from reality are considered. In the end, the effectiveness of our results is shown by some examples. An important direction for future research is to investigate output tracking problem by reinforcement learning method. In addition, we can study the problems considered in this paper in BNs with time delay or switching signal.

Acknowledgment

This work was jointly supported by the Natural Science Foundation of Zhejiang Province of China under Grant Nos. LR20F030001 and LD19A010001, and the National Natural Science Foundation of China under Grant 62173308.

Conflict of interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

1. S. Kauffman, Metabolic stability and epigenesis in randomly constructed genetic nets, *J. Theor. Biol.,* **22** (1969), 437–467. http://doi.org/10.1016/0022-5193(69)90015-0
2. I. Shmulevich, E. Dougherty, S. Kim, W. Zhang, Probabilistic Boolean networks: a rule-based uncertainty model for gene regulatory networks, *Bioinformatics,* **18** (2002), 261–274. http://doi.org/10.1093/bioinformatics/18.2.261
3. T. Sun, X. Sun, Y. Gao, P. Sun, Stabilizability analysis of logical networks with switching signal and control input, *Nonlinear Analysis-Hybrid Systems,* **36** (2020), 100875. http://doi.org/10.1016/j.nahs.2020.100875
4. D. Cheng, H. Qi, Z. Li, *Analysis and control of Boolean networks: a semi-tensor product approach*, London: Springer Science & Business Media, 2011.
5. J. Zhong, D. Ho, J. Lu, Q. Jiao, Pinning controllers for activation output tracking of Boolean network under one-bit perturbation, *IEEE T. Cybernetics,* **49** (2019), 3398–3408. http://doi.org/10.1109/tcyb.2018.2842819
6. J. Lu, B. Li, J. Zhong, A novel synthesis method for reliable feedback shift registers via Boolean networks, *Sci. China Inform. Sci.,* **64** (2021), 152207. http://doi.org/10.1007/s11432-020-2981-4
7. D. Cheng, Z. Ji, J. Feng, S. Fu, J. Zhao, Perfect hypercomplex algebras: semi-tensor product approach, *Math. Model. Contr.,* **1** (2021), 177–187.
8. D. Cheng, Y. Li, J. Feng, J. Zhao, On numerical/non-numerical algebra: semi-tensor product method, *Mathematical Modelling and Control,* 2021.
9. Y. Li, J. Zhu, Cascading decomposition of Boolean control networks: a graph-theoretical method, *Frontiers of Information Technology* & *Electronic Engineering,* **21** (2020), 304–315. http://doi.org/10.1631/fitee.1900422
10. S. Wang, J. Feng, J. Zhao, J. Xia, Controllability decomposition of dynamic-algebraic Boolean control networks, *Int. J. Control,* **93** (2020), 1684–1695. http://doi.org/10.1080/00207179.2018.1527040
11. Y. Zhao, Y. Liu, Output controllability and observability of mix-valued logic control networks, *Math. Model. Contr.,* **1** (2021), 145–156.
12. S. Zhu, J. Lu, L. Lin, Y. Liu, Minimum-time and minimum-triggering control for the observability of stochastic Boolean networks, *IEEE T. Automat. Contr.,* **67** (2022), 1558–1565. http://doi.org/10.1109/tac.2021.3069739
13. Y. Yu, M. Meng, J. Feng, G. Chen, Observability criteria for Boolean networks, *IEEE T.*

Automat. Contr., **67** (2022), 6248–6254. http://doi.org/10.1109/tac.2021.3131436

14. J. Zhong, B. Li, Y. Liu, W. Gui, Output feedback stabilizer design of Boolean networks based on network structure, *Frontiers of Information Technology & Electronic Engineering,* **21** (2020), 247–259. http://doi.org/10.1631/fitee.1900229

15. X. Yang, H. Li, On state feedback asymptotical stabilization of probabilistic Boolean control networks, *Syst. Control Lett.,* **160** (2022), 105107. http://doi.org/10.1016/j.sysconle.2021.105107

16. Z. Liu, J. Zhong, Y. Liu, W. Gui, Weak stabilization of Boolean networks under state-flipped control, *IEEE T. Neur. Net. Lear. Syst.,* 2021. http://doi.org/10.1109/tnnls.2021.3106918

17. J. Liu, L. Li, H. Fardoun, Complete synchronization of coupled Boolean networks with arbitrary finite delays, *Frontiers of Information Technology & Electronic Engineering,* **21** (2020), 281–293. http://doi.org/10.1631/fitee.1900438

18. T. Sun, R. Wang, X. Zhao, P. Sun, Partial and global stabilization at an attractor for k-valued logical control networks, *Journal of the Franklin Institute-Engineering and Applied Mathematics,* **357** (2020), 7003–7019. http://doi.org/10.1016/j.jfranklin.2020.04.054

19. A. Yerudkar, C. Del Vecchio, L. Glielmo, Sampled-data set stabilization of switched Boolean control networks, *IFAC-PapersOnLine,* **53** (2020), 6139–6144. http://doi.org/10.1016/j.ifacol.2020.12.1694

20. A. Yerudkar, C. Del Vecchio, L. Glielmo, Control of switched Boolean control networks by state feedback, *18th European Control Conference (ECC),* (2019), 1999–2004. http://doi.org/10.23919/ecc.2019.8796149

