Self-Triggered Set Stabilization of Boolean Control Networks and Its Applications

Rong Zhao, Jun-e Feng, and Dawei Zhang

***Abstract*—Set stabilization is one of the essential problems in engineering systems, and self-triggered control (STC) can save the storage space for interactive information, and can be successfully applied in networked control systems with limited communication resources. In this study, the set stabilization problem and STC design of Boolean control networks are investigated via the semi-tensor product technique. On the one hand, the largest control invariant subset is calculated in terms of the strongly connected components of the state transition graph, by which a graph-theoretical condition for set stabilization is derived. On the other hand, a characteristic function is exploited to determine the triggering mechanism and feasible controls. Based on this, the minimum-time and minimum-triggering open-loop, state-feedback and output-feedback STCs for set stabilization are designed, respectively. As classic applications of self-triggered set stabilization, self-triggered synchronization, self-triggered output tracking and self-triggered output regulation are discussed as well. Additionally, several practical examples are given to illustrate the effectiveness of theoretical results.**

***Index Terms*—Boolean control networks (BCNs), output regulation, self-triggered control, semi-tensor product of matrices, set stabilization, synchronization.**

I. Introduction

BOOLEAN networks (BNs), introduced by Kauffman in 1969 to model gene regulatory networks [1], have aroused significant research interest. This booming filed of research is mainly attributed to two aspects: One is the significant application prospects in systems biology including cell differentiation, immune response, biological evolution, neural networks and gene regulation [2], and the other is the emergence of a powerful tool, called the semi-tensor product (STP) [3]. By resorting to the STP, an algebraic state-space representation is established for BNs and Boolean control networks (BCNs), which has greatly contributed to the development of BNs and BCNs. Moreover, the framework of BCNs has been extended to various engineering applications and dynamical systems, such as discrete-event systems [4], networked evolutionary games [5], encryption and face recognition [6], and so on.

Stability and stabilization are fundamental issues in control systems, which determine whether a system converges to or can be stabilized to a single point. Cheng *et al.* [7] took the lead in studying the stability and stabilization of BNs and BCNs and presented some algebraic criteria. Subsequently, the designs of state-feedback stabilizers [8] and output-feedback stabilizers [9] were investigated. As promotions of stability and stabilization, set stability refers to whether a BN can converge to a given subset, and set stabilization focuses on whether there exists a controller to stabilize a BCN to a given subset [10]. Correspondingly, different control strategies have been adopted for set stabilization, such as state-feedback control [11] and output-feedback control [12], [13], sampled-data control [14], event-triggered control [15] and pinning control [16]. The typical problems of set stability/stabilization, synchronization [17], output tracking [18] and output regulation [19] have important applications in practical engineering, such as the therapeutic intervention of diseases, attitude tracking of aircraft, and operational control of robots, etc.

Traditional control laws, requiring continuous network communication and updating of the controller, inevitably lead to communication limitations, network congestion, resource waste and other problems [20], [21]. In order to overcome these issues and reduce the waste of unnecessary computation and communication resources, event-triggered control (ETC) [22] and self-triggered control (STC) [23] came into being. The difference between ETC and STC is that the former is passive, while the latter is active. As a matter of fact, in ETC, triggering conditions based on the current measurement are continuously monitored. Once the triggering conditions are satisfied, an event is triggered and the control task is then executed [24], [25]. In STCs, the next update time is pre-calculated at the current triggering moment based on previously received data and knowledge about the system dynamics [26]. Compared with ETCs, STCs can avoid continuous monitoring of the system state by event triggers, and thus can significantly reduce the computational resources.

In the study of BCNs, ETCs have attracted much attention. For instance, the event-triggered stabilization of BCNs [27] and Markovian jump BCNs [28], and event-triggered set stabilization of switched BNs [15] were comprehensively discussed. The optimal ETC strategy for stabilization of BCNs was obtained in [29] as well. In addition to stability problem,

Manuscript received September 2, 2023; revised October 10, 2023; accepted October 14, 2023. This work was supported by the National Natural Science Foundation of China (62273201, 62173209, 72134004, 62303170) and the Research Fund for the Taishan Scholar Project of Shandong Province of China (TSTP20221103). Recommended by Associate Editor Xiaohua Ge. *(Corresponding author: Jun-e Feng.)*

Citation: R. Zhao, J.-e Feng, and D. Zhang, "Self-triggered set stabilization of Boolean control networks and its applications," *IEEE/CAA J. Autom. Sinica*, vol. 11, no. 7, pp. 1631–1642, Jul. 2024.

The authors are with the School of Mathematics, Shandong University, Jinan 250100, China (e-mail: zhaorong@mail.sdu.edu.cn; fengjune@sdu.edu.cn; dwzhang@sdu.edu.cn).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/JAS.2023.124050

ETCs were also applied to the research of observability of Markovian jump BNs [30] and robust invariance of probabilistic BCNs [31]. With regard to STCs for BCNs, there are only a few related works up to now. Acernese *et al.* [32] designed the state-feedback self-triggered scheme for the model-free stabilization problem by the Q-learning technique. Meng *et al.* [33] combined Lyapunov function theory to derive open-loop self-triggering conditions for stabilization depending on known state-feedback stabilizers. Bajaria *et al.* [34] first developed a Q-learning random forest algorithm to design state-feedback controllers for stabilization, and then followed the idea of [33] to devise STC strategies. It should be pointed out that the methods of [32], [34] are model-free, and the method of [33] is model-based. But the core idea of [34] is consistent with [33], where state-feedback controllers need to be designed in advance.

From the aforementioned overview, it can be seen that the self-triggered set stabilization of BCNs still have a great deal of research space worth exploring. In the most of the existing works on the set stabilization problem [7], [10], [13], [35], the obtained discriminate conditions are usually algebraic criteria, which have high computational complexity and are not intuitive enough. Moreover, in the research of STCs of BCNs [32]–[34], only stabilization problem is considered. However, there is no study of STCs for set stabilization, and the self-triggered scheduling in [33], [34] depends on pre-designed state feedback controllers. To cope with the above problems, we first give intuitive graphical criteria for set stabilization by resorting to the strongly connected components of the state transition graph in Section IV, and then develop a general method to design the self-triggered set stabilizers in Section V. The main contributions of this paper are highlighted as follows.

1) By resorting to Tarjan's algorithm, graphical criteria for the largest control invariant subset and set stabilization are provided. Compared with the existing results, our results reduce the computational complexity significantly (see Remark 2).

2) A characteristic function is employed to determine the triggering mechanism and feasible controls. Correspondingly, the minimum-time and minimum-triggering open-loop STCs (OLSTCs), state-feedback STCs (SFSTCs) and output-feedback STCs (OFSTCs) for set stabilization are designed. Compared with [33], [34], our STC schemes do not require a state-feedback controller to be designed in advance. Besides, only OLSTCs and SFSTCs are studied in [32]–[34] (see Remark 4).

3) The results of self-triggered set stabilization is also applied to self-triggered synchronization, self-triggered output tracking and self-triggered output regulation. To the best of our knowledge, there is no literature on the design of STCs for synchronization, output tracking and output regulation of BCNs up to now.

The remainder of this paper is organized as follows. Section II includes some preliminaries. Section III presents the problem formulation. Section IV characterizes the set stabilization problem from a new standpoint. Section V discusses the design of minimum-time and minimum-triggering STCs for set stabilization. Section VI contains the applications of self-triggered set stabilization. Section VII provides a brief conclusion.

II. Preliminaries

In this section, the STP of matrices and the algebraic state-space representation of BCNs are reviewed. Some notations used in the sequel are listed below:

1) $\mathbb{N}$: The set of non-negative integers.
2) $\mathbb{R}^{m\times n}$: The set of $m\times n$ real matrices.
3) $[m,n]=\{m,m+1,\dots,n\}$, $m\le n\in\mathbb{N}$.
4) $|\cdot|$: The cardinality of a set.
5) $\|\cdot\|_\infty$: The infinite norm of a vector.
6) $\mathrm{sgn}(\cdot)$: The symbolic function. Specifically, $\mathrm{sgn}(x)$ equals 1 for $x>0$, 0 for $x=0$, and -1 for $x<0$.
7) M^T: The transpose of matrix M.
8) $\mathrm{Col}_i(M)$: The i-th column of matrix M.
9) $[M]_{i,j}$: The (i,j)-element of matrix M.
10) $\Delta_m=\{\delta_m^i|i=1,2,\dots,m\}$, where $\delta_m^i=\mathrm{Col}_i(I_m)$, and I_m is the $m\times m$ identity matrix.
11) $\mathbf{1}_n=[\underbrace{1,1,\dots,1}_n]^T$, $\mathbf{0}_n=[\underbrace{0,0,\dots,0}_n]^T$, $n\in\mathbb{N}$.
12) $\mathbb{B}=\{0,1\}$; $\mathbb{B}^{m\times n}$: The set of $m\times n$ Boolean matrices; $\mathbb{B}^n=\mathbb{B}^{n\times 1}$.
13) $\delta_m[i_1\ i_2\ \cdots\ i_n]$: A logical matrix $[\delta_m^{i_1}\ \delta_m^{i_2}\ \cdots\ \delta_m^{i_n}]$.
14) $\mathcal{L}_{m\times n}$: The set of $m\times n$ logical matrices.
15) $\otimes$: The Kronecker product of matrices; $*$: the Khatri-Rao product of matrices; $\odot$: The Hadamard product of matrices.

Then, the STP is introduced. Further properties about STP can be found in [3].

Definition 1 [3]: Let $M\in\mathbb{R}^{a\times b}$ and $N\in\mathbb{R}^{c\times d}$. The STP of M and N is defined as $M\ltimes N=(M\otimes I_{e/b})(N\otimes I_{e/c})$, where e is the least common multiple of b and c.

Note that when $b=c$, STP degrades into the traditional matrix product. Without confusion, the symbol $\ltimes$ is omitted throughout this paper.

The following lemma is fundamental to convert logical functions into algebraic forms equivalently.

Lemma 1 [3]:

1) Given a logical function $f:\mathbb{B}^n\to\mathbb{B}$, there exists a unique matrix $F_f\in\mathcal{L}_{2\times 2^n}$ such that

$$\delta_2^{2-f(\mathbf{x}_1,\mathbf{x}_2,\dots,\mathbf{x}_n)}=F_f\ltimes_{i=1}^n x_i$$

where $\mathbf{x}_i\in\mathbb{B}$, $x_i=\delta_2^{2-\mathbf{x}_i}\in\Delta_2$ and F_f is called the structure matrix of f.

2) Assume $y=F_y\ltimes_{i=1}^n x_i$, $z=F_z\ltimes_{i=1}^n x_i$, where $x_i\in\Delta_2$, $i=1,2,\dots,n$, $F_y,F_z\in\mathcal{L}_{2\times 2^n}$. Then,

$$yz=(F_y*F_z)\ltimes_{i=1}^n x_i$$

where $M_y*F_z=[Col_1(F_y)\otimes Col_1(F_z),\dots,Col_{2^n}(F_y)\otimes Col_{2^n}(F_z)]$.

III. Problem Formulation

Consider a BCN with n state nodes, m input nodes and p output nodes

$$\begin{cases}\mathbf{x}_1(t+1) = f_1(\mathbf{u}_1(t),\ldots,\mathbf{u}_m(t),\mathbf{x}_1(t),\ldots,\mathbf{x}_n(t))\\ \mathbf{x}_2(t+1) = f_2(\mathbf{u}_1(t),\ldots,\mathbf{u}_m(t),\mathbf{x}_1(t),\ldots,\mathbf{x}_n(t))\\ \quad\vdots\\ \mathbf{x}_n(t+1) = f_n(\mathbf{u}_1(t),\ldots,\mathbf{u}_m(t),\mathbf{x}_1(t),\ldots,\mathbf{x}_n(t))\\ \mathbf{y}_j(t) = g_j(\mathbf{x}_1(t),\ldots,\mathbf{x}_n(t)),\ \ j=1,2,\ldots,p\end{cases} \tag{1}$$

where $\mathbf{x}_i(t)\in\mathbb{B}$, $\mathbf{u}_k(t)\in\mathbb{B}$ and $\mathbf{y}_j(t)\in\mathbb{B}$ represent the state, input and output of BCN (1), respectively, and $f_i:\mathbb{B}^{m+n}\to\mathbb{B}$, $g_j:\mathbb{B}^n\to\mathbb{B}$ are logical functions. Based on Lemma 1, the equivalent algebraic state-space representation of BCN (1) is

$$\begin{cases}x(t+1) = Lu(t)x(t)\\ y(t) = Hx(t)\end{cases} \tag{2}$$

where $x(t)=\ltimes_{i=1}^n x_i(t)\in\Delta_{2^n}$, $u(t)=\ltimes_{k=1}^m u_k(t)\in\Delta_{2^m}$ and $y(t)=\ltimes_{j=1}^p y_j(t)\in\Delta_{2^p}$. Moreover, $L=F_{f_1}*F_{f_2}*\cdots*F_{f_n}\in\mathcal{L}_{2^n\times 2^{m+n}}$ and $H=H_{g_1}*H_{g_2}*\cdots*H_{g_p}\in\mathcal{L}_{2^p\times 2^n}$, where $F_{f_i}\in\mathcal{L}_{2\times 2^{m+n}}$ and $H_{g_j}\in\mathcal{L}_{2\times 2^n}$ are the structure matrices of f_i and g_j, respectively.

