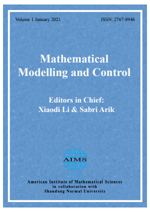
Mathematical Modelling and Control

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



MMC, 3(1): 61–72
DOI:10.3934/mmc.2023006
Received: 23 September 2022
Revised: 28 January 2023
Accepted: 07 February 2023
Published: 15 February 2023

Research article

Distributed pinning controllers design for set stabilization of *k*-valued logical control networks

Yanfei Wang, Changxi Li and Jun-e Feng*

School of Mathematics, Shandong University, Jinan 250100, China

* **Correspondence:** Email: fengjune@sdu.edu.cn.

Abstract: Design of distributed pinning controllers for set stabilization of k-valued logical control networks is investigated in this paper. Firstly, by analyzing the coupling correlations among nodes, pinned node set is determined. Secondly, based on the solvability of a class of matrix equations, sufficient conditions which resort to local information are put forward for the design of pinning controllers. Furthermore, an algorithm for designing pinning feedback controllers is presented, where the designed controllers are related to part of state variables instead of all variables. Finally, two illustrative examples are presented to verify the effectiveness of the main results.

Keywords: distributed pinning control; k-valued logical network; semi-tensor product of matrices

1. Introduction

The Boolean network (BN) was first proposed by Kauffman to simulate gene regulatory networks [1]. As an extension of BNs, k-valued logical networks (LNs) and logical control networks (LCNs) were presented for the study of cellular networks. The difference between BNs and LNs is that the nodes of LNs take values from $\{0, 1, \cdots, k-1\}$ while that of BNs take values from $\{0, 1\}$. As LNs and LCNs are more general mathematical models, they have attracted considerable attention from various areas [2, 3]. Although LNs and LCNs are useful tools in the investigation of cellular networks, it is not until the emergence of the Cheng product that both of them develop rapidly [4]. As a powerful tool for the analysis and control of LNs and LCNs, Cheng product, also called the semi-tensor product (STP), was first proposed by Prof. Cheng and his colleagues. With the help of Cheng product, the dynamics of LNs and LCNs can be converted into equivalent algebraic forms [5]. Various research results have been obtained, including but not limited to controllability [6–9], stability and stabilization [10–12], synchronization [13, 14], decoupling problem [15, 16], and output tracking control of LCNs [17].

Stabilization, one of the fundamental problems of LCNs, aims to design controllers to stabilize a given LCN to a desired state. As a more general case, set stabilization was investigated in [18], aiming to stabilize LCNs to a given state subset. To stabilize a given LCN to a state or a state subset, various controllers have been designed, such as event-triggered controllers [19], state-feedback controllers [20, 21], output feedback controllers [17] and so on. The common feature of the controllers mentioned above is that all nodes need to be controlled. But in many practical systems, the desired control objective can be achieved by only controlling part of essential nodes instead of all nodes. For instance, by only controlling node Mdm2 or Wip1, a p53 network can be steered into the apoptosis attractor in the presence of DNA damage. Motivated by it, the pinning control strategy was proposed in [22].

As a novel and effective approach, the pinning control technique makes systems attain the desired behavior by controlling a small fraction of nodes. Using the pinning approach, the controllability and reachability [22, 23], output tracking problem [24] and disturbance decoupling problem [25] were investigated. In addition, pinning controllers were designed for the stabilization and set stabilization of LCNs [26, 27]. A pinning control design method proposed by [26] is called the transition matrix-based pinning approach. By changing columns of the transition matrix and solving a series of logical matrix equations, pinning controllers were devised to stabilize a given LCN to a desired state. But the design of transition matrix-based pinning controllers relies on global information, and the computational complexity is quite large. To overcome the above weaknesses, distributed pinning controllers for the set stabilization of BNs were designed in [27], which has been successfully used to deal with the T-LGL survival signaling network and T-cell receptor signaling network.

It is worth noting that the distributed pinning controllers have not been employed to study the set stabilization of LCNs. Owing to its lower computational complexity, this paper investigates the distributed pinning controller design for the set stabilization of LCNs. There are three difficulties in the process of the distributed pinning controllers design. Firstly, it is difficult to associate the solution of a k-valued matrix equation with the acquisition of a pinning feedback controller. Secondly, selecting the pinned nodes is not easy due to the intricate coupling correlations among nodes. Finally, there is no unified method to determine proper control functions and logical couplings for LCNs.

The main contributions of this paper are three folds:

(i) For the nodes with fixed desired states, pinned node set is determined in accordance with the associated network graph, which consists of two disjoint subsets: one gathers nodes with arcs to be deleted, and the other one is a collection of nodes without deleted arcs.

(ii) The existence of pinning feedback controllers is obtained. And a novel method is proposed to devise the distributed pinning controllers for the set stabilization of LCNs.

(iii) The computational complexity of the proposed method is $O(n^2 + ps^3 + sk^{\varepsilon})$, which is lower than the transition matrix-based pinning approach in [26]. s is the number of fixed-state nodes, and p is the sum of the number of state variables for all fixed-state nodes. ε is the maximum in-degree of the pinned nodes.

The rest of this paper is organized as follows: Section 2 provides some necessary preliminaries. Section 3 investigates how to design distributed pinning controllers. Section 4 proposes two illustrative examples to verify the effectiveness of the main results. A brief conclusion is given in Section 5.

2. Preliminaries

For convenience of description, we give some necessary preliminaries. Some notations are provided as follows:

- $\mathcal{M}_{m\times n}$, $\mathbb{N}$, and $\mathbb{Z}_+$ are the set of all $m \times n$ real matrices, natural numbers, and positive integers, respectively.
- $\mathbb{R}^n$ is n dimensional Euclidean space.
- $\mathcal{D}_k := \{0, 1, 2, \cdots, k-1\}$. Especially, $\mathcal{D} = \{0, 1\}$.
- δ_n^i is the i-th column of n dimensional identity matrix I_n.
- $\Delta_n = \{\delta_n^i | i = 1, 2, \cdots, n\}$.
- $[m : n]$ is the set of all positive integers from m to n.
- $\mathbf{1}_n := [\underbrace{1, \cdots, 1}_{n}]^T$.
- $Col_i(M)$ is the i-th column of matrix M.
- A matrix $L \in \mathcal{M}_{m\times n}$ is called a logical matrix if $Col_i(L) \in \Delta_m$, $i = 1, 2, \cdots, n$. And $\mathcal{L}_{m\times n}$ is the set of all $m \times n$ logical matrices.
- A logical matrix $L = [\delta_m^{i_1}, \delta_m^{i_2}, \cdots, \delta_m^{i_n}]$ is briefly denoted as $L = \delta_m[i_1, i_2, \cdots, i_n]$.
- $\cap$, $\cup$ and $-$ are *intersection*, *union* and *difference* of finite sets, respectively.
- $\vee, \wedge, \neg, \leftrightarrow$ denote the logical operators *disjunction*, *conjunction*, *negation* and *bi-conditional*, respectively.

By identifying $i \sim \delta_k^{k-i}$, a logical variable $x \in \mathcal{D}_k$ can be expressed by a k dimensional column vector. Thus logical operations can be transformed into algebraic operations.

