Mathematics

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

AIMS Mathematics, 7(6): 9668–9682.
DOI: 10.3934/math.2022538
Received: 23 December 2021
Revised: 23 February 2022
Accepted: 01 March 2022
Published: 15 March 2022

Research article

Trajectory tracking approach to logical (control) networks

Xiaoyu Zhao[1,2,*] **and Shihua Fu**[3,4]

[1] School of Mathematics, Shandong University, Jinan, Shandong 250100, China

[2] Research Center for Mathematics and Interdisciplinary Sciences, Shandong University, Qingdao 266237, China

[3] Research Center of Semi-tensor Product of Matrices: Theory and Applications, Liaocheng University, Liaocheng 252000, China

[4] School of Mathematical Science, Liaocheng University, Liaocheng 252000, China

* **Correspondence:** Email: ustcxyz@hotmail.com.

Abstract: Vector form expression of logical (control) networks is presented. From this aspect, the trajectory table is proposed to investigate Boolean networks. Based on it, the topology structure, controllability and observability of logical (control) networks are analyzed. Compared to the method of logical matrix, vector form expression called structure vector method decreases the computational complex. Numerical examples show that the complexity of the structure vector method is greatly reduced.

Keywords: semi-tensor product (STP) of matrices; vector form expression; attractors; controllability matrix
Mathematics Subject Classification: 93-XX

1. Introduction

Since Kauffman proposed the Boolean network to formulate a genetic regulatory network in 1969 [1], there has been considerable researches going on in this field. The most important problem is the topological structure of a Boolean network, which involves the fixed points, cycles, and their basins of attraction [2].

Since 2009, the semi-tensor product (STP) of matrices has been introduced into the study of Boolean networks [3, 4]. STP provides a convenient way to formulate Boolean networks [5]. It, therefore, promotes the development of the theory of Boolean network. The development has been merged mainly in two directions. One is the control of Boolean networks. Various of control problems, such as controllability [6–8], observability [9–11], stabilization [12–14], optimal control [15, 16], tracking [17, 18], decoupling [19, 20] etc. Another direction is that the values of states in the networks have been generated from Boolean case, which allows only $\{0, 1\}$, to k-valued logical networks [21], mix-valued logical networks [22], or finite fields [23]. Based on the applications of STP to finite valued systems, many valuable results have been obtained, such as game theory [24], quantum networks [25], shift registers [26, 27], finite automaton [28, 29], fuzzy systems [30], and so on. In addition, STP has important applications in the analysis of finite base spaces, such as numerical/non-numerical algebra [31], finite Boolean-type algebras [32], etc. Besides, researchers have focused on reinforcement learning-based techniques to study Boolean networks. Literature [33–35] uses a model-free method based on reinforcement learning to study control issues such as stability and output tracking problems, which is a good alternative to the STP-based methods when the actual system is unknown.

Although the previous research about Boolean networks has tended to be perfect, a severe obstacle in applying STP to finite-valued networks is the computational complexity. Using previous algebraic methods to study finite-value networks tends to bring a high computational cost. In order to reduce the computational complexity, vector form expression of logical (control) networks is proposed in this paper, which is called the structure vector method (SVM). Then, through a one-step trajectory table, some classical problems in control theory, such as topology structure, controllability, and observability, are analyzed again to demonstrate the effectiveness of this method in reducing computational complexity. In fact, $2^n \times 2^n$ dimensional matrices should be computed in the algebraic method used in the existing literature, while only 2^n dimensional vectors are used in the structure vector method. Therefore, compared with the algebraic method, the vector form expression can greatly reduce the computational complexity.

The rest of this paper is organized as follows: Section 2 presents some relevant previous results on STP and Boolean networks. Section 3 is devoted to analyzing the attractor and its basin of logical networks. Sections 4 and 5 discuss the controllability and observability of logical networks with some numerical examples. Section 6 summarizes our results in this paper.

Before ending this section, we give a list of notations:

(1) $\mathcal{M}_{m\times n}$: The set of $m \times n$ dimensional real matrices.
(2) $\mathbb{R}^n$: The set of n-dimensional real vectors.
(3) $\mathrm{Col}(A)$: The set of columns (rows) of A; $\mathrm{Col}_i(A)$ ($\mathrm{Row}_i(A)$): The i-th column (row) of A.
(4) $\mathcal{D}_k$: $\mathcal{D}_k = \{1, 2, \cdots, k-1, 0\}$.
(5) δ_k^i: The i-th column of identity matrix I_k.
(6) Δ_k: $\Delta_k = \mathrm{Col}(I_k) = \left\{\delta_k^i \mid i = 1, \cdots, k\right\}$
(7) $A \in \mathcal{M}_{m\times n}$ is called a logical matrix, if $\mathrm{Col}(A) \subset \Delta_m$.
(8) $\mathcal{L}_{m\times n}$: The set of $m \times n$ logical matrices.
(9) Assume $L \in \mathcal{L}_{m\times n}$, then $L = [\delta_m^{i_1}, \delta_m^{i_2}, \cdots, \delta_m^{i_n}] := \delta_m[i_1, i_2, \cdots, i_n]$.
(10) M_{ij}: The (i, j)-th element of matrix M, and $M = [M_{ij}]$ denotes a matrix whose (i, j)-th element is M_{ij}.
(11) A^T denotes the transpose of matrix A.