For an initial state $x(0)=x_0\in\Delta_{2^n}$ and an input sequence $U:=\{u(t)\}_{t=0}^{\infty}$, the state trajectory of BCN (2) is denoted by $x(t;x_0,U)$, $t\in\mathbb{N}$. The set stabilization problem is introduced as follows.

Definition 2 [10]: Given a target set $\mathcal{M}\subseteq\Delta_{2^n}$, BCN (2) is said to be $\mathcal{M}$-stabilizable, if for any initial state x_0, there exist an input sequence U and an integer $T(x_0,U)$ such that

$$x(t;x_0,U)\in\mathcal{M},\ \forall t\geq T(x_0,U). \tag{3}$$

Denote by $T_m(x_0,U)$ the smallest integer satisfying (3). Then $T_m(x_0):=\min\limits_U T_m(x_0,U)$ is said to be the shortest transition period of x_0 and $T_m:=\max_{x_0\in\Delta_{2^n}}T_m(x_0)$ is said to be the shortest transition period of BCN (2).

In general, the self-triggering mechanism of a control system can be depicted by Fig. 1. Corresponding to BCN (2), an open-loop self-triggered controller (OLSTC) can be described as

$$\begin{cases}u(t)=u(t_k)\in\mathbf{U}\{x(t_k)\},\ t\in[t_k,t_{k+1}-1]\\ t_{k+1}=t_k+\pi(x(t_k))\end{cases} \tag{4}$$

where t_k, $k\in\mathbb{N}$ is the triggering moment with $t_0=0$, $\pi(x(t_k))$ is the interval between two execution moments, and $\mathbf{U}\{x(t_k)\}$ is the set of feasible controls at moment t_k. Further, the state-feedback self-triggered controller (SFSTC) and the output-feedback self-triggered controller (OFSTC) are in the form of

$$\begin{cases}u(t)=Kx(t_k),\ t\in[t_k,t_{k+1}-1]\\ t_{k+1}=t_k+\bar{\pi}_1(x(t_k))\end{cases} \tag{5}$$

$$\begin{cases}u(t)=Gy(t_k),\ t\in[t_k,t_{k+1}-1]\\ t_{k+1}=t_k+\bar{\pi}_2(y(t_k))\end{cases} \tag{6}$$

where $K\in\mathcal{L}_{2^m\times 2^n}$ and $G\in\mathcal{L}_{2^m\times 2^p}$ are the feedback gain matrices, and $\bar{\pi}_1(x(t_k))$, $\bar{\pi}_2(y(t_k))$ are the triggering intervals.

Remark 1: Different from the ordinary control law, (4)−(6) show that an STC is composed of two parts. One is the selection of feasible controls, and the other is the design of the triggering mechanism. In other words, the feasible controls and triggering mechanism should be devised simultaneously. Obviously, if $t_{k+1}-t_k=1$ holds for all $k\in\mathbb{N}$, then STCs (4)−(6) are trivial, i.e., (4)−(6) degrade into ordinary controllers.

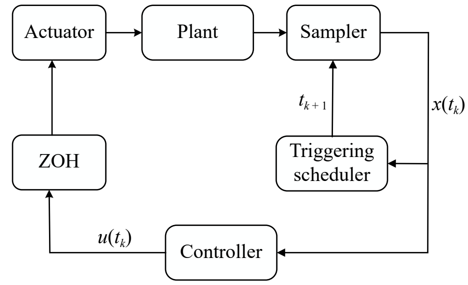

Fig. 1. Block diagram of the STC framework. ZOH represents the zero-order hold.

Definition 3: STCs (4)−(6) are said to be minimum-time and minimum-triggering $\mathcal{M}$-stabilizers, if under which, BCN (2) is $\mathcal{M}$-stabilizable with the shortest transition period and the number of triggering moments of STCs (4)−(6) is the least.

The objective of this paper is to characterize the $\mathcal{M}$-stabilization from a graph-theoretical perspective that is different from the algebraic criteria in [7], [10], [13], [35], and then design all possible minimum-time and minimum-triggering OLSTCs, SFSTCs and OFSTCs for $\mathcal{M}$-stabilization.

IV. Set Stabilization

In this section, set stabilization problem is discussed and new criteria are obtained. Given an initial state, the trajectory of a BCN will enter into a stable structure, which is usually called the control invariant subset.

Definition 4 [10]: A subset $\mathcal{M}_1\subseteq\mathcal{M}$ is said to be a control invariant subset (CIS) of $\mathcal{M}$, if for any $x_0\in\mathcal{M}_1$, there exists a control sequence U such that $x(t;x_0,U)\in\mathcal{M}_1$, $\forall t\in\mathbb{N}$. The union of all CISs of $\mathcal{M}$, denoted by $\mathcal{I}_{\mathcal{M}}$, is said to be the largest CIS (LCIS) of $\mathcal{M}$.

To interpret the set stabilization of BCNs with a new perspective, some knowledge of graph theory is introduced, where we can refer to [36]. A directed graph consists of a set of vertices $\mathcal{V}$ and a set of directed edges $\mathcal{E}\subseteq\mathcal{V}\times\mathcal{V}$. A sequence of directed edges $(v_{i_1},v_{i_2})\to(v_{i_2},v_{i_3})\to\cdots\to(v_{i_{l-1}},v_{i_l})$ is called a path from v_{i_1} to v_{i_l}. A subset of vertices is called strongly connected if each vertex has at least one path to all other vertices. A strongly connected component (SCC) is a maximal strongly connected subset of vertices. The SCC composed of only one single vertex v with $(v,v)\notin\mathcal{E}$ is called trivial, and is called non-trivial otherwise.

The state transition graph (STG) of BCN (2) is a directed graph, which is defined as $\mathcal{G}=(\mathcal{V}(\mathcal{G}),\mathcal{E}(\mathcal{G}))$ with $\mathcal{V}(\mathcal{G})=\Delta_{2^n}$ and $\mathcal{E}(\mathcal{G})=\{(\delta_{2^n}^i,\delta_{2^n}^j)|[\mathcal{A}]_{j,i}=1\}$, where $\mathcal{A}=\mathrm{sgn}(L\mathbf{1}_{2^m})$. Let $\mathcal{G}|_{\mathcal{M}}$ be the induced subgraph[1] of $\mathcal{G}$ with respect to $\mathcal{M}$. Denote by $\mathcal{S}_N^1,\mathcal{S}_N^2,\ldots,\mathcal{S}_N^{\alpha}$ all non-trivial SCCs of $\mathcal{G}|_{\mathcal{M}}$, and $\mathcal{S}_T^1,\mathcal{S}_T^2,\ldots,\mathcal{S}_T^{\beta}$ all trivial SCCs of $\mathcal{G}|_{\mathcal{M}}$, where $0\leq\alpha,\beta\leq 2^n$ and $\alpha+\beta\geq 1$. By means of Tarjan's algorithm[2], all SCCs of $\mathcal{G}|_{\mathcal{M}}$ can be obtained. Then, the following proposition is made to obtain the LCIS of $\mathcal{M}$.

[1] The induced subgraph $\mathcal{G}|_{\mathcal{M}}$ is the subgraph of $\mathcal{G}$, which has $\mathcal{M}$ as its set of vertices and contains all the edges of $\mathcal{G}$ that have both endpoints in $\mathcal{M}$.

[2] Tarjan's algorithm is a linear-time algorithm proposed by Tarjan [37] to solve the SCCs of a directed graph.

Proposition 1: $\mathcal{I}_{\mathcal{M}} \neq \emptyset$ if and only if $\mathcal{G}|_{\mathcal{M}}$ has at least one non-trivial SCC, i.e., $\alpha \geq 1$. Moreover, if $\alpha \geq 1$, then $\mathcal{I}_{\mathcal{M}}$ can be determined by

$$\mathcal{I}_{\mathcal{M}} = \begin{cases} \bigcup_{i=1}^{\alpha} \mathcal{S}_N^i = \mathcal{M}, & \beta = 0 \\ \bigcup_{i=1}^{\alpha} \bigcup_{j=1}^{\beta} \mathcal{S}_N^i \hat{\mathcal{S}}_T^j, & \beta \geq 1 \end{cases} \tag{7}$$

where $\hat{\mathcal{S}}_T^j = \mathcal{S}_T^j$ if there exists a path from $\mathcal{S}_T^j$ to $\bigcup_{i=1}^{\alpha} \mathcal{S}_N^i$, otherwise $\hat{\mathcal{S}}_T^j = \emptyset$.

Proof (Sufficiency): Assume $\alpha \geq 1$. For any $\delta_{2^n}^{i_1} \neq \delta_{2^n}^{i_2} \in \mathcal{S}_N^i$, $\delta_{2^n}^{i_1}$ and $\delta_{2^n}^{i_2}$ are strongly connected. That is, there exist two paths

$$(\delta_{2^n}^{i_1}, \delta_{2^n}^{j_1}) \to (\delta_{2^n}^{j_1}, \delta_{2^n}^{j_2}) \to \cdots \to (\delta_{2^n}^{j_{l-1}}, \delta_{2^n}^{j_l}) \to (\delta_{2^n}^{j_l}, \delta_{2^n}^{i_2})$$

$$(\delta_{2^n}^{i_2}, \delta_{2^n}^{j'_1}) \to (\delta_{2^n}^{j'_1}, \delta_{2^n}^{j'_2}) \to \cdots \to (\delta_{2^n}^{j'_{l'-1}}, \delta_{2^n}^{j'_{l'}}) \to (\delta_{2^n}^{j'_{l'}}, \delta_{2^n}^{i_1})$$

where $\delta_{2^n}^{j_t}, \delta_{2^n}^{j'_{t'}} \in \mathcal{S}_N^i$, $t \in [1, l]$, $t' \in [1, l']$. Note that $(\delta_{2^n}^{t_1}, \delta_{2^n}^{t_2}) \in \mathcal{E}(\mathcal{G}|_{\mathcal{M}})$ if and only if $[\mathcal{A}]_{t_2, t_1} = 1$, i.e., there is a control $u \in \Delta_{2^m}$ such that $Lu\delta_{2^n}^{t_1} = \delta_{2^n}^{t_2}$. Therefore, there exist two control sequences U_1, U_2 such that

$$\begin{cases} x(t; \delta_{2^n}^{i_1}, U_1) \in \mathcal{S}_N^i, \ \ t \in [1, l] \\ x(t'; \delta_{2^n}^{i_2}, U_2) \in \mathcal{S}_N^i, \ \ t' \in [1, l'] \\ x(l+1; \delta_{2^n}^{i_1}, U_1) = \delta_{2^n}^{i_2}, x(l'+1; \delta_{2^n}^{i_2}, U_2) = \delta_{2^n}^{i_1}. \end{cases} \tag{8}$$

Hence, it follows from Definition 4 that each $\mathcal{S}_N^i$ is a CIS, and $\bigcup_{i=1}^{\alpha} \mathcal{S}_N^i \subseteq \mathcal{I}_{\mathcal{M}} \neq \emptyset$. If $\beta = 0$, then, $\mathcal{I}_{\mathcal{M}} \subseteq \mathcal{M} = \bigcup_{i=1}^{\alpha} \mathcal{S}_N^i$, which yields $\mathcal{I}_{\mathcal{M}} = \bigcup_{i=1}^{\alpha} \mathcal{S}_N^i = \mathcal{M}$.