Definition 2.1. *[28] Given two matrices $A \in \mathcal{M}_{m\times n}$ and $B \in \mathcal{M}_{p\times q}$. The (left) semi-tensor product of A and B, denoted by $A \ltimes B$, is defined as*

$$A \ltimes B := (A \otimes I_{t/n})(B \otimes I_{t/p}),$$

where $t = lcm(n, p)$ is the least common multiple of n and p, and $\otimes$ is the Kronecker product.

Remark 2.1. *(i) The right STP can be similarly defined [29]. Compared with the right STP, the left STP has* *more superior properties. For example, it satisfies the block multiplication of matrices. Therefore, only the left STP is considered in this paper, and it is referred to as the STP for short.*

(ii) STP is a generalization of traditional matrix product, which preserves almost all properties of traditional matrix product. Thus the matrix product in this paper defaults to STP, and the symbol $\ltimes$ is often omitted.

Definition 2.2. *[30] Let $x_i \in \mathcal{D}_k, i = 1, 2, \cdots, n$. A mapping $f : \mathcal{D}_k^n \to \mathcal{D}_k$, denoted by $y = f(x_1, x_2, \cdots, x_n)$, is called a k-valued logical function.*

Proposition 2.1. *[31] For a given k-valued logical function $f : \mathcal{D}_k^n \to \mathcal{D}_k$, there exists a unique structure matrix $M_f \in \mathcal{L}_{k\times k^n}$, such that f is expressed in the vector form as*

$$f(x_1, x_2, \cdots, x_n) = M_f \ltimes_{i=1}^n x_i.$$

Definition 2.3. *[31] (1) A (p, q)-th dimensional swap matrix is defined as*

$$W_{[p,q]} = [I_q \otimes \delta_p^1, I_q \otimes \delta_p^2, \cdots, I_q \otimes \delta_p^p].$$

(2) $F_{[m,n]}$ and $R_{[m,n]}$ are called (m, n)-th dimensional dummy matrices, where

$$F_{[m,n]} = I_m \otimes \mathbf{1}_n^T, R_{[m,n]} = \mathbf{1}_m^T \otimes I_n.$$

(3) An m-th dimensional power reducing matrix is defined as

$$R_m^P = diag\{\delta_m^1, \delta_m^2, \cdots, \delta_m^m\}.$$

Proposition 2.2. *[31] Let $X \in \mathbb{R}^p$, $Y \in \mathbb{R}^q$, $x \in \Delta_m$, $y \in \Delta_n$, and A is a real matrix, then*

$$XA = (I_p \otimes A)X,$$

$$W_{[p,q]}XY = YX,$$

$$x^2 = R_m^P x,$$

$$F_{[m,n]}xy = x, R_{[m,n]}xy = y.$$

Lemma 2.1. *[32] Let $f(x_1, x_2, \cdots, x_n)$ be a k-valued logical function, with $L = [L_1, L_2, \cdots, L_{k^{n-1}}] \in \mathcal{L}_{k\times k^n}$ being its structure matrix and $L_j \in \mathcal{L}_{k\times k}, j = 1, 2, \cdots, k^{n-1}$. Then the logical (disjunctive normal) form of f can be expressed as*

$$f = \bigvee_{i_1=1}^{k} \cdots \bigvee_{i_{n-1}=1}^{k} (\triangleright_k^{i_1}(x_1) \wedge \triangleright_k^{i_2}(x_2) \wedge \cdots \wedge \triangleright_k^{i_{n-1}}(x_{n-1}) \wedge \phi_j(x_n)),$$

where

$$j = (i_1 - 1)k^{n-1} + (i_2 - 1)k^{n-2} + \cdots + (i_{n-2} - 1)k + i_{n-1},$$

$\triangleright_k^i$ and ϕ_j are unary mappings with $M_{\triangleright_k^i}$ and L_j being their structure matrices respectively, and

$$M_{\triangleright_k^i} = \delta_k[k, k, \cdots, \underbrace{1}_{i\text{-}th}, \cdots, k], i = 1, 2, \cdots, k.$$

3. Design of distributed pinning controllers

3.1. Problem formulation

The dynamics of k-valued logical networks can be described as

$$\begin{cases} x_1(t + 1) = f_1(\{x_j(t)|j \in \mathcal{N}_1\}), \\ x_2(t + 1) = f_2(\{x_j(t)|j \in \mathcal{N}_2\}), \\ \quad \vdots \\ x_n(t + 1) = f_n(\{x_j(t)|j \in \mathcal{N}_n\}), \end{cases} \tag{3.1}$$

where $x_i \in \mathcal{D}_k$ denotes the state variable of node i, and $f_i : \mathcal{D}_k^n \to \mathcal{D}_k$ are logical functions. $\mathcal{N}_i$ is the set of in-neighbors of node i, which will be introduced in detail in the next paragraph.

For logical network (3.1), we associate it with a directed network graph $G := (\mathcal{N}, \mathcal{E})$, which consists of a labeled vertex set $\mathcal{N} = \{1, 2, \cdots, n\}$ and an arc set $\mathcal{E}$. The vertex labeled by i corresponds to the node i, and there exists an arc from vertex j to i if and only if there exists an interaction between x_j and x_i. Given an arc from j to i, j and i are called the tail and head of this arc respectively. Besides, j is called the in-neighbor of i. For two vertices i_1 and i_k, a sequence $i_1 i_2 \cdots i_k$ is called a path from i_1 to i_k, if it satisfies $i_j \neq i_s$, $1 \leq j \neq s \leq k$, and there exists an arc from i_j to i_{j+1}, $j = 1, 2, \cdots, k - 1$. Especially, if $i_1 = i_k$, it is called a cycle. If there exists no cycle in G, then G is said to be acyclic.

The logical network (3.1) with external control inputs is expressed as

$$x_i(t + 1) = \begin{cases} u_i(t) \oplus_i f_i(\{x_j(t)|j \in \mathcal{N}_i\}), \ i \in \Theta, \\ f_i(\{x_j(t)|j \in \mathcal{N}_i\}), \ i \notin \Theta, \end{cases} \tag{3.2}$$

where $u_i(t)$ is the control input, $\oplus_i$ is a k-valued binary logical operator which couples the control u_i and logical function f_i. And Θ denotes the node set to be controlled, which will be discussed in detail in Subsection 3.2. Furthermore, the control $u_i(t)$ can be either open-loop control or closed-loop control $u_i(t) = \mu_i(\{x_j(t)|j \in \mathcal{N}_i\})$, where μ_i is the state feedback control function.

Definition 3.1. *[26] A logical network* (3.2) *is said to be globally stabilized to the given state set $\Lambda \subseteq \mathcal{D}_k^n$, if for every initial state $x(0) := x_0 \in \mathcal{D}_k^n$, there exists a control sequence $\mathbf{u(t)} = \{u(0), u(1), \cdots, u(t) : t \in \mathbb{N}\}$ and a positive integer T, such that $x(t; \mathbf{u(t)}, x_0) \in \Lambda$ holds, for every $t \geq T$.*

Lemma 3.1. *[33] Logical network* (3.2) *can be globally stabilized at a certain state, if its corresponding network graph is acyclic.*

Definition 3.2. *[34] In a digraph G, a set of arcs S is called a feedback arc set if $G - S$ is acyclic. And if its cardinality is minimum, it is called a minimum feedback arc set.*

For the k-valued logical network (3.1) and given subset Λ, we devote to designing controllers to convert network (3.1) to (3.2), such that (3.2) is globally stabilized to set Λ. Set Λ is said to be the desired state set, and the i-th element of each state in Λ is called the desired state of node i. Denote $u_i(t) \oplus_i f_i(\{x_j(t)|j \in \mathcal{N}_i\})$ as $\vec{f_i}$, where $\vec{f_i}$ is called the desired dynamics of node i.