2. Preliminaries

2.1. Semi-tensor product of matrices

Since STP is a fundamental tool in our construction, this section will give a brief survey for STP. We refer to [36] for more details.

Definition 2.1. *Let $A \in \mathcal{M}_{m\times n}$ and $B \in \mathcal{M}_{p\times q}$, $t = \mathrm{lcm}(n, p)$ be the least common multiple of n and p. Then the STP of A and B, denoted by $A \ltimes B$, is defined as*

$$A \ltimes B := (A \otimes I_{t/n})\left(B \otimes I_{t/p}\right), \tag{1}$$

where $\otimes$ is Kronecker product.

It is easy to see that STP is a generalization of conventional matrix product. That is, when $n = p$, it degenerates to the conventional matrix product, i.e., $A \ltimes B = AB$. Because of this, we omit the symbol $\ltimes$ in most cases.

One of the most important advantages of STP is that it keeps most properties of conventional matrix product available, including association, distribution, etc. In the following we introduce some additional properties of STP, which will be used in the sequel.

Define a swap matrix $W_{[m,n]} \in \mathcal{M}_{mn\times mn}$ as follows:

$$W_{[m,n]} := \left[I_n \otimes \delta_m^1, I_n \otimes \delta_m^2, \cdots, I_n \otimes \delta_m^m, \right] \in \mathcal{L}_{mn\times mn}. \tag{2}$$

Proposition 2.2. *Let $x \in \mathbb{R}^m$ and $y \in \mathbb{R}^n$ be two column vectors. Then*

$$W_{[m,n]}x \ltimes y = y \ltimes x. \tag{3}$$

The following proposition shows how to “swap” a vector with a matrix:

Proposition 2.3. *Let $x \in \mathbb{R}^t$ be a column vector, and A be an arbitrary matrix. Then*

$$x \ltimes A = (I_t \otimes A) \ltimes x. \tag{4}$$

Definition 2.4. *Let $A \in \mathcal{M}_{p\times n}$ and $B \in \mathcal{M}_{q\times n}$. The Kratri-Rao product of A and B, denoted by $A * B \in \mathcal{M}_{pq\times n}$, is defined by*

$$\mathrm{Col}_i(A * B) = \mathrm{Col}_i(A) \ltimes \mathrm{Col}_i(B), \quad i \in [1, n]. \tag{5}$$

Define a matrix $PR_k \in \mathcal{L}_{k^2\times k}$, called the power reducing matrix, as follows:

$$PR_k := diag\left(\delta_k^1, \delta_k^2, \cdots, \delta_k^k\right). \tag{6}$$

The function of the power reducing matrix is the following.

Proposition 2.5. *Assume $x \in \Delta_k$, then*

$$x^2 = PR_k x. \tag{7}$$

Throughout this paper the default matrix product is assumed to be STP, and the symbol $\ltimes$ is omitted if there is no possible confusion.

2.2. Matrix expression of logical networks

Assume $X \in \mathcal{D}_k$. The vector form expression of X, denoted by $x = \vec{X}$ is defined as

$$x = \begin{cases} \delta_k^X, & 1 \le X \le k-1 \\ \delta_k^k, & X = 0. \end{cases}$$

It is worth noting that since the data values used in this paper are all integers, the intervals in the following refer to integer values unless otherwise specified. For example, $[1, n]$ refers to integers from 1 to n.

Definition 2.6. *Let $X_i \in \mathcal{D}_{k_i}$, $i \in [1, n]$.*

(i) A mapping $F : \prod_{i=1}^n \mathcal{D}_{k_i} \to \mathcal{D}_{k_0}$ is called a mix-valued logical function, expressed by

$$Y = F(X_1, X_2, \cdots, X_n). \tag{8}$$

(ii) If $k_i = k$, $i \in [1, n]$, F is called a k-valued logical function.

(iii) If $k_i = 2$, $i \in [1, n]$, F is called a Boolean function.