If $\beta \geq 1$ and for any $\delta_{2^n}^{\hat{j}} \in \bigcup_{j=1}^{\beta} \hat{\mathcal{S}}_T^j$, there exists a path from $\delta_{2^n}^{\hat{j}}$ to $\bigcup_{i=1}^{\alpha} \mathcal{S}_N^i$, then there exist $\mathcal{S}_N^i$, $\delta_{2^n}^{i_1} \in \mathcal{S}_N^i$ and a path

$$(\delta_{2^n}^{\hat{j}}, \delta_{2^n}^{\hat{j}_1}) \to (\delta_{2^n}^{\hat{j}_1}, \delta_{2^n}^{\hat{j}_2}) \to \cdots \to (\delta_{2^n}^{\hat{j}_{\hat{l}-1}}, \delta_{2^n}^{\hat{j}_{\hat{l}}}) \to (\delta_{2^n}^{\hat{j}_{\hat{l}}}, \delta_{2^n}^{i_1})$$

where $\delta_{2^n}^{\hat{j}_t} \in \bigcup_{j=1}^{\beta} \hat{\mathcal{S}}_T^j$, $t \in [1, \hat{l}]$. Hence, there exists a control sequence U_3 such that

$$\begin{cases} x(t; \delta_{2^n}^{\hat{j}}, U_3) \in \bigcup_{j=1}^{\beta} \hat{\mathcal{S}}_T^j, \ \ t \in [1, \hat{l}] \\ x(\hat{l}+1; \delta_{2^n}^{\hat{j}}, U_3) \in \mathcal{S}_N^i. \end{cases} \tag{9}$$

By Definition 4, (8) and (9) show that $\bigcup_{i=1}^{\alpha} \bigcup_{j=1}^{\beta} \mathcal{S}_N^i \hat{\mathcal{S}}_T^j$ is a CIS. Thus, $\bigcup_{i=1}^{\alpha} \bigcup_{j=1}^{\beta} \mathcal{S}_N^i \hat{\mathcal{S}}_T^j \subseteq \mathcal{I}_{\mathcal{M}}$. Next, we prove that any CIS of $\mathcal{M}$ is contained in $\bigcup_{i=1}^{\alpha} \bigcup_{j=1}^{\beta} \mathcal{S}_N^i \hat{\mathcal{S}}_T^j$. By contradiction, assume that $\mathcal{M}_1$ is a CIS of $\mathcal{M}$, but $\mathcal{M}_1 \nsubseteq \bigcup_{i=1}^{\alpha} \bigcup_{j=1}^{\beta} \mathcal{S}_N^i \hat{\mathcal{S}}_T^j$. Since $\mathcal{M} = \bigcup_{i=1}^{\alpha} \bigcup_{j=1}^{\beta} \mathcal{S}_N^i \mathcal{S}_T^j$, there must exist $\mathcal{S}_T^{j'} = \{\delta_{2^n}^{j'}\} \subseteq \mathcal{M}_1$ satisfying $\hat{\mathcal{S}}_T^{j'} = \emptyset$, i.e., there is no path from $\mathcal{S}_T^{j'}$ to $\bigcup_{i=1}^{\alpha} \mathcal{S}_N^i$. Thus, for any control sequence U, there exists an integer $k' \geq 1$ such that $x(k'; \delta_{2^n}^{j'}, U) \in \Delta_{2^n} \backslash \mathcal{M} \nsubseteq \mathcal{M}_1$. This contradicts the fact that $\mathcal{M}_1$ is a CIS, and therefore $\mathcal{I}_{\mathcal{M}} = \bigcup_{i=1}^{\alpha} \bigcup_{j=1}^{\beta} \mathcal{S}_N^i \hat{\mathcal{S}}_T^j$.

Proof (Necessity): If $\mathcal{G}|_{\mathcal{M}}$ does not have non-trivial SCCs, then $\mathcal{M} = \bigcup_{j=1}^{\beta} \mathcal{S}_T^j$. This means for any $x_0 \in \mathcal{M}$ and any control sequence U, there exists T_0 such that $x(T_0; x_0, U) \notin \mathcal{M}$, which contradicts with $\mathcal{I}_{\mathcal{M}} \neq \emptyset$. ■

Similarly, all SCCs of $\mathcal{G}$, denoted by $\mathcal{S}_1, \mathcal{S}_2, \ldots, \mathcal{S}_\gamma$, can be derived immediately using Tarjan's algorithm. Combining with Proposition 1, a new criterion for the set stabilization of BCNs is obtained.

Proposition 2: BCN (2) is $\mathcal{M}$-stabilizable if and only if $\mathcal{G}|_{\mathcal{M}}$ has at least one non-trivial SCC, and for each $\mathcal{S}_i$ of $\mathcal{G}$ satisfying $\mathcal{S}_i \bigcap \mathcal{I}_{\mathcal{M}} = \emptyset$, there exists a path from $\mathcal{S}_i$ to $\mathcal{I}_{\mathcal{M}}$, $i \in [1, \gamma]$.

Proof: By Proposition 1, the existence of the non-trivial SCCs of $\mathcal{G}|_{\mathcal{M}}$ is equivalent to $\mathcal{I}_{\mathcal{M}} \neq \emptyset$. According to [10], BCN (2) is $\mathcal{M}$-stabilizable if and only if it is $\mathcal{I}_{\mathcal{M}}$-stabilizable.

Sufficiency: Evidently, the SCCs of $\mathcal{G}$ satisfy $\bigcup_{i=1}^{\gamma} \mathcal{S}_i = \Delta_{2^n}$ and $\mathcal{S}_i \cap \mathcal{S}_j = \emptyset$, $\forall i \neq j \in [1, \gamma]$. Thus, for any $x_0 \in \Delta_{2^n}$, there exists a unique $\mathcal{S}_i$ such that $x_0 \in \mathcal{S}_i$. Two cases should be considered.

Case 1: $\mathcal{S}_i \bigcap \mathcal{I}_{\mathcal{M}} \neq \emptyset$. Assume $\hat{x} \in \mathcal{S}_i \bigcap \mathcal{I}_{\mathcal{M}}$. Then, we can find a control sequence U and an integer T_1 such that $x(T_1; x_0, U) = \hat{x} \in \mathcal{I}_{\mathcal{M}}$, since $x_0, \hat{x} \in \mathcal{S}_i$ are strongly connected.

Case 2: $\mathcal{S}_i \bigcap \mathcal{I}_{\mathcal{M}} = \emptyset$. In light of the conditions, there exists a path from $\mathcal{S}_i$ to $\mathcal{I}_{\mathcal{M}}$. Then there is also a path from x_0 to $\mathcal{I}_{\mathcal{M}}$ as $x_0 \in \mathcal{S}_i$.

Necessity: $\mathcal{I}_{\mathcal{M}} \neq \emptyset$ is obvious. Assume that for $\mathcal{S}_i \bigcap \mathcal{I}_{\mathcal{M}} = \emptyset$, there is no path from $\mathcal{S}_i$ to $\mathcal{I}_{\mathcal{M}}$. Then for any $x_0 \in \mathcal{S}_i$ and any control sequence U, either $x(t; x_0, U) \in \mathcal{S}_i$ or $x(t; x_0, U) \in Delta_{2^n} \backslash \mathcal{I}_{\mathcal{M}}$, $\forall t \in \mathbb{N}$. This is a contradiction. ■

Obviously, BCN (2) is controllable if and only if $\gamma = 1$, i.e., $\mathcal{G}$ has only one SCC. Then, the following corollary is apparent.

Corollary 1: If $\mathcal{G}$ has a unique SCC, then BCN (2) is $\mathcal{M}$-stabilizable if and only if $\mathcal{G}|_{\mathcal{M}}$ has at least one non-trivial SCC. In other words, a controllable BCN is $\mathcal{M}$-stabilizable if and only if $\mathcal{I}_{\mathcal{M}} \neq \emptyset$.

Remark 2: As is well-known, the complexity of Tarjan's algorithm is $O(|\mathcal{V}| + |\mathcal{E}|)$. To calculate $\mathcal{I}_{\mathcal{M}}$ and determine the $\mathcal{M}$-stabilizability, we need to get the SCCs of $\mathcal{G}|_{\mathcal{M}}$ and $\mathcal{G}$ by Tarjan's algorithm, respectively. Note that $|\mathcal{V}(\mathcal{G}|_{\mathcal{M}})| = |\mathcal{M}|$, $|\mathcal{E}(\mathcal{G}|_{\mathcal{M}})| \leq |\mathcal{M}|^2$ and $|\mathcal{V}(\mathcal{G})| = 2^n$, $|\mathcal{E}(\mathcal{G})| \leq 2^{2n}$. Thus, the time complexities of Propositions 1 and 2 are $O(|\mathcal{M}|^2)$ and $O(2^{2n})$, respectively. Besides, when $\mathcal{M} = \{x_e\}$, the set stabilization degrades into the stabilization. Compared with the existing works, our results reduce computational complexity, which is shown in Table I.

Example 1: Consider a biological example: the lambda switch. Laschov and Margaliot built the following BCN model for the lambda switch [38]:

$$\begin{cases} N(t+1) = \neg cI(t) \wedge (\neg cro(t)) \\ cI(t+1) = \neg cro(t) \wedge (cI(t) \vee cII(t)) \\ cII(t+1) = \neg cI(t) \wedge \mathbf{u}(t) \wedge (N(t) \vee cIII(t)) \\ cIII(t) = \neg cI(t) \wedge \mathbf{u}(t) \wedge N(t) \\ cro(t) = \neg cI(t) \wedge (\neg cII(t)) \\ \mathbf{y}(t) = cI(t) \wedge (\neg cro(t)) \end{cases} \tag{10}$$

where $N, cI, cII, cIII, cro \in \mathbb{B}$ represent the phage genes, and $\mathbf{u}, \mathbf{y} \in \mathbb{B}$ are the input and output. Based on Lemma 1, (10) can be converted into the algebraic form (2) with $x(t) \in \Delta_{32}$, $u(t) \in \Delta_2$ and $L = \delta_{32}[32$ 24 32 24 32 24 32 24 26 2 26 2 25 9 25 9 32 24 32 24 32 24 32 24 28 4 32 8 27 11 31 15 32 24 32 24 32 24 32 24 32 8 32 8 31 15 31 15 32 24 32 24

TABLE I
COMPLEXITY

Problem	Result	Time complexity	Space complexity
LCIS	Proposition 1	$O(\|\mathcal{M}\|^2)$	$O(\|\mathcal{M}\|^2)$
	Proposition 2 [10]	$O(\|\mathcal{M}\|2^{3n})$	$O(2^{2n})$
	Proposition 2 [13]	$O(\|\mathcal{M}\|^3)$	$O(\|\mathcal{M}\|2^n)$
$\mathcal{M}$-stabilization	Proposition 2	$O(2^{2n})$	$O(2^{2n})$
	Proposition 5 [10]	$O(2^{4n})$	$O(2^{3n})$
	Proposition 5 [35]	$O(2^{4n})$	$O(2^{2n})$
x_e-stabilization	Proposition 2	$O(2^{2n})$	$O(2^{2n})$
	Theorem 5.18 [7]	$O(2^{3n+m2^n})$	$O(2^{2n+m2^n})$
	Corollary 1 [35]	$O(2^{4n})$	$O(2^{2n})$

32 24 32 24 32 8 32 8 31 15 31 15] $\in \mathcal{L}_{32\times 64}$, $H = \delta_2$[2 1 2 1 2 1 2 1 2 1 2 1 2 1 2 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2] $\in \mathcal{L}_{2\times 32}$.

Let the target set be $\mathcal{M} = \{\delta_{32}^6, \delta_{32}^{12}, \delta_{32}^{15}, \delta_{32}^{24}, \delta_{32}^{25}, \delta_{32}^{28}, \delta_{32}^{29}, \delta_{32}^{31}, \delta_{32}^{32}\}$. By resorting to Tarjan's algorithm, we can determine that the induced subgraph $\mathcal{G}|_{\mathcal{M}}$ has 7 SCCs including 3 non-trivial SCCs $\mathcal{S}_N^1 = \{\delta_{32}^{15}, \delta_{32}^{25}, \delta_{32}^{32}\}$, $\mathcal{S}_N^2 = \{\delta_{32}^{24}\}$, $\mathcal{S}_N^3 = \{\delta_{32}^{31}\}$ and 4 trivial SCCs $\mathcal{S}_T^1 = \{\delta_{32}^6\}$, $\mathcal{S}_T^2 = \{\delta_{32}^{29}\}$, $\mathcal{S}_T^3 = \{\delta_{32}^{28}\}$, $\mathcal{S}_T^4 = \{\delta_{32}^{12}\}$, which are also shown in Fig. 2. By Proposition 1, we get that

$$\mathcal{I}_{\mathcal{M}} = \bigcup_{i=1}^{3}\bigcup_{j=1}^{2} \mathcal{S}_N^i \mathcal{S}_T^j = \{\delta_{32}^6, \delta_{32}^{15}, \delta_{32}^{24}, \delta_{32}^{25}, \delta_{32}^{29}, \delta_{32}^{31}, \delta_{32}^{32}\}.$$

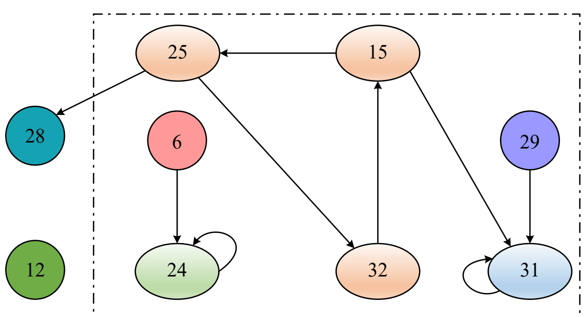

Fig. 2. The induced subgraph $\mathcal{G}|_{\mathcal{M}}$ in Example 1. Each color represents an SCC. The ovals represent the nontrivial SCCs and the circles represent the trivial SCCs. The states in the dotted box constitute $\mathcal{I}_{\mathcal{M}}$.

Furthermore, the STG $\mathcal{G}$ of BCN (10) is shown in Fig. 3, which is composed of 30 SCCs. Using the graph theory toolbox of Matlab, it is easy to verify that for $\mathcal{S}_i \bigcap \mathcal{I}_{\mathcal{M}} = \emptyset$, there exists a path from $\mathcal{S}_i$ to $\mathcal{I}_{\mathcal{M}}$. Then it follows from Theorem 2 that BCN (10) is $\mathcal{M}$-stabilizable.

V. Self-Triggered Control Design

In this section, the self-triggered strategies including open-loop and closed-loop cases are designed.

Note that if BCN (2) is not $\mathcal{M}$-stabilizable, then there is no STC to guarantee $\mathcal{M}$-stabilization. Besides, $\mathcal{M}$-stabilization can be easily verified by Propositions 1 and 2. Thus, the following assumption is reasonable.

Assumption 1: Throughout this section, BCN (2) is assumed to be $\mathcal{M}$-stabilizable.