In this paper, there are three components that need to be determined: pinned node set Θ, feedback control functions μ_i and logical couplings $\oplus_i$.

3.2. Determining pinned nodes

In this subsection, we discuss how to obtain set $\Theta \subseteq \mathcal{N}$ with respect to the desired state set Λ.

Without loss of generality, we consider the desired state set Λ which has the following form

$$\Lambda = \{(x_1 = \vec{\xi}_1, x_2 = \vec{\xi}_2, \cdots, x_s = \vec{\xi}_s, x_{s+1}, \cdots, x_n)\} \subseteq \Delta_k^n,$$

where $\vec{\xi}_i \in \Delta_k$ is the desired state of $x_i, i = 1 \cdots s$, and there exists no restriction on the desired state of $x_i, i = s+1, \cdots, n$.

Based on the desired states of all nodes, we first divide node set $\mathcal{N}$ roughly into Γ and Γ^c as

$$\Gamma = [1 : s], \Gamma^c = [s + 1 : n],$$

where Γ gathers nodes whose desired states are fixed ones, and Γ^c is a collection of nodes whose desired states can be arbitrary.

For each node i in Γ, we consider all arcs with i being their head in G. According to the desired state set Λ, in order to make i be unaffected by nodes in set Γ^c whose desired states are arbitrary ones, all arcs from Γ^c to i need to be deleted. Denote the tails of these deleted arcs as $\hat{\mathcal{N}}_i^d \subseteq \mathcal{N}_i$. According to Lemma 3.1, an acyclic graph is required to ensure that the logical network can be globally stabilized to a certain state. To get the acyclic graph, find the minimum feedback arc set to be deleted using the algorithm proposed in [35]. And denote the tails of these deleted arcs as $\check{\mathcal{N}}_i^d$. Let

$$\mathcal{N}_i^d = \hat{\mathcal{N}}_i^d \cup \check{\mathcal{N}}_i^d,$$

where $\mathcal{N}_i^d \subseteq \mathcal{N}_i$ is the tail set of all deleted arcs of i.

Then we consider Γ even further, take

$$\Theta_1 = \{i \in \Gamma | \mathcal{N}_i^d \neq \emptyset\},$$

where Θ_1 is the node set in which for each $i \in \Gamma$, there exist arcs to be deleted.

Take

$$\Theta_2 = \{i \in \Gamma \backslash \Theta_1 | M_i(\ltimes_{j \in \mathcal{N}_i} \vec{\xi}_j) \neq \vec{\xi}_i\},$$

where M_i is the structure matrix of f_i. And Θ_2 is the node set in which there exists no arc to be deleted, but the nodes cannot reach their desired states without controllers.

Finally, the pinned node set Θ can be expressed by the union of Θ_1 and Θ_2 as

$$\Theta = \Theta_1 \cup \Theta_2 \subseteq \Gamma.$$

3.3. Design of state feedback control functions and logical couplings

In this subsection, we aim to obtain the state feedback control functions μ_i and logical couplings $\oplus_i$ of (3.2). Since pinned node set Θ consists of two disjoint parts: Θ_1 and Θ_2, we will discuss the controller design for these two types of nodes respectively. For each type of pinned node, sufficient conditions for nodes to attain their desired dynamics will be given, through which the structure matrices of μ_i and $\oplus_i$ can be derived.

3.3.1. Design of controllers for nodes in set Θ_1

We first consider the controller design for the nodes in subset Θ_1, which will be given in Theorem 3.1. Before that, a special kind of matrix called σ-permutation matrix will be introduced.

Lemma 3.2. *[36] Consider a node $i \in [1:n]$ with $\mathcal{N}_i = \{n_i^1, n_i^2, \cdots, n_i^{m_i}\}$ and $\mathcal{N}_i^d = \{d_i^1, d_i^2, \cdots, d_i^{c_i}\}$. Then we can construct a matrix W_{σ_i} associated with the variables arrangement from $\ltimes_{j\in\mathcal{N}_i^d} x_j(t) \ltimes_{j\in\mathcal{N}_i\setminus\mathcal{N}_i^d} x_j(t)$ to $\ltimes_{j\in\mathcal{N}_i} x_j(t)$, such that*

$$\ltimes_{j\in\mathcal{N}_i} x_j(t) = W_{\sigma_i} \ltimes_{j\in\mathcal{N}_i^d} x_j(t) \ltimes_{j\in\mathcal{N}_i\setminus\mathcal{N}_i^d} x_j(t). \tag{3.3}$$

In Lemma 3.2, the matrix W_{σ_i} is called σ-permutation matrix.

Theorem 3.1. *Consider logical network (3.1). For each $i \in \Theta_1$, x_i can attain its desired dynamics, if there exists controller with control function $\hat{\mu}_i$ and logical coupling $\hat{\oplus}_i$ satisfying*

$$M_{\hat{\oplus}_i} M_{\hat{\mu}_i} (I_{k^{|\mathcal{N}_i|}} \otimes M_i) R^P_{k^{|\mathcal{N}_i|}} W_{\sigma_i} = \vec{M}_i R_{[k^{|\mathcal{N}_i^d|}, k^{|\mathcal{N}_i\setminus\mathcal{N}_i^d|}]}, \tag{3.4}$$

where $M_{\hat{\oplus}_i} \in \mathcal{L}_{k\times k^2}$, $M_{\hat{\mu}_i} \in \mathcal{L}_{k\times k^{|\mathcal{N}_i|}}$, $M_i \in \mathcal{L}_{k\times k^{|\mathcal{N}_i|}}$ and $\vec{M}_i \in \mathcal{L}_{k\times k^{|\mathcal{N}_i\setminus\mathcal{N}_i^d|}}$ are the structure matrices of $\hat{\oplus}_i$, $\hat{\mu}_i$, f_i and $\vec{f}_i$, respectively.