21. T. Sun, X. Sun, New results for hybrid delay systems with initial data: Razumikhin-type stability criteria, *IEEE Transactions on Circuits and Systems II: Express Briefs,* **69** (2022), 3194–3198. http://doi.org/10.1109/tcsii.2021.3136225

22. T. Sun, X. Zhao, X. Sun, Switched dynamic systems with logic switching and its stability analysis, *SIAM J. Control Optim.,* **59** (2021), 1188–1217. http://doi.org/10.1137/20m1337958

23. Y. Guo, Y. Wu, W. Gui, Stability of discrete-time systems under restricted switching via logic dynamical generator and STP-based mergence of hybrid states, *IEEE T. Automat. Contr.,* **67** (2022), 3472–3483. http://doi.org/10.1109/tac.2021.3105319

24. C. Sun, H. Li, Algebraic formulation and application of multi-input single-output hierarchical fuzzy systems with correction factors, *IEEE Transactions on Fuzzy Systems,* 2022.

25. H. Lyu, W. Wang, X. Liu, Universal approximation of fuzzy relation models by semitensor product, *IEEE T. Fuzzy Syst.,* **28** (2020), 2972–2981. http://doi.org/10.1109/TFUZZ.2019.2946512

26. C. Sun, H. Li, Parallel fuzzy relation matrix factorization towards algebraic formulation, universal approximation and interpretability of MIMO hierarchical fuzzy systems, *Fuzzy Sets and Systems,* **450** (2022), 68–86. http://doi.org/10.1016/j.fss.2022.07.008

27. H. Li, Y. Wang, L. Xie, Output tracking control of Boolean control networks via state feedback: constant reference signal case, *Automatica,* **59** (2015), 54–59. http://doi.org/10.1016/j.automatica.2015.06.004

28. Z. Man, M.Palaniswami, Robust tracking control for rigid robotic manipulators, *IEEE T. Automat. Contr.,* **39** (1994), 154–159. http://doi.org/10.1109/9.273355

29. K. Melhem, W. Wang, Global output tracking control of flexible joint robots via factorization of the manipulator mass matrix, *IEEE T. Robot.,* **25** (2009), 428–437. http://doi.org/10.1109/tro.2009.2012016

30. S. Al-Hiddabi, N. McClamroch, Tracking and maneuver regulation control for nonlinear nonminimum phase systems: application to flight control, *IEEE T. Contr. Syst. Tech.,* **10** (2002), 780–792. http://doi.org/10.1109/tcst.2002.804120

31. Y. Chen, P. Sun, T. Sun, M. Alassafi, A. Ahmad, Optimal output tracking of switched Boolean networks, *Asian J. Control,* **24** (2021), 1235–1246. http://doi.org/10.1002/asjc.2509

32. X. Zhang, Y. Wang, D. Cheng, Output tracking of Boolean control networks, *IEEE T. Automat. Contr.,* **65** (2020), 2730–2735. http://doi.org/10.1109/tac.2019.2944903

33. Z. Zhang, T. Leifeld, P. Zhang, Finite horizon tracking control of Boolean control networks, *IEEE T. Automat. Contr.,* **63** (2018), 1798–1805. http://doi.org/10.1109/acc.2016.7526770

34. A. Yerudkar, C. Del Vecchio, L. Glielmo, Output tracking control of probabilistic Boolean control networks, *IEEE International Conference on Systems, Man and Cybernetics,* 2019. https://ieeexplore.ieee.org/document/8914068

35. A. Yerudkar, C. Del Vecchio, L. Glielmo, Output tracking control design of switched Boolean control networks, *IEEE Control Systems Letters,* **4** (2019), 355–360. http://doi.org/10.1109/lcsys.2019.2928474

36. T. Cimen, S. Banks, Nonlinear optimal tracking control with application to supertankers for autopilot design, *Automatica,* **40** (2004), 1845–1863. https://doi.org/10.1016/j.automatica.2004.05.015

37. F. Schreiber, M. Valcher, Formal assessment of some properties of context-aware systems, *International Journal of Next-Generation Computing,* **10** (2019).

38. H. Chen, X. Li, J. Sun, Stabilization, controllability and optimal control of Boolean networks with impulsive effects and state constraints, *IEEE T. Automat. Contr.,* **60** (2015), 806–811. http://doi.org/10.1109/tac.2014.2330432

39. T. Kitson, Stabilization, The effect of cephalosporin antibiotics on alcohol metabolism: a review, *Alcohol,* **4** (1987), 143–148. http://doi.org/10.1016/0741-8329(87)90035-8

40. X. Jin, S. Tang, Q. Chen, J. Zou, T. Zhang, F. Liu, et al., Furazolidone induced oxidative DNA damage via up-regulating ROS that caused cell cycle arrest in human hepatoma G2 cells, *Toxicol. Lett.,* **201** (2011), 205–212. http://doi.org/10.1016/j.toxlet.2010.12.021

41. H. Li, L. Xie, Y. Wang, Output regulation of Boolean control networks, *IEEE T. Automat. Contr.,* **62** (2017), 2993–2998. http://doi.org/10.1109/tac.2016.2606600

AIMS Press