A series of vectors induced by $\mathcal{I}_{\mathcal{M}}$ are iteratively constructed as follows:

$$\begin{cases} V_k(\mathcal{I}_{\mathcal{M}}) = \text{sgn}\left[\mathcal{A}^T V_{k-1}(\mathcal{I}_{\mathcal{M}})\right], & k \in [1, 2^n] \\ V_0(\mathcal{I}_{\mathcal{M}}) = \sum_{\delta_{2^n}^i \in \mathcal{I}_{\mathcal{M}}} \delta_{2^n}^i \in \mathbb{B}^{2^n} \end{cases} \tag{11}$$

where $\mathcal{A} = \text{sgn}(L\mathbf{1}_{2^m})$. Clearly, if BCN (2) is $\mathcal{M}$-stabilizable, then there exists a unique integer $\omega \in [1, 2^n]$ such that

$$V_0(\mathcal{I}_{\mathcal{M}}) < V_1(\mathcal{I}_{\mathcal{M}}) < \cdots < V_\omega(\mathcal{I}_{\mathcal{M}}) = \mathbf{1}_{2^n}. \tag{12}$$

In fact, $\omega = T_m$, i.e., ω is exactly the shortest transition period of BCN (2). Otherwise, if $\omega \neq T_m$, then there exists $x_0' \in \Delta_{2^n}$ such that $T_m(x_0') \geq \omega + 1$, which is a contradiction, since $V_\omega(\mathcal{I}_{\mathcal{M}}) = \mathbf{1}_{2^n}$ means that any state $x_0 \in \Delta_{2^n}$ can reach $\mathcal{I}_{\mathcal{M}}$ in ω steps.

To determine the minimum-time and minimum-triggering STCs, we define a characteristic function $\phi : \Delta_{2^n} \to [0, \omega]$ of BCN (2) by

$$\phi(x) = \left\| \sum_{j=0}^{\omega} jx \odot \left[V_j(\mathcal{I}_{\mathcal{M}}) - V_{j-1}(\mathcal{I}_{\mathcal{M}})\right] \right\|_\infty$$

where $V_{-1}(\mathcal{I}_{\mathcal{M}}) = \mathbf{0}_{2^n}$. The characteristic function ϕ is used to achieve the co-design of feasible controls and triggering mechanism. Based on the characteristic function, for a state $x \in \Delta_{2^n}$, two integers $\rho \geq 1$ and $\hat{\rho} \geq 0$, we denote

$$\mathbf{U}_\rho\{x\} = \{u | \phi((Lu)^t x) - \phi((Lu)^{t-1} x) < 0, t \in [1, \rho]\}$$

$$\hat{\mathbf{U}}_{\hat{\rho}}\{x\} = \{u | \phi((Lu)^t x) = 0, t \in [0, \hat{\rho}]\}.$$

Clearly, $\mathbf{U}_\rho\{x\}$ and $\hat{\mathbf{U}}_{\hat{\rho}}\{x\}$ contain all possible controls such that the value of the characteristic function is non-increasing along the trajectory of BCN (2) starting from initial state $x \in \Delta_{2^n}$ in time periods $[1, \rho]$ and $[0, \hat{\rho}]$, respectively. Moreover, we have $\mathbf{U}_\rho\{x\} \supseteq \mathbf{U}_{\rho+1}\{x\}$ and $\hat{\mathbf{U}}_{\hat{\rho}}\{x\} \supseteq \hat{\mathbf{U}}_{\hat{\rho}+1}\{x\}$. Then, let

$$\mu_1(x(t_k)) = \max\{\rho | \mathbf{U}_\rho\{x(t_k)\} \neq \emptyset, \rho \in [1, \omega]\}$$

$$\mu_2(x(t_k)) = \max\left\{\hat{\rho} | \hat{\mathbf{U}}_{\hat{\rho}}\{x(t_k + \mu_1(x(t_k)))\} \neq \emptyset, \hat{\rho} \in [0, |\mathcal{I}_{\mathcal{M}}|]\right\}$$

where $\max \emptyset = 0$. One sees that $\mu_1(x(t_k))$ is actually the maximum number of steps that keeps the value of the characteristic function ϕ strictly decreasing starting from $x(t_k)$. $\mu_2(x(t_k))$ is the maximum number of steps that makes the value of ϕ equal 0 starting from $x(t_k + \mu_1(x(t_k)))$. From the analysis above, we have the following lemma that is used to devise the minimum-time and minimum-triggering STCs.

Lemma 2: 1) $\mu_1(x(t_k)) = 0$ if and only if $x(t_k) \in \mathcal{I}_{\mathcal{M}}$.

2) $\mu_2(x(t_k)) > 0$ if and only if $x(t_k) \in \mathcal{I}_{\mathcal{M}}$ or $x(t_k + \mu_1(x(t_k))) \in \mathcal{I}_{\mathcal{M}}$.

Proof: In light of (12), for any $\delta_{2^n}^i \in \Delta_{2^n}$, there exists a unique $k(i) \in [0, \omega]$ such that $\delta_{2^n}^i \odot [V_{k(i)}(\mathcal{I}_{\mathcal{M}}) - V_{k(i)-1}(\mathcal{I}_{\mathcal{M}})] = \delta_{2^n}^i$. Thus, we have

$$\delta_{2^n}^i \odot [V_j(\mathcal{I}_{\mathcal{M}}) - V_{j-1}(\mathcal{I}_{\mathcal{M}})] = \begin{cases} \delta_{2^n}^i, & j = k(i) \\ \mathbf{0}_{2^n}, & j \neq k(i) \end{cases}$$

which indicates that

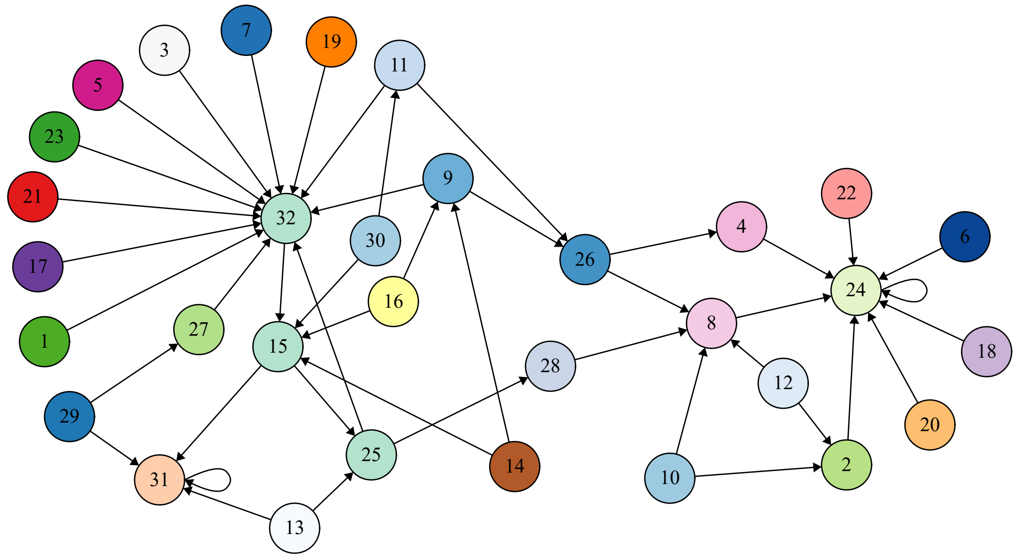

Fig. 3. The STG $\mathcal{G}$ of BCN (10) in Example 1 consisting of 30 SCCs. Each color represents an SCC.

$$\begin{aligned}\phi(\delta_{2^n}^i) &= \left\|\left|\sum_{j=0}^{\omega} j\delta_{2^n}^i \odot [V_j(\mathcal{I}_{\mathcal{M}}) - V_{j-1}(\mathcal{I}_{\mathcal{M}})]\right|\right\|_\infty \\ &= \left\|k(i)\delta_{2^n}^i \odot [V_j(\mathcal{I}_{\mathcal{M}}) - V_{j-1}(\mathcal{I}_{\mathcal{M}})]\right\|_\infty \\ &= \left\|k(i)\delta_{2^n}^i\right\|_\infty = k(i).\end{aligned}$$

1) Given a state $x \in \mathcal{I}_{\mathcal{M}}$, it holds that for any $u \in \Delta_{2^m}$,

$$\begin{cases}\phi(Lux) - \phi(x) > 0, \ Lux \notin \mathcal{I}_{\mathcal{M}} \\ \phi(Lux) - \phi(x) = 0, \ Lux \in \mathcal{I}_{\mathcal{M}}.\end{cases}$$

Therefore, if $x \in \mathcal{I}_{\mathcal{M}}$, then for any $\rho \geq 1$, $\mathbf{U}_\rho(x) = \emptyset$, which means $\mu_1(x) = 0$. If $x \notin \mathcal{I}_{\mathcal{M}}$, then we know from (12) that there exist $k \in [1, \omega]$ and $u \in \Delta_{2^m}$ such that

$$\begin{cases}x \odot [V_k(\mathcal{I}_{\mathcal{M}}) - V_{k-1}(\mathcal{I}_{\mathcal{M}})] = x \\ Lux \odot [V_{k-1}(\mathcal{I}_{\mathcal{M}}) - V_{k-2}(\mathcal{I}_{\mathcal{M}})] = Lux\end{cases}$$

which implies

$$\phi(Lux) - \phi(x) = k - 1 - k = -1 < 0.$$

Hence, for any $x \notin \mathcal{I}_{\mathcal{M}}$, we have $\mathbf{U}_1\{x\} \neq \emptyset$, so $\mu_1(x) \geq 1 > 0$. The argument above shows that $\mu_1(x(t_k)) = 0$ if and only if $x(t_k) \in \mathcal{I}_{\mathcal{M}}$.

2) Apparently, $\hat{\mathbf{U}}_1\{x\} \neq \emptyset$ if and only if $x \in \mathcal{I}_{\mathcal{M}}$. If $x(t_k) \in \mathcal{I}_{\mathcal{M}}$, then $x(t_k + \mu_1(x(t_k))) = x(t_k) \in \mathcal{I}_{\mathcal{M}}$, which means $\hat{\mathbf{U}}_1\{x(t_k + \mu_1(x(t_k)))\} = \hat{\mathbf{U}}_1\{x(t_k)\} \neq \emptyset$, and therefore $\mu_2(x(t_k)) \geq 1 > 0$. If $x(t_k) \notin \mathcal{I}_{\mathcal{M}}$, then $\mu_1(x(t_k)) > 0$, which yields that $\mu_2(x(t_k)) > 0$ when $x(t_k + \mu_1(x(t_k))) \in \mathcal{I}_{\mathcal{M}}$ and $\mu_2(x(t_k)) = 0$ otherwise. ■

For simplicity, we denote $\tilde{\mathbf{U}}\{x(t_k)\} := \mathbf{U}_{\mu_1(x(t_k))}\{x(t_k)\} \cap \hat{\mathbf{U}}_{\mu_2(x(t_k))}\{x(t_k + \mu_1(x(t_k)))\}$. Together with Lemma 2, the following theorem gives the criteria for the design of OLSTCs.

Theorem 1: OLSTC (4) is a minimum-time and minimum-triggering $\mathcal{M}$-stabilizer, if and only if $\pi(x(t_k))$ and $\mathbf{U}\{x(t_k)\}$ are designed as follows.

1) If $\mu_1(x(t_k)) = 0$ nd $\mu_2(x(t_k)) > 0$, then

$$\begin{cases}\pi(x(t_k)) = \mu_2(x(t_k)) \\ \mathbf{U}\{x(t_k)\} = \hat{\mathbf{U}}_{\mu_2(x(t_k))}\{x(t_k)\}.\end{cases} \tag{13}$$

2) If $\mu_1(x(t_k)) > 0$, $\mu_2(x(t_k)) \geq 0$ and $\tilde{\mathbf{U}}\{x(t_k)\} = \emptyset$, then

$$\begin{cases}\pi(x(t_k)) = \mu_1(x(t_k)) \\ \mathbf{U}\{x(t_k)\} = \mathbf{U}_{\mu_1(x(t_k))}\{x(t_k)\}.\end{cases} \tag{14}$$

3) If $\mu_1(x(t_k)) > 0$, $\mu_2(x(t_k)) > 0$ and $\tilde{\mathbf{U}}\{x(t_k)\} \neq \emptyset$, then

$$\begin{cases}\pi(x(t_k)) = \mu_1(x(t_k)) + \mu_2(x(t_k)) \\ \mathbf{U}\{x(t_k)\} = \tilde{\mathbf{U}}\{x(t_k)\}.\end{cases} \tag{15}$$

Proof: Firstly, OLSTC (4) is a minimum-time $\mathcal{M}$-stabilizer if and only if under (4), the state trajectory of BCN (2) starting from any initial state x_0 can reach $\mathcal{I}_{\mathcal{M}}$ in the shortest path. Actually, from (11) and (12), ω is the shortest transition period. Secondly, OLSTC (4) is minimum-triggering if and only if the transition period is minimum and the triggering interval $\pi(x(t_k))$ is maximal for any $k \in \mathbb{N}$. In what follows, four cases are taken into account to determine the triggering interval.