Proof. Assume that there exists controller $\hat{u}_i$ with $\hat{\mu}_i$ and $\hat{\oplus}_i$ being its control function and logical coupling respectively, then applying $\hat{u}_i$ to x_i, the dynamics of x_i is converted into

$$x_i(t+1) = \hat{u}_i(t) \hat{\oplus}_i f_i(\{x_j(t) | j \in \mathcal{N}_i\}),$$

and the corresponding algebraic form can be expressed as

$$\begin{aligned} x_i(t+1) &= M_{\hat{\oplus}_i} \hat{u}_i(t) M_i \ltimes_{j\in\mathcal{N}_i} x_j(t) \\ &= M_{\hat{\oplus}_i} M_{\hat{\mu}_i} \ltimes_{j\in\mathcal{N}_i} x_j(t) M_i \ltimes_{j\in\mathcal{N}_i} x_j(t) \\ &= M_{\hat{\oplus}_i} M_{\hat{\mu}_i} (I_{k^{|\mathcal{N}_i|}} \otimes M_i) \ltimes_{j\in\mathcal{N}_i} x_j(t) \ltimes_{j\in\mathcal{N}_i} x_j(t) \\ &= M_{\hat{\oplus}_i} M_{\hat{\mu}_i} (I_{k^{|\mathcal{N}_i|}} \otimes M_i) R^P_{k^{|\mathcal{N}_i|}} \ltimes_{j\in\mathcal{N}_i} x_j(t). \end{aligned} \tag{3.5}$$

By substituting (3.3) into (3.5), one has

$$\begin{aligned} x_i(t+1) =& M_{\hat{\oplus}_i} M_{\hat{\mu}_i} (I_{k^{|\mathcal{N}_i|}} \otimes M_i) R^P_{k^{|\mathcal{N}_i|}} W_{\sigma_i} \ltimes_{j\in\mathcal{N}_i^d} x_j(t) \\ & \ltimes_{j\in\mathcal{N}_i\setminus\mathcal{N}_i^d} x_j(t). \end{aligned} \tag{3.6}$$

By substituting (3.4) into (3.6), one has

$$\begin{aligned} x_i(t+1) &= \vec{M}_i R_{[k^{|\mathcal{N}_i^d|}, k^{|\mathcal{N}_i\setminus\mathcal{N}_i^d|}]} \ltimes_{j\in\mathcal{N}_i^d} x_j(t) \ltimes_{j\in\mathcal{N}_i\setminus\mathcal{N}_i^d} x_j(t) \\ &= \vec{M}_i \ltimes_{j\in\mathcal{N}_i\setminus\mathcal{N}_i^d} x_j(t), \end{aligned} \tag{3.7}$$

which is the algebraic form of the desired dynamics of x_i. □

3.3.2. Design of controllers for nodes in set Θ_2

Similar to Theorem 3.1, the controller design for the nodes in subset Θ_2 will be presented in Theorem 3.2.

Theorem 3.2. *Consider logical network (3.1). For each $i \in \Theta_2$, x_i can attain its desired dynamics, if there exists controller with control function $\check{\mu}_i$ and logical coupling $\check{\oplus}_i$ satisfying*

$$M_{\check{\oplus}_i} M_{\check{\mu}_i} (I_{k^{|\mathcal{N}_i|}} \otimes M_i) R^P_{k^{|\mathcal{N}_i|}} = \vec{M}_i, \tag{3.8}$$

where $M_{\check{\oplus}_i} \in \mathcal{L}_{k\times k^2}$, $M_{\check{\mu}_i} \in \mathcal{L}_{k\times k^{|\mathcal{N}_i|}}$, $M_i \in \mathcal{L}_{k\times k^{|\mathcal{N}_i|}}$ and $\vec{M}_i \in \mathcal{L}_{k\times k^{|\mathcal{N}_i|}}$ are the structure matrices of $\check{\oplus}_i$, $\check{\mu}_i$, f_i and $\vec{f}_i$, respectively.

Proof. The algebraic form of

$$x_i(t+1) = \check{u}_i(t) \check{\oplus}_i f_i(\{x_j(t) | j \in \mathcal{N}_i\}),$$

can be expressed as

$$\begin{aligned} x_i(t+1) &= M_{\check{\oplus}_i} \check{u}_i(t) M_i \ltimes_{j\in\mathcal{N}_i} x_j(t) \\ &= M_{\check{\oplus}_i} M_{\check{\mu}_i} \ltimes_{j\in\mathcal{N}_i} x_j(t) M_i \ltimes_{j\in\mathcal{N}_i} x_j(t) \\ &= M_{\check{\oplus}_i} M_{\check{\mu}_i} (I_{k^{|\mathcal{N}_i|}} \otimes M_i) \ltimes_{j\in\mathcal{N}_i} x_j(t) \ltimes_{j\in\mathcal{N}_i} x_j(t) \\ &= M_{\check{\oplus}_i} M_{\check{\mu}_i} (I_{k^{|\mathcal{N}_i|}} \otimes M_i) R^P_{k^{|\mathcal{N}_i|}} \ltimes_{j\in\mathcal{N}_i} x_j(t). \end{aligned} \tag{3.9}$$

By substituting (3.8) to (3.9), one has

$$x_i(t+1) = \vec{M}_i \ltimes_{j\in\mathcal{N}_i} x_j(t),$$

which is the algebraic form of the desired dynamics of x_i. □

According to Theorems 3.1 and 3.2, if we can solve $M_{\oplus_i}$ and M_{μ_i} from (3.4) and (3.8), then the existence of pinning controllers can be derived naturally. The following proposition is provided to show the solvability of (3.4) and (3.8).

Proposition 3.1. *Given $P, Q \in \mathcal{L}_{k\times k^n}$, there exist logical matrices $S \in \mathcal{L}_{k\times k^2}$ and $C \in \mathcal{L}_{k\times k^n}$, such that*

$$SC(I_{k^n} \otimes P)R^P_{k^n} = Q. \tag{3.10}$$

Proof. Assume that

$$P = (p_{ij})_{k\times k^n}, Q = (q_{ij})_{k\times k^n},$$

$$S = (s_{ij})_{k\times k^2}, C = (c_{ij})_{k\times k^n},$$

where P, Q, S, C are four logical matrices, and

$$s_{kj} = 1 - \sum_{i=1}^{k-1} s_{ij}, \quad j = 1, 2, \cdots, k^2;$$

$$p_{kj} = 1 - \sum_{i=1}^{k-1} p_{ij}, \quad q_{kj} = 1 - \sum_{i=1}^{k-1} q_{ij},$$

$$c_{kj} = 1 - \sum_{i=1}^{k-1} c_{ij}, \quad j = 1, 2, \cdots, k^n.$$

Using STP, the left-hand side of (3.10) can be expressed as

$$\begin{aligned}
&SC(I_{k^n} \otimes P)R^P_{k^n}\\
=&S(C \otimes I_k)(I_{k^n} \otimes P)R^P_{k^n}\\
=&S(CI_{k^n} \otimes I_kP)R^P_{k^n}\\
=&S(C \otimes P)R^P_{k^n}\\
=&\begin{bmatrix} s_{1,1} & s_{1,2} & \cdots & s_{1,k^2}\\ s_{2,1} & s_{2,2} & \cdots & s_{2,k^2}\\ \vdots & \vdots & & \vdots\\ s_{k,1} & s_{k,2} & \cdots & s_{k,k^2}\end{bmatrix}\\
&\ltimes \begin{bmatrix} c_{1,1}p_{1,1} & \cdots & c_{1,1}p_{1,k^n} & \cdots & c_{1,k^n}p_{1,k^n}\\ \vdots & & \vdots & & \vdots\\ c_{1,1}p_{k,1} & \cdots & c_{1,1}p_{k,k^n} & \cdots & c_{1,k^n}p_{k,k^n}\\ \vdots & & \vdots & & \vdots\\ c_{k,1}p_{k,1} & \cdots & c_{k,1}p_{k,k^n} & \cdots & c_{k,k^n}p_{k,k^n}\end{bmatrix} R^P_{k^n}\\
=&\begin{bmatrix} s_{1,1} & s_{1,2} & \cdots & s_{1,k^2}\\ s_{2,1} & s_{2,2} & \cdots & s_{2,k^2}\\ \vdots & \vdots & & \vdots\\ s_{k,1} & s_{k,2} & \cdots & s_{k,k^2}\end{bmatrix}\\
&\ltimes \begin{bmatrix} c_{1,1}p_{1,1} & c_{1,2}p_{1,2} & \cdots & c_{1,k^n}p_{1,k^n}\\ \vdots & \vdots & & \vdots\\ c_{1,1}p_{k,1} & c_{1,2}p_{k,2} & \cdots & c_{1,k^n}p_{k,k^n}\\ \vdots & \vdots & & \vdots\\ c_{k,1}p_{k,1} & c_{k,2}p_{k,2} & \cdots & c_{k,k^n}p_{k,k^n}\end{bmatrix}.
\end{aligned}$$