Case 1: $\mu_1(x(t_k)) = 0$ and $\mu_2(x(t_k)) > 0$, which means $x(t_k) \in \mathcal{I}_{\mathcal{M}}$. It is evident that for any $u \in \hat{\mathbf{U}}_{\hat{\rho}}\{x(t_k)\}$,

$$\phi((Lu)^t x(t_k)) = 0, \ \forall t \in [0, \hat{\rho}].$$

Then, to guarantee the minimum time and minimum triggering, $\pi(x(t_k))$ should be determined by

$$\pi(x(t_k)) = \max\left\{\hat{\rho} | \hat{\mathbf{U}}_{\hat{\rho}}\{x(t_k)\} \neq \emptyset, \hat{\rho} \in [0, |\mathcal{I}_{\mathcal{M}}|]\right\} = \mu_2(x(t_k))$$

which indicates $\mathbf{U}\{x(t_k)\} = \hat{\mathbf{U}}_{\mu_2(x(t_k))}\{x(t_k)\}$. At this time, $x(t) \in \mathcal{I}_{\mathcal{M}}$ holds for all $t \in [t_k, t_{k+1}]$.

Case 2: $\mu_1(x(t_k)) > 0$, $\mu_2(x(t_k)) > 0$ and $\tilde{\mathbf{U}}\{x(t_k)\} = \emptyset$. In this case, $x(t_k) \notin \mathcal{I}_{\mathcal{M}}$, $x(t_k + \mu_1(x(t_k))) \in \mathcal{I}_{\mathcal{M}}$, and for any $u \in \mathbf{U}_{\mu_1(x(t_k))}\{x(t_k)\}$, any $\bar{t} \in [1, \mu_1(x(t_k))]$ and $\hat{t} \in [1, \mu_2(x(t_k))]$,

$$\phi\left((Lu)^{\bar{t}} x(t_k)\right) - \phi\left((Lu)^{\bar{t}-1} x(t_k)\right) < 0 \tag{16}$$

$$\phi(x(t_k + \mu_1(x(t_k)))) = 0 \tag{17}$$

$$\phi\left((Lu)^{\hat{t}} x(t_k + \mu_1(x(t_k)))\right) \neq 0. \tag{18}$$

Immediately, (16) ensures the minimum time. Further, (17) and (18) mean that the maximal interval $\pi(x(t_k))$ should be $\mu_1(x(t_k))$. That is $t_{k+1} = t_k + \mu_1(x(t_k))$ and $\mathbf{U}\{x(t_k)\} = \mathbf{U}_{\mu_1(x(t_k))} \times$

$\{x(t_k)\}$, which yields that $x(t_{k+1}) \in \mathcal{I}_\mathcal{M}$.

Case 3: $\mu_1(x(t_k)) > 0$, $\mu_2(x(t_k)) > 0$ and $\tilde{\mathbf{U}}\{x(t_k)\} \neq \emptyset$. In this case, $x(t_k) \notin \mathcal{I}_\mathcal{M}$, $x(t_k + \mu_1(x(t_k))) \in \mathcal{I}_\mathcal{M}$, and there exists $u \in \tilde{\mathbf{U}}\{x(t_k)\}$ such that for any $\bar{t} \in [1, \mu_1(x(t_k))]$ and $\hat{t} \in [0, \mu_2(x(t_k))]$,

$$\phi\left((Lu)^{\bar{t}} x(t_k)\right) - \phi\left((Lu)^{\bar{t}-1} x(t_k)\right) < 0 \tag{19}$$

$$\phi\left((Lu)^{\hat{t}} x(t_k + \mu_1(x(t_k)))\right) = 0. \tag{20}$$

Similarly, (19) makes sure the minimum time. Inequalities (19) and (20) indicate that the maximal interval $\pi(x(t_k))$ should be $\mu_1(x(t_k)) + \mu_2(x(t_k))$. Then, $t_{k+1} = t_k + \mu_1(x(t_k)) + \mu_2(x(t_k))$ and $\mathbf{U}\{x(t_k)\} = \tilde{\mathbf{U}}\{x(t_k)\}$, which implies that $x(t_{k+1}) \in \mathcal{I}_\mathcal{M}$.

Case 4: $\mu_1(x(t_k)) > 0$ and $\mu_2(x(t_k)) = 0$. In this case, $x(t_k) \notin \mathcal{I}_\mathcal{M}$ and $x(t_k + \mu_1(x(t_k))) \notin \mathcal{I}_\mathcal{M}$, which means $\hat{\mathbf{U}}_{\mu_2(x(t_k))}\{x(t_k + \mu_1(x(t_k)))\} = \emptyset$, and thus $\tilde{\mathbf{U}}\{x(t_k)\} = \emptyset$. For any $u \in \mathbf{U}_{\mu_1(x(t_k))} \times \{x(t_k)\}$, condition (16) still holds. Thus, $\pi(x(t_k))$ and $\mathbf{U}\{x(t_k)\}$ should also be determined by (14) to guarantee the minimum time and minimum triggering. Then, we have

$$\begin{cases} x(t_{k+1}) = x(t_k + \mu_1(x(t_k))) \notin \mathcal{I}_\mathcal{M} \\ \phi(x(t_k)) > \phi(x(t_k+1)) > \cdots > \phi(x(t_{k+1})) > 0. \end{cases}$$

Repeating the process above, there must exist $\hat{k}$ such that $\phi(x(t_{k+\hat{k}})) > 0$ and $\phi(x(t_{k+\hat{k}} + \mu_1(x(t_{k+\hat{k}})))) = 0$, i.e., $x(t_{k+\hat{k}}) \notin \mathcal{I}_\mathcal{M}$, $x(t_{k+\hat{k}} + \mu_1(x(t_{k+\hat{k}}))) \in \mathcal{I}_\mathcal{M}$. So, this boils down to Case 2 or Case 3, both of which ultimately evolve into Case 1.

In conclusion, Cases 1–4 show that for any initial state $x(0) = x(t_0)$, under OLSTC (4) designed by (13)–(15), the state trajectory of BCN (2) will enter into $\mathcal{I}_\mathcal{M}$ and then stay at $\mathcal{I}_\mathcal{M}$ in minimum time and minimum triggering. ■

From Theorem 1, we have the following result for SFSTCs naturally.

Corollary 2: SFSTC (5) is a minimum-time and minimum-triggering $\mathcal{M}$-stabilizer, if and only if K and $\bar{\pi}_1(x(t_k))$ are designed by

$$\begin{cases} \bar{\pi}_1(x(t_k)) = \pi(x(t_k)) \\ \mathrm{Col}_j(K) \in \mathbf{U}\{\delta_{2^n}^i\},\ i \in [1, 2^n] \end{cases}$$

where $\pi(x(t_k))$ and $\mathbf{U}\{\delta_{2^n}^i\}$ are designed in Theorem 1.

In terms of the output information, we further discuss the OFSTC $\mathcal{M}$-stabilizer based on Theorem 1 and Corollary 2.

Theorem 2: OFSTC (6) is a minimum-time and minimum-triggering $\mathcal{M}$-stabilizer, if and only if for any $j \in [1, 2^p]$, $\Upsilon_j = \{i | [\mathrm{Col}_j(H^T)]_i = 1\} \neq \emptyset$ implies $\bigcap_{i \in \Upsilon_j} \mathbf{U}\{\delta_{2^n}^i\} \neq \emptyset$. Moreover, if the above-mentioned condition holds, then G and $\bar{\pi}_2(y(t_k))$ should be designed by

$$\mathrm{Col}_j(G) \in \begin{cases} \bigcap_{i \in \Upsilon_j} \mathbf{U}\{\delta_{2^n}^i\}, & \Upsilon_j \neq \emptyset \\ \Delta_{2^m}, & \Upsilon_j = \emptyset \end{cases}$$

and

$$\bar{\pi}_2(y(t_k)) = \min\{\pi(x(t_k)) | x(t_k) \in \Delta_{2^n}, y^T(t_k) H x(t_k) = 1\}.$$

Proof: Since $u(t) = Gy(t_k) = GHx(t_k)$, we see that an OFSTC is a special SFSTC. Assume $G = \delta_{2^m}[\varrho_1, \varrho_2, \ldots, \varrho_{2^p}]$ and $H = \delta_{2^p}[h_1, h_2, \ldots, h_{2^n}]$. In view of $GH = \delta_{2^m}[\varrho_{h_1}, \varrho_{h_2}, \ldots, \varrho_{h_{2^n}}]$, there exists an output-feedback gain matrix $G \in \mathcal{L}_{2^m \times 2^p}$, if and only if there exists a state-feedback gain matrix $K = \delta_{2^m}[\kappa_1, \kappa_2, \ldots, \kappa_{2^n}]$ such that $h_i = h_{i'}$, $i \neq i'$ implies $\kappa_i = \kappa_{i'}$. Note that $\Upsilon_j \neq \emptyset$ means that $h_i = j$ holds for all $i \in \Upsilon_j$. Consequently, based on Theorem 1 and Corollary 2, OFSTC (6) is a minimum-time and minimum-triggering $\mathcal{M}$-stabilizer, if and only if $\Upsilon_j \neq \emptyset$ implies $\bigcap_{i \in \Upsilon_j} \mathbf{U}\{\delta_{2^n}^i\} \neq \emptyset$. Meanwhile, $\mathrm{Col}_j(G)$ corresponding to $\Upsilon_j \neq \emptyset$ takes value from $\bigcap_{i \in \Upsilon_j} \mathbf{U}\{\delta_{2^n}^i\}$, and $\mathrm{Col}_j(G)$ corresponding to $\Upsilon_j = \emptyset$ can take any value in Δ_{2^m}. Clearly, $y^T(t_k) H x(t_k) = 1$ if and only if the corresponding output of $x(t_k)$ is $y(t_k)$. Thus, to guarantee the minimum time and minimum triggering, the triggering interval $\bar{\pi}_2(y(t_k))$ should be determined as the minimum value of $\pi(x(t_k))$ satisfying $y^T(t_k) H x(t_k) = 1$. ■

In the above, we have designed the time-optimal and triggering-optimal OLSTCs, SFSTCs and OFSTCs for set stabilization. Another interesting problem is the number of triggering moments for the STCs, which is discussed as follows.

Proposition 3: The number of triggering moments of $\mathcal{M}$-stabilizers (4)–(6) can be finite if and only if for the non-trivial SCC $\mathcal{S}_N^i$, $i \in [1, \alpha]$ satisfying

$$Lux \notin \mathcal{I}_\mathcal{M},\ \forall u \in \Delta_{2^m} \backslash \hat{\mathbf{U}}_1\{x\},\ \ \forall x \in \mathcal{S}_N^i \tag{21}$$

the following condition holds:

$$\hat{\mathbf{U}}_{|\mathcal{S}_N^i|}\{x\} \neq \emptyset,\ \forall x \in \mathcal{S}_N^i. \tag{22}$$

Proof: Since BCN (2) is $\mathcal{M}$-stabilizable, for any initial state $x(0)$, there exists an $\mathcal{S}_N^i$ such that $x(0)$ ultimately enters into $\mathcal{S}_N^i$ in a finite time and stay in $\mathcal{S}_N^i$ after that. The states in $\mathcal{S}_N^i$ satisfying (21) either remain in $\mathcal{S}_N^i$ or evolve into $\Delta_{2^n} \backslash \mathcal{I}_\mathcal{M}$ in the next step. Therefore, guaranteeing a finite number of triggering moments is equivalent to guaranteeing all states in $\mathcal{S}_N^i = \{x^1, x^2, \ldots, x^{|\mathcal{S}_N^i|}\}$ satisfying (21) have a common control u such that

$$Lux^i = x^{i+1},\ \ i = 1, 2, \ldots, |\mathcal{S}_N^i| - 1 \tag{23}$$

$$Lux^{|\mathcal{S}_N^i|} = Lux^1. \tag{24}$$

It is not hard to find that (23) and (24) are equivalent to

$$\phi((Lu)^t x) = 0,\ \forall t \in [0, |\mathcal{S}_N^i|],\ \ \forall x \in \mathcal{S}_N^i. \tag{25}$$

Evidently, (25) holds if and only if (22) holds, which completes the proof. ■

Remark 3: Proposition 3 emphasizes that the number of triggering moments of the self-triggered set stabilizers is not necessarily finite. This is different from the self-triggered stabilization, which must have a finite number of triggering moments is finite. It can be regarded as a special feature of self-triggered set stabilization distinguishing from self-triggered stabilization.

Remark 4: Comparisons with existing STC methods for BCNs:

1) In [32]–[34], the self-triggered strategy is only designed for stabilization problem, which is not suitable for set stabilization problem. In this paper, the proposed self-triggered strategies for set stabilization can also be used to stabilization problem, which indicates that the research content of this paper is more general and applicable than [32]–[34].

2) A precondition for designing STCs in [33] is that a state-feedback controller $u(t) = Kx(t)$ is utilized as a prior. This

means that to design an STC using the method proposed in [33], one needs to design a state-feedback controller in advance, which is redundant. Although [34] adopts a model-free pattern, the main idea of [34] follows that of [33]. In comparison, our approach avoids this deficiency.

3) The STCs designed in this paper are minimum-time and minimum-triggering, which cannot be solved by the methods in [32]–[34]. Moreover, our approach provides an unified framework to design OLSTCs, SFSTCs and OFSTCs successively. But in [32]–[34], only OLSTCs or SFSTCs are considered.

Remark 5 (Computational complexity analysis of the STC design): To design STCs, we first need to iteratively calculate a series of vectors by (11), the time and space complexities of which are $O(\omega 2^{2n})$ and $O(2^{2n})$, respectively. And then, for all $x(t_k)\in\Delta_{2^n}$, we need to calculate $\mathbf{U}\{x(t_k)\}$ and $\pi(x(t_k))$ by (13)−(15), which takes time and space complexities $O((\omega^2+|\mathcal{I}_{\mathcal{M}}|^2)2^{3n+m})$ and $O(2^{2n+m})$, respectively. Owing to $\omega\in[1,2^n]$ and $\omega+|\mathcal{I}_{\mathcal{M}}|\le 2^n$, the total time complexity for designing all possible minimum-time and minimum-triggering STCs is $O(2^{5n+m})$, and the space complexity is $O(2^{2n+m})$.

Example 2: Consider a programmable logic circuit shown in Fig. 4. A programmable logic device refers to any type of integrated circuit used for implementing hardware, where the chip can be configured by the end user to realize different designs [39]. Compared to a fixed logic device, programmable logic device has shorter development cycle, lower development cost, higher scalability, and more flexibility for subsequent functional upgrades, and thus it is preferred and widely used by industries such as electronics and communication. From Fig. 4, one can model this circuit as a BCN in the form of (2) with structure matrices $L=\delta_8$[8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 1, 1, 1, 5, 3, 3, 3, 7, 1, 1, 1, 5, 3, 3, 3, 7, 3, 3, 3, 7, 4, 4, 4, 8, 4, 4, 4, 8, 4, 4, 4, 8] and $H=\delta_2[1,2,2,2,1,2,2,2]$.

Suppose that the goal is to make the output of this circuit constant at 1 after a certain time. Accordingly, taking $\mathcal{M}=\{\delta_8^1,\delta_8^5\}$, one can verify that the circuit is $\mathcal{M}$-stabilizable. Then, we aim to design the self-triggered set stabilizers. Through routine calculations, the minimum-time and minimum-triggering OLSTCs is of the form: $u(t_0)\in\{\delta_8^5,\delta_8^6\}$, $\pi(x(t_0))=\infty$, $\forall x(t_0)\in\Delta_8$. Immediately, there are a total of 2^8 and 2^2 minimum-time and minimum-triggering SFSTCs and OFSTCs, respectively

$$u(t)=Kx(t_0),\ t\in[t_0,\infty) \tag{26}$$

$$u(t)=Gy(t_0),\ t\in[t_0,\infty) \tag{27}$$

with $\mathrm{Col}_i(K),\mathrm{Col}_j(G)\in\{\delta_8^5,\delta_8^6\}$, $\forall i\in[1,8]$, $j\in[1,2]$.

Using the method of [40], one can design the following state feedback controller:

$$u(t)=\delta_8[5,7,5,7,5,6,5,6]x(t) \tag{28}$$

which can stabilize the system to set $\mathcal{M}=\{\delta_8^1,\delta_8^5\}$. The state trajectories under STC (26) and state feedback controller (28) are shown in Fig. 5. Clearly, the initial state δ_8^2 can be stabilized to set $\mathcal{M}=\{\delta_8^1,\delta_8^5\}$ in 2 steps under STC (26), but it takes 3 steps under state feedback controller (28), which indicates

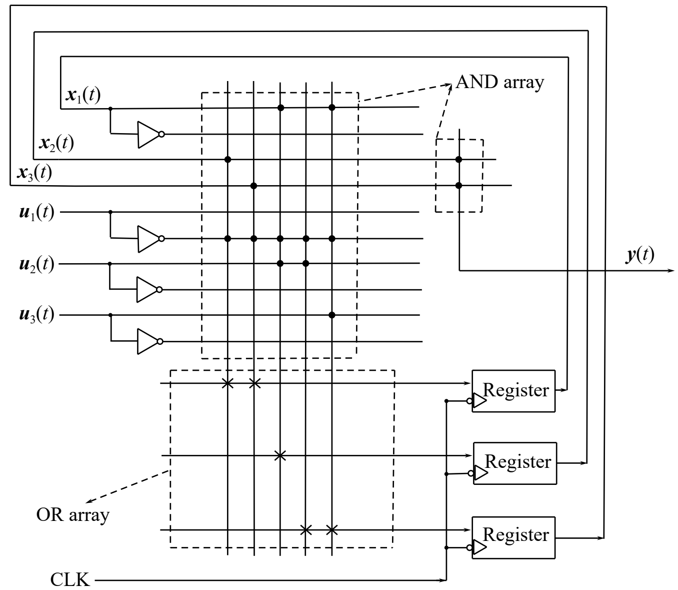

Fig. 4. A logic circuit with feedback paths.

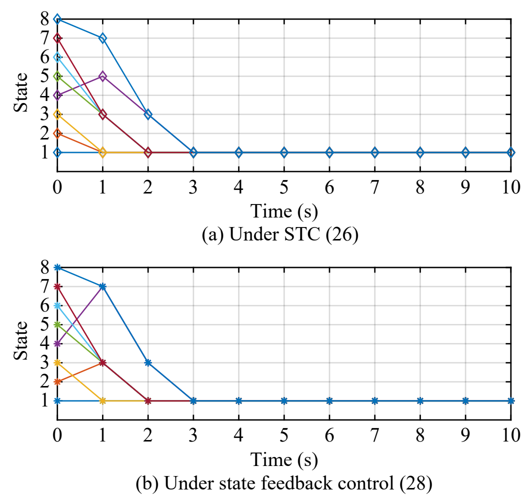

Fig. 5. State trajectories under STC (26) and state feedback controller (28) in Example 2.

the efficiency of our STC method. Besides, to further show the superiority of the developed STC method, employing the methods in [12], [13] and [14], [41], we design the sampled-data control $u(t)=\delta_8[5,6,6,5,5,6,6,5]x(t_l)$, $2l\le t<2(l+1)$, $l\in\mathbb{N}$ and output feedback control $u(t)=\delta_2[5,6]y(t)$, both of which can guarantee the $\mathcal{M}$-stabilization in minimum time. However, the resource waste that they produce is unavoidable due to unnecessary samples passed to the controller. The inter-execution intervals of the sampled-data control and output feedback control are presented in Fig. 6. One sees that the sampled-data control and output feedback control need to be updated 51 and 101 times, respectively, in the $[0,100]$ period, whereas the STCs (26) and (27) only need to be updated once. This means that the self-triggering implementations can significantly reduce system resource usage.

VI. Applications

In this section, the self-triggered set stabilization is applied

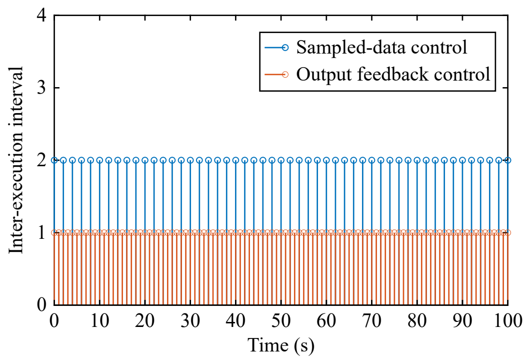

Fig. 6. Inter-execution intervals of the sampled-data control and output feedback control for Example 2.

to three typical set stabilization problems including synchronization, output tracking and output regulation of BCNs. As above-mentioned, they are a class of fundamental control problems existing widely in engineering practice, and have wide applications in biological systems, multi-agent systems, communication systems and other fields.

A. Self-Triggered Synchronization

Consider the following drive BN and response BCN:

$$x(t+1) = L_x x(t) \tag{29}$$

$$z(t+1) = L_z u(t)x(t)z(t) \tag{30}$$

where $x(t), z(t) \in \Delta_{2^n}$ are the states of (29) and (30), respectively, and $u(t) \in \Delta_{2^m}$ is the input of (30). Then, the notion of synchronization for drive-response BCNs is introduced.

Definition 5 [17]: Drive BN (29) and response BCN (30) are said to be synchronized, if for any initial states x_0 and z_0, there exist a control sequence U and an integer $T(x_0, z_0, U)$ such that $x(t; x_0) = z(t; x_0, z_0, U)$, $\forall t \geq T(x_0, z_0, U)$.

Let $\chi(t) = x(t)z(t) \in \Delta_{2^{2n}}$. Then an augmented system can be derived as follows:

$$\chi(t+1) = L_\chi u(t)\chi(t) \tag{31}$$

where $L_\chi = L_x(\mathbf{1}_{2^m}^T \otimes I_{2^n} \otimes \mathbf{1}_{2^n}^T) * L_z \in \mathcal{L}_{2^{2n}\times 2^{m+2n}}$. Denote by $\mathcal{M}_s = \{\delta_{2^{2n}}^{(i-1)2^n+i} | i = 1, 2, \ldots, 2^n\}$ the synchronized state set. Then the drive BN (29) and response BCN (30) are synchronized if and only if system (31) is $\mathcal{M}_s$-stabilizable. Subsequently, the results obtained in Sections IV and V can be applied to solve the synchronization problem by designing corresponding STCs.

Example 3: Consider the following coupled biochemical oscillators which were proposed in [17]:

$$\begin{cases} C_1(t+1) = \neg X_1(t) \\ M_1(t+1) = C_1(t) \\ X_1(t+1) = M_1(t) \end{cases} \tag{32}$$

$$\begin{cases} C_2(t+1) = \neg X_2(t) \vee \neg X_1(t) \\ M_2(t+1) = C_2(t) \\ X_2(t+1) = M_2(t) \vee \mathbf{u}(t) \end{cases} \tag{33}$$

where $C_i, M_i, X_i \in \mathbb{B}$ represent the cyclin, cdk and cdk-activated ubiquitin ligase, respectively, $i = 1, 2$, and $\mathbf{u} \in \mathbb{B}$ is an input. It is easy to derive the augmented system (31) with $L_\chi \in \mathcal{L}_{64\times 128}$. Immediately, $\mathcal{M}_s = \{\delta_{2^6}^{(i-1)2^3+i} | i \in [1, 8]\} = \{\delta_{64}^1, \delta_{64}^{10}, \delta_{64}^{19}, \delta_{64}^{28}, \delta_{64}^{37}, \delta_{64}^{46}, \delta_{64}^{55}, \delta_{64}^{64}\}$. By Propositions 1 and 2, we get that $\mathcal{I}_{\mathcal{M}_s} = \mathcal{M}_s \neq \emptyset$, and (31) is $\mathcal{M}_s$-stabilizable, i.e., (32) and (33) are synchronized. Then, we aim to design the minimum-time and minimum-triggering STCs for the synchronization of (32) and (33). Based on Theorem 1, we can design the minimum-time and minimum-triggering OLSTC for synchronization, which is shown in Table II. Consequently, there is a unique minimum-time and minimum-triggering SFSTC

$$u(t) = Kx(t_k),\ t \in [t_k, t_{k+1} - 1],\ t_{k+1} = t_k + \pi(x(t_k))$$

where $K \in \mathcal{L}_{2\times 64}$ and $\pi(x(t_k))$ are determined by

$$\begin{cases} \mathrm{Col}_i(K) = \delta_2^1, & \delta_{64}^i \in \Gamma_1 \bigcup \Gamma_2 \bigcup \Gamma_3 \bigcup \Gamma_4 \\ \mathrm{Col}_i(K) = \delta_2^2, & \delta_{64}^i \in \Gamma_5 \\ \pi(x(t_k)) = j, & x(t_k) \in \Gamma_j,\ \ j = 1, 2, 3, 4 \\ \pi(x(t_k)) = \infty, & x(t_k) \in \Gamma_5. \end{cases}$$

Taking the approaches of [17], [42], one can also design corresponding sampled-data control and state feedback control to make drive BN (32) and response BCN (33) achieve synchronization, but the sample frequencies of them are higher than our STC scheme. For instance, taking initial state δ_{64}^{22}, the inter-execution intervals of three kinds of controllers in $[0, 50]$ period are shown in Fig. 7, from which, the sampled-data control and state feedback control need to be updated 10 and 51 times, respectively, while the STC only needs to be updated twice.

B. Self-Triggered Output Tracking and Output Regulation

Consider BCN (2) and the following reference BN:

$$\begin{cases} \tilde{x}(t+1) = \tilde{L}\tilde{x}(t) \\ \tilde{y}(t) = \tilde{H}\tilde{x}(t) \end{cases} \tag{34}$$

where $\tilde{x}(t) \in \Delta_{2^{\tilde{n}}}$ and $\tilde{y}(t) \in \Delta_{2^p}$ are the state and output of (34). The output tracking and output regulation problems are introduced as follows.

Definition 6 [18], [19]: 1) Given a reference signal $y^* = \delta_{2^p}^\alpha \in \Delta_{2^p}$, the output tracking problem of BCN (2) is solvable, if for any initial state $x_0 \in \Delta_{2^n}$, there exist a control sequence U and an integer $T(x_0, U) > 0$ such that $y(t) = y^*$, $\forall t \geq T(x_0, U)$.

2) The output regulation problem of BCN (2) and reference BN (34) is solvable, if for any initial state $x_0 \in \Delta_{2^n}$, there exist a control sequence U and an integer $T(x_0, U) > 0$ such that $y(t) = \tilde{y}(t)$, $\forall t \geq T(x_0, U)$.