Hence, (3.10) is equivalent to the following equations

$$\begin{cases} s_{1,1}c_{1,j}p_{1,j} + s_{1,2}c_{1,j}p_{2,j} + \cdots + s_{1,k}c_{1,j}p_{k,j} + \cdots\\ \quad + s_{1,(k-1)k+1}c_{k,j}p_{1,j} + \cdots + s_{1,k^2}c_{k,j}p_{k,j} = q_{1,j},\\ \qquad \vdots\\ s_{k-1,1}c_{1,j}p_{1,j} + s_{k-1,2}c_{1,j}p_{2,j} + \cdots + s_{k-1,k}c_{1,j}p_{k,j} + \cdots\\ \quad + s_{k-1,(k-1)k+1}c_{k,j}p_{1,j} + \cdots + s_{k-1,k^2}c_{k,j}p_{k,j} = q_{k-1,j}, \end{cases} \tag{3.11}$$

where $j = 1, 2, \cdots, k^n$.

For each $j \in [1 : k^n]$, according to different values of $p_{i,j}$ and $q_{i,j}$, we can divide them into the following several cases.

(Case 1)

If there exist $m, l \in [1 : k-1]$, such that

$$q_{m,j} = 1, p_{l,j} = 1,$$

then taking $s_{m,l} = 1, c_{1,j} = 1$, one has (3.11) holds.

(Case 2)

If for any $m, l \in [1 : k-1]$, such that

$$q_{m,j} = 0, p_{l,j} = 0,$$

then taking $s_{k,k} = 1, c_{1,j} = 1$, one has (3.11) holds.

(Case 3)

For any $l \in [1 : k-1]$, if there exists $m \in [1 : k-1]$, such that

$$q_{m,j} = 1, p_{l,j} = 0,$$

then taking $s_{m,k^2} = 1, c_{k,j} = 1$, one has (3.11) holds.

(Case 4)

If for any $m \in [1 : k-1]$, there exists $l \in [1 : k-1]$, such that

$$q_{m,j} = 0, p_{l,j} = 1,$$

then taking $s_{k,(k-1)k+l} = 1, c_{k,j} = 1$, one has (3.11) holds.

Thus, it can be concluded that for $P, Q \in \mathcal{L}_{k\times k^n}$, there exist $S \in \mathcal{L}_{k\times k^2}$, $C \in \mathcal{L}_{k\times k^n}$, such that (3.11) holds. That is, (3.10) holds. □

Remark 3.1. *However, using Proposition* 3.1*, the corresponding pinning controller can be either open-loop or closed-loop. For the open-loop case, it can be derived that a state feedback controller can also be obtained by exploring another solution C' to matrix equation (3.10).*

According to Proposition 3.1*, the open-loop controller dues to two special forms of solution C to matrix equation (3.10): the first or last row of C is $\mathbf{1}_{k^n}^T$. Without loss of generality, we assume the first row of C is $\mathbf{1}_{k^n}^T$. It comes from the fact that for each $j \in [1 : k^n]$, all of them are in the Case 1, Case 2 or both of them. For the above three cases, we give detailed steps to obtain the solution C', which are shown as follows.*

- *If for each $j \in [1 : k^n]$, it satisfies Case 1, then we choose $j_0 \in [1 : k^n]$ arbitrarily. Suppose there exist $m_0, l_0 \in [1 : k-1]$, such that $q_{m_0,j_0} = 1, p_{l_0,j_0} = 1$, then we take $s_{m_0,k+l_0} = 1, c_{2,j_0} = 1$. As for $j \in [1 : k^n]\backslash\{j_0\}$, we refer to the discussion of Case 1 in Proposition 3.1.*
- *If for each $j \in [1 : k^n]$, it satisfies Case 2, then we choose $j_0 \in [1 : k^n]$ arbitrarily. Since for any $m, l \in [1 : k-1]$, such that $q_{m,j_0} = 1, p_{l,j_0} = 1$, then we take $s_{k,2k} = 1, c_{2,j_0} = 1$. As for $j \in [1 : k^n]\backslash\{j_0\}$, we refer to the discussion of Case 2 in Proposition 3.1.*
- *If for each $j \in [1 : k^n]$, it satisfies Case 1 or Case 2. First, choose $j_0 \in [1 : k^n]$ which satisfies Case 1. Assuming that there exist $m_0, l_0 \in [1 : k-1]$, such that $q_{m_0,j_0} = 1, p_{l_0,j_0} = 1$, then we take $s_{m_0,k+l_0} = 1, c_{2,j_0} = 1$. As for $j \in [1 : k^n]\backslash\{j_0\}$, we refer to all cases proposed in Proposition* 3.1.

Thus we get the solution C', which corresponds to a state feedback controller.

Using Proposition 3.1, the solvability of (3.4) and (3.8) can be obtained immediately. Besides, Remark 3.1 guarantees the pinning feedback controllers always exist. Furthermore, according to Lemma 2.1, the logical form of μ_i and $\oplus_i$ can be obtained.

Based on the analysis above, we could derive the design of distributed pinning controllers using Theorems 3.1 and 3.2 together. Next, an algorithm is presented.

Algorithm 1 Design of Distributed Pinning Feedback Controllers

Input: Set Λ, a k-valued logical network and its associated directed graph G.

Output: State feedback control functions and logical couplings.

Set $\Gamma = \{1, 2, \cdots, s\}$, $\Gamma^c = \mathcal{N}\backslash\Gamma$, $\Theta_1 = \Theta_2 = \emptyset$.
for $i = 1, 2, \cdots, s$ **do**
  Set $\hat{\mathcal{N}}_i^d = \emptyset, \hat{\mathcal{E}}_i = \emptyset$.
  if there exist arcs from set Γ^c to node i **then**
    Collect these arcs in set $\hat{\mathcal{E}}_i$, and denote the tails of them as $\hat{\mathcal{N}}_i^d$. Set $\Theta_1 = \Theta_1 \cup \{i\}$.
  end if
end for
Denote the subgraph induced by Γ as G', where

$$G' = (\mathcal{N}', \mathcal{E}') = (\mathcal{N}', (\mathcal{E}\backslash(\underset{i\in\Gamma}{\cup}\hat{\mathcal{E}}_i)) \cap (\mathcal{N}' \times \mathcal{N}')),$$

with $\mathcal{N}' = \Gamma$. And $(\mathcal{N}' \times \mathcal{N}')$ is the subset of $\mathcal{E}$ in which the head and tail of each arc both belong to Γ.
if G' is not acyclic **then**
  Find the minimum feedback arc set via the algorithm developed in [35].
end if
for $i = 1, 2, \cdots, s$ **do**
  Set $\check{\mathcal{N}}_i^d = \emptyset, \check{\mathcal{E}}_i = \emptyset$.
  if G' is not acyclic **then**
    Based on the minimum feedback arc set, collect the arcs with i being their head in $\check{\mathcal{E}}_i$. And denote the tails of these arcs as $\check{\mathcal{N}}_i^d$.
  end if
  Set

$$\mathcal{N}_i^d = \hat{\mathcal{N}}_i^d \cup \check{\mathcal{N}}_i^d, \Theta_1 = \Theta_1 \cup \{i\}.$$

  if $i \in \Theta_1$ **then**
    Find $\vec{M}_i$, which satisfies $\vec{M}_i(\ltimes_{j\in\mathcal{N}_i\backslash\mathcal{N}_i^d}\vec{\xi}_j) = \vec{\xi}_i$.
    Design pinning controllers for node i, according to Theorem 3.1.
  end if
end for
for each $i \in \Gamma\backslash\Theta_1$ **do**
  if $M_i(\ltimes_{j\in\mathcal{N}_i}\vec{\xi}_j) \neq \vec{\xi}_i$ **then**
    $\Theta_2 = \Theta_2 \cup \{i\}$.
    if $i \in \Theta_2$ **then**
      Find $\vec{M}_i$, which satisfies $\vec{M}_i(\ltimes_{j\in\mathcal{N}_i}\vec{\xi}_j) = \vec{\xi}_i$.

Design pinning controllers for node i, according to Theorem 3.2.
end if
end if
end for
Return the state feedback control functions and logical couplings.

Theorem 3.3. *A k-valued logical network can be globally stabilized to the given set* Λ *under the designed distributed pinning controllers according to Algorithm 1.*

Proof. There exists no constrain on the states of nodes in Γ^c, so the problem of stabilizing the logical network to Λ is converted into stabilizing all nodes in Γ to their desired states $(\vec{\xi}_1, \vec{\xi}_2, \cdots, \vec{\xi}_s)$. Since the structure of the subnetwork induced by Γ is an acyclic one, it is globally stable. We will complete the proof by showing the steady state is $(\vec{\xi}_1, \vec{\xi}_2, \cdots, \vec{\xi}_s)$.

Under the designed controllers, all nodes in set Γ can be unaffected by those in Γ^c whose desired states can be arbitrary. For each $i \in \Theta_1$, the resulting dynamics can be expressed as

$$x_i(t+1) = \vec{M}_i \ltimes_{j \in \mathcal{N}_i \backslash \mathcal{N}_i^d} x_j(t). \qquad (3.12)$$

Plugging $x_j(t) = \vec{\xi}_j$, $j \in \mathcal{N}_i \backslash \mathcal{N}_i^d$ into the right-hand side of (3.12), and combining with the selection of $\vec{M}_i$, it can be concluded that $x_i(t+1) = \vec{\xi}_i$. For each $i \in \Theta_2$, the proof is similar to the nodes in Θ_1, so it is omitted. As for $i \in \Gamma \backslash (\Theta_1 \cup \Theta_2)$, $M_i(\ltimes_{j \in \mathcal{N}_i} \vec{\xi}_j) = \vec{\xi}_i$ holds. □

Remark 3.2. *Consider the time complexity of the Algorithm 1. Checking the reachability from each node in set* Γ^c *to each node in* Γ *can be realized in time* $O(n^2)$*. Besides, to obtain an acyclic graph, the minimum feedback arc set which needs to be deleted can be found in time* ps^2 *using the algorithm in [35], where* $p = |\mathcal{E}'|$, $s = |\mathcal{N}'| = |\Gamma|$*. The control functions and logical coupling operators can be calculated in time* sk^{ε}*, where* ε *is the maximum in-degree of the nodes to be controlled. The whole time complexity is* $O(n^2 + ps^3 + sk^{\varepsilon})$*.*

4. Illustrative examples

In this section, two examples are presented to demonstrate the validity and advantage of the obtained results.

Example 4.1. *Consider the following 3-valued logical system*

$$\begin{cases} x_1(t+1) = x_1(t) \leftrightarrow x_3(t), \\ x_2(t+1) = x_2(t) \vee x_3(t), \\ x_3(t+1) = \neg x_2(t). \end{cases} \qquad (4.1)$$

It is easy to get the algebraic form of (4.1) as follows

$$\begin{cases} x_1(t+1) = M_1 x_1(t) x_3(t), \\ x_2(t+1) = M_2 x_2(t) x_3(t), \\ x_3(t+1) = M_3 x_2(t), \end{cases} \qquad (4.2)$$

where $M_1 = \delta_3[1,2,3,2,2,2,3,2,1]$*,* $M_2 = \delta_3[1,1,1,1,2,1,1,2,3]$*,* $M_3 = \delta_3[3,2,1]$*.*

In this example, we only take care of the state of x_1*, and would like to globally stabilize its state to* δ_3^1*. It amounts to study the global* Λ*-stabilization of 3-valued logical network* (4.1) *with*

$$\Lambda = \{x_1 = \delta_3^1, x_2, x_3\}.$$

It is obvious that system (4.1) is not globally Λ*-stable.*

First, it can be easily derived that $\Theta = \Theta_1 = \{1\}$*,* $\mathcal{N}_1^d = \{1,3\}$*,* $W_{\sigma_1} = I$*.*

Then, assume that

$$M_{\oplus_1} = \begin{pmatrix} s_{1,1} & s_{1,2} & \cdots & s_{1,9} \\ s_{2,1} & s_{2,2} & \cdots & s_{2,9} \\ s_{3,1} & s_{3,2} & \cdots & s_{3,9} \end{pmatrix},$$

$$M_{\mu_1} = \begin{pmatrix} t_{1,1} & t_{1,2} & \cdots & t_{1,9} \\ t_{2,1} & t_{2,2} & \cdots & t_{2,9} \\ t_{3,1} & t_{3,2} & \cdots & t_{3,9} \end{pmatrix}.$$

By solving

$$M_{\oplus_1} M_{\mu_1} (I_{3^2} \otimes M_1) R_{3^2}^{P} W_{\sigma_1} = \vec{M}_1 R_{[3^2,3^0]},$$

with $\vec{M}_1 = [1,0]^T$*, we can find one feasible solution*

$$M_{\oplus_1} = \delta_3[1,1,1,3,3,3,1,1,1],$$

$$M_{\mu_1} = \delta_3[1,1,3,1,1,1,3,1,1].$$

At last, using Lemma 2.1, we have

$$\oplus_1 = (\triangleright_3^1(x_1) \wedge \phi_1(x_3)) \vee (\triangleright_3^2(x_1) \wedge \phi_2(x_3)) \vee (\triangleright_3^3(x_1) \wedge \phi_3(x_3)),$$

$$\mu_1 = (\triangleright_3^1(x_1) \wedge \phi_1'(x_3)) \vee (\triangleright_3^2(x_1) \wedge \phi_2'(x_3)) \vee (\triangleright_3^3(x_1) \wedge \phi_3'(x_3)),$$

where $M_{\oplus_1}^1 = \delta_3[1,1,1], M_{\oplus_1}^2 = \delta_3[3,3,3], M_{\oplus_1}^3 = \delta_3[1,1,1], M_{\mu_1}^1 = \delta_3[1,1,3], M_{\mu_1}^2 = \delta_3[1,1,1], M_{\mu_1}^3 = \delta_3[3,1,1]$ *are structure matrices of* ϕ_1, ϕ_2, ϕ_3, ϕ_1', ϕ_2', ϕ_3' *respectively. Furthermore, it can be briefly expressed as*

$$\oplus_1 = \triangleright_3^1(x_1) \vee \triangleright_3^3(x_1),$$

$$\mu_1 = (\triangleright_3^1(x_1) \wedge \phi_1'(x_3)) \vee \triangleright_3^2(x_1) \vee (\triangleright_3^3(x_1) \wedge \phi_3'(x_3)).$$