Set $\mathcal{M}_o = \{\delta_{2^n}^i | \mathrm{Col}_i(H) = y^*\}$. Then the output tracking problem is solvable, if and only if BCN (2) is $\mathcal{M}_o$-stabilizable. Combining with BCN (2) and reference BN (34), we derive the following augmented BCN:

$$\begin{cases} \tilde{\chi}(t+1) = L_{\tilde{\chi}} u(t)\tilde{\chi}(t) \\ y_{\tilde{\chi}}(t) = H_{\tilde{\chi}}\tilde{\chi}(t) \end{cases} \tag{35}$$

where $\tilde{\chi}(t) = x(t)\tilde{x}(t) \in \Delta_{2^{n+\tilde{n}}}$, $y_{\tilde{\chi}}(t) = y(t)\tilde{y}(t) \in \Delta_{2^{2p}}$, $L_{\tilde{\chi}} = L(I_{2^{m+n}} \otimes \mathbf{1}_{2^{\tilde{n}}}^T) * \tilde{L}(\mathbf{1}_{2^{m+n}}^T \otimes I_{2^{\tilde{n}}}) \in \mathcal{L}_{2^{n+\tilde{n}}\times 2^{n+\tilde{n}+m}}$ and $H_{\tilde{\chi}} = H(I_{2^n} \otimes \mathbf{1}_{2^{\tilde{n}}}^T) * \tilde{H}(\mathbf{1}_{2^n}^T \otimes I_{2^{\tilde{n}}}) \in \mathcal{L}_{2^{2p}\times 2^{n+\tilde{n}}}$. Assume $\mathcal{M}_{\tilde{o}} = \bigcup_{j=1}^{2^p} \Theta_j$, where $\Theta_j = \{\delta_{2^{n+\tilde{n}}}^i | \mathrm{Col}_i(H_{\tilde{\chi}}) = \delta_{2^{2p}}^{(j-1)2^p+j}\}$. Then, the output regulation problem is solvable, if and only if system (35) is $\mathcal{M}_{\tilde{o}}$-stabiliz-

TABLE II
THE OLSTC STRATEGY IN EXAMPLE 3

Initial states	Triggering moments	Corresponding controls
$\Gamma_1=\{\delta_{64}^{3},\delta_{64}^{39},\delta_{64}^{47},\delta_{64}^{48}\}$	$t_0=0,\ t_1=1$	$u(t_0)=\delta_2^1,\ u(t_1)=\delta_2^2$
$\Gamma_2=\{\delta_{64}^{5},\delta_{64}^{7},\delta_{64}^{21},\delta_{64}^{23},\delta_{64}^{35}\}$	$t_0=0,\ t_1=2$	$u(t_0)=\delta_2^1,\ u(t_1)=\delta_2^2$
$\Gamma_3=\{\delta_{64}^{4},\delta_{64}^{6},\delta_{64}^{8},\delta_{64}^{11},\delta_{64}^{12},\delta_{64}^{13},\delta_{64}^{14},\delta_{64}^{15},\delta_{64}^{16},\delta_{64}^{36},\delta_{64}^{40}\}$	$t_0=0,\ t_1=3$	$u(t_0)=\delta_2^1,\ u(t_1)=\delta_2^2$
$\Gamma_4=\{\delta_{64}^{22},\delta_{64}^{24},\delta_{64}^{29},\delta_{64}^{30},\delta_{64}^{31},\delta_{64}^{32}\}$	$t_0=0,\ t_1=4$	$u(t_0)=\delta_2^1,\ u(t_1)=\delta_2^2$
$\Gamma_5=\{\delta_{64}^{1},\delta_{64}^{2},\delta_{64}^{9},\delta_{64}^{10},\delta_{64}^{17},\delta_{64}^{18},\delta_{64}^{19},\delta_{64}^{20},\delta_{64}^{25},\delta_{64}^{26},\delta_{64}^{27},\delta_{64}^{28},\delta_{64}^{33},\delta_{64}^{34},\delta_{64}^{37},\delta_{64}^{38},\delta_{64}^{41},\delta_{64}^{42},\delta_{64}^{43},$ $\delta_{64}^{44},\delta_{64}^{45},\delta_{64}^{46},\delta_{64}^{49},\delta_{64}^{50},\delta_{64}^{51},\delta_{64}^{52},\delta_{64}^{53},\delta_{64}^{54},\delta_{64}^{55},\delta_{64}^{56},\delta_{64}^{57},\delta_{64}^{58},\delta_{64}^{59},\delta_{64}^{60},\delta_{64}^{61},\delta_{64}^{62},\delta_{64}^{63},\delta_{64}^{64}\}$	$t_0=0$	$u(t_0)=\delta_2^2$

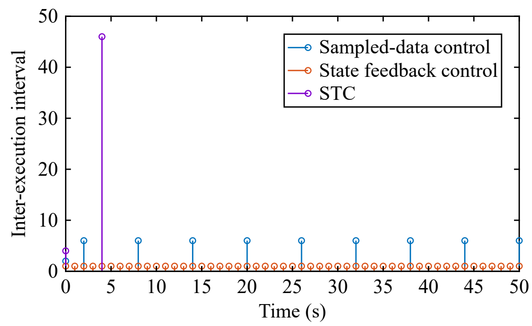

Fig. 7. Inter-execution intervals of the sampled-data control, state feedback control and STC for Example 3.

able. On the basis of these, one can design corresponding STCs for output tracking and output regulation.

Example 4: Consider the "minimal" model of the lactose operon in Escherichia coli [19]

$$\begin{cases}\mathbf{x}_1(t+1)=\neg\mathbf{u}_1(t)\wedge(\mathbf{x}_3(\mathbf{t})\vee\mathbf{u}_2(t))\\ \mathbf{x}_2(t+1)=\mathbf{x}_1(t)\\ \mathbf{x}_3(t+1)=\neg\mathbf{u}_1(t)\wedge((\mathbf{x}_2(t)\wedge\mathbf{u}_2(t))\\ \qquad\qquad\vee(\neg\mathbf{x}_2(t)\wedge\mathbf{x}_3(t)))\\ \mathbf{y}(t)=\mathbf{x}_1(t)\end{cases}\tag{36}$$

where $\mathbf{x}_1,\mathbf{x}_2,\mathbf{x}_3\in\mathbb{B}$ denote the mRNA, lacZ polypeptide, intracellular lactose, and $\mathbf{u}_1,\mathbf{u}_2\in\mathbb{B}$ are the external glucose and external lactose, respectively. The reference BN is another model for the lactose operon in Escherichia coli [19]

$$\begin{cases}\tilde{\mathbf{x}}_1(t+1)=\tilde{\mathbf{x}}_3(t)\\ \tilde{\mathbf{x}}_2(t+1)=\tilde{\mathbf{x}}_1(t)\\ \tilde{\mathbf{x}}_3(t+1)=\tilde{\mathbf{x}}_3(t)\vee(\tilde{\mathbf{x}}_2(t)\wedge\tilde{\mathbf{x}}_4(t))\\ \tilde{\mathbf{x}}_4(t+1)=\tilde{\mathbf{x}}_5(t)\vee(\neg\tilde{\mathbf{x}}_2(t)\wedge\tilde{\mathbf{x}}_4(t))\\ \tilde{\mathbf{x}}_5(t+1)=\tilde{\mathbf{x}}_1(t)\\ \tilde{\mathbf{y}}(t)=\tilde{\mathbf{x}}_1(t)\end{cases}\tag{37}$$

where $\tilde{\mathbf{x}}_1,\tilde{\mathbf{x}}_2,\tilde{\mathbf{x}}_3,\tilde{\mathbf{x}}_4,\tilde{\mathbf{x}}_5\in\mathbb{B}$ are the mRNA, β-galactosidase, allolactose, intracellular lactose and lac permease, respectively.

1) Given the reference signal $y^*=\delta_2^1$, we discuss the output tracking of BCN (36). Through routine calculations, we get $\mathcal{M}_o=\{\delta_8^1,\delta_8^2,\delta_8^3,\delta_8^4\}$. By Propositions 1 and 2, we have $\mathcal{I}_{\mathcal{M}_o}=\mathcal{M}_o\neq\emptyset$, and (36) is $\mathcal{M}_o$-stabilizable, i.e., the output tracking problem of (36) is solvable. Next, we design the STCs for the output tracking. In fact, it can be obtained from Theorem 1 that there is a unique minimum-time and minimum-triggering STC: $u(t)=u(t_0)=\delta_4^3,\ t\in[t_0,\infty)$.

2) Consider the output regulation of BCN (36) and reference BN (37). First, we can convert (36) and (37) into the augmented system (35). And then, $\mathcal{M}_{\tilde{o}}=\bigcup_{j=1}^{2}\Theta_j=\{\delta_{28}^i|\mathrm{Col}_i(H_{\tilde{\chi}})=\delta_4^1\text{ or }\mathrm{Col}_i(H_{\tilde{\chi}})=\delta_4^4\}$ can be derived. By Propositions 1 and 2, we get that $\mathcal{I}_{\mathcal{M}_{\tilde{o}}}=\mathcal{M}_{\tilde{o}}\neq\emptyset$, and (35) is $\mathcal{M}_{\tilde{o}}$-stabilizable. That is, the output regulation problem of (36) and (37) is solvable. Skipping some tedious steps, the minimum-time and minimum-triggering OLSTC can be designed for the output regulation of (36) and (37), which is shown in Table III. Furthermore, 6^{64} minimum-time and minimum-triggering SFSTCs are obtained: $u(t)=Kx(t_k),\ t\in[t_k,t_{k+1}-1],\ t_{k+1}=t_k+\pi(x(t_k))$, where $K\in\mathcal{L}_{4\times256}$ and $\pi(x(t_k))$ are determined by

$$\begin{cases}\mathrm{Col}_i(K)=\delta_4^3, & \delta_{256}^i\in\tilde{\Gamma}_1\\ \mathrm{Col}_i(K)\in\{\delta_4^1,\delta_4^2\}, & \delta_{256}^i\in\tilde{\Gamma}_2\bigcup\tilde{\Gamma}_4\bigcup\tilde{\Gamma}_5\\ \mathrm{Col}_i(K)\in\{\delta_4^1,\delta_4^2,\delta_4^4\}, & \delta_{256}^i\in\tilde{\Gamma}_3\bigcup\tilde{\Gamma}_6\bigcup\tilde{\Gamma}_7\\ \pi(x(t_k))=1, & x(t_k)\in\tilde{\Gamma}_4\bigcup\tilde{\Gamma}_6\\ \pi(x(t_k))=2, & x(t_k)\in\tilde{\Gamma}_5\bigcup\tilde{\Gamma}_7\\ \pi(x(t_k))=\infty, & x(t_k)\in\tilde{\Gamma}_1\bigcup\tilde{\Gamma}_2\bigcup\tilde{\Gamma}_3.\end{cases}$$

But, there is no minimum-time and minimum-triggering OFSTC for the output regulation of (36) and (37). In comparison, 4^{240} state feedback controllers were obtained in [19]. Although the number of controllers obtained in [19] is abundant, continuous updating of the controllers leads to resource waste inevitably. Fortunately, the minimum-time and minimum-triggering STCs designed in this paper can effectively deal with this issue.

VII. CONCLUDING REMARKS

In this paper, set stabilization of BCNs and corresponding STC strategies have been investigated. Using Tarjan's algorithm, we have given graph-theoretical conditions for set stabilization based on SCCs. In light of a characteristic function, we have designed the minimum-time and minimum-triggering OLSTCs, SFSTCs and OFSTCs for set stabilization. Compared with the existing works, the graph-theoretical criteria have reduced the computational complexity (Remark 2), and our design approach for STCs is more general and effective (Remark 4). Meanwhile, several crucial applications: self-triggered synchronization, output tracking and output regulation have been also considered.