Example 4.2. *Consider a reduced network in the T-LGL survival signaling network [37], which can be simulated by the following BN:*

$$\begin{cases} x_1(t+1) = \neg(x_4(t) \vee x_6(t)), \\ x_2(t+1) = \neg(x_5(t) \vee x_6(t)), \\ x_3(t+1) = \neg(x_1(t) \vee x_6(t)). \\ x_4(t+1) = x_3(t) \vee \neg(x_1(t) \vee x_6(t)), \\ x_5(t+1) = (x_4(t) \vee (x_3(t) \wedge \neg x_2(t))) \wedge \neg x_6(t), \\ x_6(t+1) = x_5(t) \vee x_6(t), \end{cases} \tag{4.3}$$

where $x_1(t)$, $x_2(t)$, $x_3(t)$, $x_4(t)$, $x_5(t)$, *and* $x_1(t)$ *are state nodes that stand for the* S1P, FLIP, Fas, Ceramide, DISC, *and* Apoptosis, *respectively.*

In this example, we focus only on the states of S1P, Ceramide, *and* Apoptosis. *We aim to globally stabilize their states to* δ_2^1, δ_2^1, *and* δ_2^2, *respectively. In fact, it is equivalent to the global* Λ*-stabilization of BN (4.3) with*

$$\Lambda = \{x_1 = \delta_2^1, x_2, x_3, x_4 = \delta_2^1, x_5, x_6 = \delta_2^2\}.$$

By simple calculations, we can obtain that BN (4.3) is not globally Λ*-stable. Then we consider how to design the distributed pinning controller to achieve global set stabilization.*

According to the network graph of BN (4.3), we can easily derive that $\mathcal{N}_1^d = \emptyset$, $\mathcal{N}_4^d = \{1,3\}$, $\mathcal{N}_6^d = \{5,6\}$. *Since without any external control inputs, node* x_1 *cannot reach its desired state, the pinned node set is* $\Theta = \{1,4,6\}$. *Set* Θ *consists of two disjoint parts* Θ_1 *and* Θ_2, *where* $\Theta_1 = \{4,6\}$, $\Theta_2 = \{1\}$. *Using the proposed method in Section 3.3, we can finally design the distributed pinning controllers* $\hat{\mu}_4$, $\hat{\mu}_6$, *and* $\check{\mu}_1$ *as follows, which are imposed on nodes* x_4, x_6, *and* x_1, *respectively:*

$$\begin{cases} x_1(t+1) = \check{\mu}_1 \leftrightarrow \neg(x_4(t) \vee x_6(t)), \\ x_2(t+1) = \neg(x_5(t) \vee x_6(t)), \\ x_3(t+1) = \neg(x_1(t) \vee x_6(t)). \\ x_4(t+1) = \hat{\mu}_4 \leftrightarrow (x_3(t) \vee \neg(x_1(t) \vee x_6(t))), \\ x_5(t+1) = (x_4(t) \vee (x_3(t) \wedge \neg x_2(t))) \wedge \neg x_6(t), \\ x_6(t+1) = \hat{\mu}_6 \wedge (x_5(t) \vee x_6(t)), \end{cases} \tag{4.4}$$

where

$$\check{\mu}_1 = \neg x_4 \wedge (\neg x_6),$$

$$\hat{\mu}_4 = (x_1 \wedge x_3 \wedge x_6) \vee (x_1 \wedge \neg x_3 \wedge x_6) \vee (\neg x_1 \wedge x_3 \wedge x_6) \vee (\neg x_1 \wedge \neg x_3),$$

$$\hat{\mu}_6 = \neg x_5 \wedge \neg x_6.$$

Remark 4.1. *From the above two examples, it is apparent that our method is superior to the transition matrix-based pinning controller. In the first example, if we adopt the transition matrix-based pinning controller proposed in [26], we need to solve the* (3×27)*-dimensional matrix, and the obtained control function involves all state variables. However, using our method, only* (3×9)*-dimensional matrices are involved, and the corresponding control function is only related to state variables of* f_1. *In the second example, the pinned node set and the maximum in-degree of the pinned nodes are {1,4,6} and 3, respectively. We only need to solve the* (2×8)*-dimensional matrix, whereas the transition matrix-based pinning controller approach requires matrices of sizes* (2×128).

5. Conclusions

Distributed pinning controllers designed for set stabilization of k-valued LCNs were considered in this paper. First, according to the coupling correlations among nodes, controller design for two types of pinned nodes was discussed respectively. Based on this, an algorithm was provided to devise the pinning feedback controllers. The proposed distributed pinning technique ensured that the designed controllers only relied on the in-neighbors information of pinned nodes rather than the global information. Furthermore, compared with the transition matrix-based pinning approach, the computational complexity of the proposed method was reduced to $O(n^2 + ps^3 + sk^{\varepsilon})$.

However, there still exist interesting questions to be solved, such as distributed optimal control of logical networks and its applications.

Acknowledgment

This work was supported by National Natural Science Foundation of China under Grant 62273201, 62103232, the Research Fund for the Taishan Scholar Project of Shandong Province of China (tstp20221103), Natural Science Fund of Shandong Province under grant ZR202102230325, and China Postdoctoral Science Foundation 2020TQ0184.

Conflict of interest

The authors declared that they have no conflicts of interest to this work.

References

1. S. A. Kauffman, Metabolic stability and epigenesis in randomly constructed genetic nets, *J. Theor. Biol.,* **22** (1969), 437–467. https://doi.org/10.1016/0022-5193(69)90015-0

2. Z. Liu, Y. Wang, H. Li, New approach to derivative calculation of multi-valued logical functions with application to fault detection of digital circuits, *IET Control Theory Appl.,* **8** (2014), 554–560. https://doi.org/10.1049/iet-cta.2013.0104

3. C. M. Files, M. A. Perkowski, Multi-valued functional decomposition as a machine learning method, *Proceedings of the 28th International Symposium on Multiple-Valued Logic,* (1998), 173–178. https://doi.org/10.1109/ISMVL.1998.679331

4. D. Cheng, Semi-tensor product of matrices and its application to Morgan's problem, *Sci. China Ser. Inf. Sci.,* **44** (2001), 195–212. https://doi.org/10.1007/BF02714570

5. D. Cheng, H. Qi, A linear representation of dynamics of Boolean networks, *IEEE Trans. Autom. Control,* **55** (2010), 2251–2258. https://doi.org/10.1109/TAC.2010.2043294