As already pointed out, STCs are of great research importance in reducing energy consumption and saving network resources. In this study, only self-triggered set stabilization is

TABLE III
THE OLSTC STRATEGY IN EXAMPLE 4

Initial states	Triggering moments	Corresponding controls
$\tilde{\Gamma}_1=\left\{\delta_{256}^i \mid i\in\bigcup_{j=1}^{32}[a_j,a_j+3], a_j=(j-1)8+1\right\}$	$t_0=0$	$u(t_0)=\delta_4^3$
$\tilde{\Gamma}_2=\left\{\delta_{256}^i \mid i\in\bigcup_{j=1}^{4}\{a_j,a_j+8,a_j+15,a_j+16,a_j+21,\ a_j+22,a_j+23,a_j+24\}, a_j=(j-1)64+8\right\}$	$t_0=0$	$u(t_0)\in\{\delta_4^1,\delta_4^2\}$
$\tilde{\Gamma}_3=\left\{\delta_{256}^i \mid i\in\bigcup_{j=1}^{4}\{a_j,a_j+8,a_j+15,a_j+16,a_j+21,\ a_j+22,a_j+23,a_j+24\}, a_j=(j-1)64+40\right\}$	$t_0=0$	$u(t_0)\in\{\delta_4^1,\delta_4^2,\delta_4^4\}$
$\tilde{\Gamma}_4=\left\{\delta_{256}^i \mid i\in\bigcup_{j=1}^{4}\{a_j,a_j+1,a_j+16,a_j+17\}, a_j=(j-1)64+5\right\}$	$t_0=0, t_1=1$	$u(t_0)\in\{\delta_4^1,\delta_4^2\}, u(t_1)=\delta_4^3$
$\tilde{\Gamma}_5=\left\{\delta_{256}^i \mid i\in\bigcup_{j=1}^{4}\{a_j,a_j+6,a_j+7,a_j+8\}, a_j=(j-1)64+7\right\}$	$t_0=0, t_1=2$	$u(t_0)\in\{\delta_4^1,\delta_4^2\}, u(t_1)=\delta_4^3$
$\tilde{\Gamma}_6=\left\{\delta_{256}^i \mid i\in\bigcup_{j=1}^{4}\{a_j,a_j+1,a_j+15,a_j+16\}, a_j=(j-1)64+37\right\}$	$t_0=0, t_1=1$	$u(t_0)\in\{\delta_4^1,\delta_4^2,\delta_4^4\}, u(t_1)=\delta_4^3$
$\tilde{\Gamma}_7=\left\{\delta_{256}^i \mid i\in\bigcup_{j=1}^{4}\{a_j,a_j+6,a_j+7,a_j+8\}, a_j=(j-1)64+39\right\}$	$t_0=0, t_1=2$	$u(t_0)\in\{\delta_4^1,\delta_4^2,\delta_4^4\}, u(t_1)=\delta_4^3$

considered. There are still a lot of classical control problems, such as observability, detectability and disturbance decoupling, deserving further investigation. On the other hand, it is also interesting and significant to scale-up the STC method proposed in this paper to large-scale BNs by combining the network structures with pinning control [16], network aggregation [43] and bisimulation [44].

REFERENCES

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

[2] H. Kitano, "Systems biology: A brief overview," *Science*, vol. 295, no. 5560, pp. 1662–1664, 2002.

[3] D. Cheng, H. Qi, and Y. Zhao, *An Introduction to Semi-Tensor Product of Matrices and Its Application*. Singapore: World Scientific, 2011.

[4] X. Han, J. Wang, Z. Li, X. Chen, and Z. Chen, "Revisiting state estimation and weak detectability of discrete-event systems," *IEEE Trans. Autom. Science and Engineering*, vol. 20, no. 1, pp. 662–674, 2022.

[5] G. Zhao, Y. Wang, and H. Li, "A matrix approach to the modeling and analysis of networked evolutionary games with time delays," *IEEE/CAA J. Autom. Sinica*, vol. 5, no. 4, pp. 818–826, 2018.

[6] S. Gao, R. Wu, X. Wang, J. Liu, Q. Li, and X. Tang, "EFR-CSTP: Encryption for face recognition based on the chaos and semi-tensor product theory," *Inform. Sciences*, vol. 621, pp. 766–781, 2023.

[7] D. Cheng, H. Qi, Z. Li, and J. Liu, "Stability and stabilization of Boolean networks," *Int. J. Robust and Nonlinear Control*, vol. 21, no. 2, pp. 134–156, 2011.

[8] J. Liang, H. Chen, and Y. Liu, "On algorithms for state feedback stabilization of Boolean control networks," *Automatica*, vol. 84, pp. 10–16, 2017.

[9] H. Li and Y. Wang, "Further results on feedback stabilization control design of Boolean control networks," *Automatica*, vol. 83, pp. 303–308, 2017.

[10] Y. Guo, P. Wang, W. Gui, and C. Yang, "Set stability and set stabilization of Boolean control networks based on invariant subsets," *Automatica*, vol. 61, pp. 106–112, 2015.

[11] S. Zhu, J. Feng, and J. Zhao, "State feedback for set stabilization of Markovian jump Boolean control networks," *Discrete & Continuous Dynamical Systems-Series S*, vol. 14, no. 4, 2021.

[12] Y. Jia, B. Wang, J. Feng, and D. Cheng, "Set stabilization of Boolean control networks via output-feedback controllers," *IEEE Trans. Systems, Man, and Cybern.: Systems*, vol. 52, no. 12, pp. 7527–7536, 2022.

[13] R. Liu, J. Lu, W. Zheng, and J. Kurths, "Output feedback control for set stabilization of Boolean control networks," *IEEE Trans. Neural Networks and Learning Systems*, vol. 31, no. 6, pp. 2129–2139, 2020.

[14] S. Zhu, Y. Liu, J. Lou, J. Lu, and F. E. Alsaadi, "Sampled-data state feedback control for the set stabilization of Boolean control networks," *IEEE Trans. Systems, Man, and Cybern.: Systems*, vol. 50, no. 4, pp. 1580–1589, 2020.

[15] Q. Zhang, J. Feng, B. Wang, and P. Wang, "Event-triggered mechanism of designing set stabilization state feedback controller for switched Boolean networks," *Applied Math. and Computation*, vol. 383, p. 125372, 2020.

[16] S. Zhu, J. Lu, L. Sun, and J. Cao, "Distributed pinning set stabilization of large-scale Boolean networks," *IEEE Trans. Autom. Control*, vol. 68, no. 3, pp. 1886–1893, 2023.

[17] Y. Liu, L. Sun, J. Lu, and J. Liang, "Feedback controller design for the synchronization of Boolean control networks," *IEEE Trans. Neural Networks and Learning Systems*, vol. 27, no. 9, pp. 1991–1996, 2016.

[18] X. Zhang, Y. Wang, and D. Cheng, "Output tracking of Boolean control networks," *IEEE Trans. Autom. Control*, vol. 65, no. 6, pp. 2730–2735, 2020.

[19] H. Li, L. Xie, and Y. Wang, "Output regulation of Boolean control networks," *IEEE Trans. Automatic Control*, vol. 62, no. 6, pp. 2993–2998, 2017.

[20] K. Zhang, Y. Liu, and J. Tan, "Finite-time stabilization of linear systems with input constraints by event-triggered control," *IEEE/CAA J. Autom. Sinica*, vol. 9, no. 8, pp. 1516–1519, 2022.

[21] X. Wang, J. Sun, G. Wang, F. Allgower, and J. Chen, "Data-driven control of distributed event-triggered network systems," *IEEE/CAA J. Autom. Sinica*, vol. 10, no. 2, pp. 351–364, 2023.

[22] K. E. Aarzen, "A simple event-based PID controller," *IFAC Proceedings Volumes*, vol. 32, pp. 8687–8692, 1999.

[23] M. Velasco, J. Fuertes, and P. Marti, "The self triggered task model for real-time control systems," *Work-in-Progress Session 24th IEEE Real-Time Systems Symposium*, vol. 384, pp. 67–70, 2003.

[24] Y. Ju, D. Ding, X. He, Q.-L. Han, and G. Wei, "Consensus control of multi-agent systems using fault-estimation-in-the-loop: Dynamic event-triggered case," *IEEE/CAA J. Autom. Sinica*, vol. 9, no. 8, pp. 1440–1451, 2022.

[25] X. Ge, S. Xiao, Q.-L. Han, X.-M. Zhang, and D. Ding, "Dynamic event-triggered scheduling and platooning control co-design for automated vehicles over vehicular ad-hoc networks," *IEEE/CAA J. Autom. Sinica*, vol. 9, no. 1, pp. 31–46, 2022.

[26] Y. Su, Q. Wang, and C. Sun, "Self-triggered consensus control for linear multi-agent systems with input saturation," *IEEE/CAA J. Autom. Sinica*, vol. 7, no. 1, pp. 150–157, 2020.

[27] L. Lin, J. Cao, M. Abdel Aty, and U. A. Al Juboori, "Topological structure, reachability, and stabilization of constrained Boolean control networks via event-triggered control," *IEEE Trans. Systems, Man, and*

Cybern.: Systems, vol. 51, no. 6, pp. 3565–3574, 2021.

[28] B. Chen, J. Cao, G. Lu, and L. Rutkowski, "Stabilization of Markovian jump Boolean control networks via event triggered control," *IEEE Trans. Autom. Control*, vol. 68, no. 2, pp. 1215–1222, 2023.

[29] Q. Zhu and W. Lin, "Stabilizing Boolean networks by optimal event-triggered feedback control," *Systems & Control Lett.*, vol. 126, pp. 40–47, 2019.

[30] S. Zhu, J. Lu, L. Lin, and Y. Liu, "Minimum-time and minimum-triggering control for the observability of stochastic Boolean networks," *IEEE Trans. Autom. Control*, vol. 67, no. 3, pp. 1558–1565, 2022.

[31] L. Lin, J. Cao, and L. Rutkowski, "Robust event-triggered control invariance of probabilistic Boolean control networks," *IEEE Trans. Neural Networks and Learning Systems*, vol. 31, no. 3, pp. 1060–1065, 2020.

[32] A. Acernese, A. Yerudkar, L. Glielmo, and C. Del Vecchio, "Model-free self-triggered control co-design for probabilistic Boolean control networks," *IEEE Control Systems Lett.*, vol. 5, no. 5, pp. 1639–1644, 2020.

[33] M. Meng, G. Xiao, and D. Cheng, "Self-triggered scheduling for Boolean control networks," *IEEE Trans. Cybern.*, vol. 52, no. 9, pp. 8911–8921, 2022.

[34] P. Bajaria, A. Yerudkar, L. Glielmo, C. Del Vecchio, and Y. Wu, "Self-triggered control of probabilistic Boolean control networks: A reinforcement learning approach," *J. Franklin Institute*, vol. 359, no. 12, pp. 6173–6195, 2022.

[35] Y. Yu, M. Meng, J. Feng, and P. Wang, "Stabilizability analysis and switching signals design of switched Boolean networks," *Nonlinear Analysis: Hybrid Systems*, vol. 30, pp. 31–44, 2018.

[36] K. H. Rosen, *Discrete Mathematics and Its Applications*. New York, USA: McGraw-Hill, 1999.

[37] R. Tarjan, "Depth-first search and linear graph algorithms," *SIAM J. Computing*, vol. 1, no. 2, pp. 146–160, 1972.

[38] D. Laschov and M. Margaliot, "Minimum-time control of Boolean networks," *SIAM J. Control and Optimization*, vol. 51, no. 4, pp. 2869–2892, 2013.

[39] P. Rössler and R. Höller, "Programmable logic devices-key components for todays and tomorrows electronic-based systems," *Elektrotechnik UND Informationstechnik*, vol. 137, pp. 45–51, 2020.

[40] J. Gao, S. Lin, J. Fan, and Y. Hu, "An average optimal control approach to the set stabilization problem for Boolean control networks," *Asian J. Control*, vol. 21, no. 6, pp. 2596–2603, 2019.

[41] L. Sun, J. Lu, J. Lou, and L. Li, "Set stabilization of Boolean networks via sampled-data control," *Asian J. Control*, vol. 21, no. 6, pp. 2685–2690, 2019.

[42] Y. Liu, L. Tong, J. Lou, J. Lu, and J. Cao, "Sampled-data control for the synchronization of Boolean control networks," *IEEE Trans. Cybern.*, vol. 49, no. 2, pp. 726–732, 2019.

[43] H. Li, Y. Liu, S. Wang, and B. Niu, "State feedback stabilization of large-scale logical control networks via network aggregation," *IEEE Trans. Autom. Control*, vol. 66, no. 12, pp. 6033–6040, 2021.

[44] R. Li, Q. Zhang, and T. Chu, "Bisimulations of probabilistic Boolean networks," *SIAM J. Control and Optimization*, vol. 60, no. 5, pp. 2631–2657, 2022.

Rong Zhao received the B.S. degree in mathematics and applied mathematics from the School of Mathematics and Statistics, Shandong Normal University in 2020. She is currently a Ph.D. candidate in operational research and cybernetics at the School of Mathematics, Shandong University. Her research interests include control theory and application, logical networks, singular systems, game theory.

Jun-e Feng received the B.S. degree in mathematics from Liaocheng Normal University (now Liaocheng University) in 1994, and the M.S. and Ph.D. degrees in mathematics from the School of Mathematics, Shandong University in 1997 and 2003, respectively. She was a Visiting Scholar at Massachusetts Institute of Technology, USA, from 2006 to 2007, and a Visiting Scholar at the University of Hong Kong, China, in 2008, 2009, and 2013. She is currently a Professor and Doctoral Supervisor the at School of Mathematics, Shandong University. Her research interests include singular systems, Boolean networks, and robust control. She serves as a Reviewer of *American Mathematical Review*, a Member of the "Cyber-Physical System Control and Decision Professional Committee" of the Chinese Society of Automation, and the Editorial Board Member of *IEEE Control Systems Society*. She has authored or coauthored more than 100 SCI papers and more than 50 EI papers.

Dawei Zhang received the B.S. degree in information and computing science and the M.Sc. degree in pattern recognition and intelligent systems from Shanxi University in 2005 and 2008, respectively, and the Ph.D. degree in computer science from Central Queensland University, Australia in 2012. From Jan. 2013 to Dec. 2018, he worked in the School of Mathematical Sciences, Shanxi University. From Feb. 2017 to Feb. 2018, he was a Post-Doctoral Research Fellow with the School of Software and Electrical Engineering, Swinburne University of Technology, Australia. He is currently a Professor with the School of Mathematics, Shandong University. His research interests include networked intelligent control, vehicle control and state estimation.