6. X. Zhao, S. Fu, Trajectory tracking approach to logical (control) networks, *AIMS Mathematics,* **7** (2022), 9668–9682. https://doi.org/10.3934/math.2022538

7. Q. Zhang, J. Feng, P. Zhao, Controllability of Markovian jump Boolean control networks: A graphical approach, *Neurocomputing,* **498** (2022), 89–97. https://doi.org/10.1016/j.neucom.2022.04.119

8. F. Li, Y. Tang, Pinning controllability for a Boolean network with arbitrary disturbance inputs, *IEEE Trans. Cybern.,* **51** (2021), 3338–3347. https://doi.org/10.1109/TCYB.2019.2930734

9. Y. Zhao, Y. Liu, Output controllability and observability of mix-valued logic control networks, *Mathematical Modelling and Control,* **1** (2021), 145–156. https://doi.org/10.3934/mmc.2021013

10. 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 Trans. Autom. Control,* **67** (2022), 3472–3483. https://doi.org/10.1109/TAC.2021.3105319

11. H. Li, X. Ding, A control Lyapunov function approach to feedback stabilization of logical control networks, *SIAM J. Control Optim.,* **57** (2019), 810–831. https://doi.org/10.1137/18M1170443

12. Y. Zhang, J. Zhong, W. Xiong, J. Cao, Stabilization and oscillations design for a family of cyclic Boolean networks via nodes connection, *Neurocomputing,* **369** (2019), 61–68. https://doi.org/10.1016/j.neucom.2019.08.062

13. L. Lin, J. Zhong, S. Zhu, J. Lu, Sampled-data general partial synchronization of Boolean control networks, *J. Franklin Inst.,* **359** (2022), 1–11. https://doi.org/10.1016/j.jfranklin.2020.08.047

14. Y. Li, H. Li, P. Duan, Synchronization of switched logical control networks via event-triggered control, *J. Franklin Inst.,* **355** (2018), 5203–5216. https://doi.org/10.1016/j.jfranklin.2018.04.028

15. J. Feng, Y. Li, S. Fu, H. Lyu, New method for disturbance decoupling of Boolean networks, *IEEE Trans. Autom. Control,* **67** (2022), 4794–4800. https://doi.org/10.1109/TAC.2022.3161609

16. Y. Li, J. Zhu, Necessary and sufficient vertex partition conditions for input-output decoupling of Boolean control networks, *Automatica,* **137** (2022), 110097. https://doi.org/10.1016/j.automatica.2021.110097

17. Y. Wang, H. Li, Output trackability of Boolean control networks via ledley antecedence solution, *IEEE Trans. Circuits Syst. II, Exp. Briefs,* **69** (2022), 1183–1187. https://doi.org/10.1109/TCSII.2021.3095487

18. Y. Guo, P. Wang, W. Gui, C. Yang, Set stability and set stabilization of Boolean control networks based on invariant subsets, *Automatica,* **61** (2015), 106–112. https://doi.org/10.1016/j.automatica.2015.08.006

19. Y. Li, H. Li, W. Sun, Event-triggered control for robust set stabilization of logical control networks, *Automatica,* **95** (2018), 556–560. https://doi.org/10.1016/j.automatica.2018.06.030

20. H. Li, X. Ding, Finite-time time-variant feedback stabilization of logical control networks with Markov jump disturbances, *IEEE Trans. Circuits Syst. II, Exp. Briefs,* **67** (2020), 2079–2083. https://doi.org/10.1109/TCSII.2019.2949558

21. Y. Ding, Y. Guo, Y. Xie, C. Yang, W. Gui, Time-optimal state feedback stabilization of switched Boolean control networks, *Neurocomputing,* **237** (2017), 265–271. https://doi.org/10.1016/j.neucom.2016.12.044

22. J. Lu, J. Zhong, C. Huang, J. Cao, On pinning controllability of Boolean control networks, *IEEE Trans. Autom. Control,* **61** (2016), 1658–1663. https://doi.org/10.1109/TAC.2015.2478123

23. Y. Li, J. Feng, X. Li, S. Xu, Pinning controller design for set reachability of state-dependent impulsive Boolean networks, *IEEE Trans. Neural Netw. Learn. Syst.*, (2022), 35536802. https://doi.org/10.1109/TNNLS.2022.3171576

24. H. Li, P. Song, Q. Yang, Pinning control design for robust output tracking of k-valued logical networks, *J. Franklin Inst.,* **354** (2017), 3039–3053. https://doi.org/10.1016/j.jfranklin.2017.02.009

25. Y. Liu, B. Li, J. Lu, J. Cao, Pinning control for the disturbance decoupling problem of Boolean networks, *IEEE Trans. Autom. Control,* **62** (2017), 6595–6601. https://doi.org/10.1109/TAC.2017.2715181

26. F. Li, H. Li, L. Xie, Q. Zhou, On stabilization and set stabilization of multivalued logical systems, *Automatica,* **80** (2017), 41–47. https://doi.org/10.1016/j.automatica.2017.01.032

27. S. Zhu, J. Lu, L. Sun, J. Cao, Distributed pinning set stabilization of large-scale Boolean networks, *IEEE Trans. Autom. Control*, https://doi.org/10.1109/TAC.2022.3169178

28. D. Cheng, H. Qi, Z. Li, *Analysis and Control of Boolean Networks: A Semi-tensor Product Approach*, London: Springer, 2011.

29. D. Cheng, Disturbance decoupling of Boolean control networks, *IEEE Trans. Autom. Control.* **56** (2011), 2–10. https://doi.org/10.1109/TAC.2010.2050161

30. Z. Li, D. Cheng, Algebraic approach to dynamics of multivalued networks, *Int. J. Bifurcat. Chaos,* **20** (2010), 561–582. https://doi.org/10.1142/S0218127410025892

31. D. Cheng, H. Qi, Y. Zhao, *An Introduction to Semi-tensor Product of Matrices and Its Applications*, Singapore: World Scientific, 2012.

32. D. Cheng, Z. Liu, H. Qi, Completeness and normal form of multi-valued logical functions, *J. Franklin Inst.,* **357** (2020), 9871–9884. https://doi.org/10.1016/j.jfranklin.2020.06.026

33. A. Veliz-Cuba, D. Murrugarra, R. Laubenbacher, Structure and dynamics of acyclic networks, *Discret. Event Dyn. Syst.-Theory Appl.,* **24** (2014), 647–658. https://doi.org/10.1007/s10626-013-0174-2

34. J. Bang-Jensen, G. Gutin, *Digraphs: Theory, Algorithms and Applications*, New York: Springer, 2008.

35. G. Even, J. Naor, B. Schieber, M. Sudan, Approximating minimum feedback sets and multicuts in directed graphs, *Algorithmica,* **20** (1998), 151–174. https://doi.org/10.1007/PL00009191

36. Y. Wang, T. Liu, D. Cheng, Some notes on semi-tensor product of matrices and swap matrix, *Journal of Systems Science and Mathematical Sciences,* **36** (2016), 1367–1375. https://doi.org/10.12341/jssms12892

37. C. Campbell, R. Albert, Stabilization of perturbed Boolean network attractors through compensatory interactions, *BMC Syst. Biol.,* **8** (2014), 53. https://org/10.1186/1752-0509-8-53