Proposition 2.7. *[5] Let $F : \prod_{i=1}^n \mathcal{D}_{k_i} \to \mathcal{D}_{k_0}$ be a mix-valued logical function. Then there exists a unique logical matrix M_F, called the structure matrix of F, such that vector form (8) can be expressed as*

$$y = M_F \ltimes_{i=1}^n x_i := M_F x, \tag{9}$$

where y and x_i, $i \in [1, n]$ are vector form of Y and X_i, $i \in [1, n]$ respectively.

Definition 2.8. *Let $X_i(t) \in \mathcal{D}_{k_i}$, $i \in [1, n]$, $t = 0, 1, \cdots$ be a set of time-varying logical variables, $G = (N, E)$ be a directed graph with $N = \{1, 2, \cdots, n\}$ and $X_i(t)$ as the logical value of node i at time t, $i \in [1, n]$.*

(i)

$$\begin{cases} X_1(t+1) = F_1(X_j(t) \mid j \in \mathcal{N}_1) \\ X_2(t+1) = F_2(X_j(t) \mid j \in \mathcal{N}_2) \\ \cdots \\ X_n(t+1) = F_n(X_j(t) \mid j \in \mathcal{N}_n) \end{cases} \tag{10}$$

is called a mix-valued logical network, where F_i, $i \in [1, n]$ are mix-valued logical functions, $\mathcal{N}_i$ $i \in [1, n]$ are neighborhood of node i.

(ii) If there are controls $U_j \in \mathcal{D}_{\alpha_j}$, $j \in [1, m]$ and outputs $Y_\ell \in \mathcal{D}_{\beta_\ell}$, $\ell \in [1, p]$, such that

$$\begin{cases} X_1(t+1) = F_1(X_j(t), U_s(t) \mid j, s \in \mathcal{N}_1) \\ X_2(t+1) = F_2(X_j(t), U_s(t) \mid j, s \in \mathcal{N}_2) \\ \cdots \\ X_n(t+1) = F_n(X_j(t), U_s(t) \mid j, s \in \mathcal{N}_n) \end{cases} \tag{11}$$
$$Y_\ell(t) = H_\ell(X_1(t), \cdots, X_n(t)), \quad \ell \in [1, p],$$

then (11) is called a mix-valued logical control network.

(iii) If $k_i = \alpha_s = \beta_\ell = k$, $i \in [1,n]$, $s \in [1,m]$ and $\ell \in [1,p]$, then (10) is called a k-valued logical network, and (11) is called a k-valued logical control network.

(iv) If $k_i = \alpha_s = \beta_\ell = 2$, $i \in [1,n]$, $s \in [1,m]$ and $\ell \in [1,p]$, then (10) is called a Boolean network, and (11) is called a Boolean control network.

Set $\kappa = \prod_{i=1}^{n} k_i$, $\alpha = \prod_{s=1}^{m} \alpha_s$, $\beta = \prod_{\ell=1}^{p} \beta_\ell$. Using Proposition 2.7 to network (10) and control network (11) respectively yields the following result.

Proposition 2.9. *[5]*

(i) Vector form (10) can be expressed as

$$\begin{cases} x_1(t+1) = M_1 x(t) \\ x_2(t+1) = M_2 x(t) \\ \cdots \\ x_n(t+1) = M_n x(t), \end{cases} \tag{12}$$

where $x(t) = \ltimes_{i=1}^{n} x_n(t)$, $M_i \in \mathcal{L}_{k_i \times \kappa}$, $i \in [1,n]$.

(ii) In vector form (11) can be expressed as

$$\begin{cases} x_1(t+1) = L_1 u(t) x(t) \\ x_2(t+1) = L_2 u(t) x(t) \\ \cdots \\ x_n(t+1) = L_n u(t) x(t), \end{cases} \\ y_\ell(t) = E_\ell x(t), \quad \ell \in [1,p]. \tag{13}$$

(12) and (13) are called the component-wise algebraic state space representation (ASSR) of (10) and (11) respectively.

Putting all the components together yields the following overall ASSR.

Proposition 2.10. *[5]*

(i) The component-wise ASSR (12) can be expressed into (overall) ASSR as

$$x(t+1) = M x(t), \tag{14}$$

where $M = M_1 * M_2 * \cdots * M_n \in \mathcal{L}_{\kappa \times \kappa}$.

(ii) The component-wise ASSR (13) can be expressed into (overall) ASSR as

$$\begin{aligned} & x(t+1) = L u(t) x(t) \\ & y(t) = E x(t), \end{aligned} \tag{15}$$

where $L = L_1 * L_2 * \cdots * L_n \in \mathcal{L}_{\kappa \times \kappa\alpha}$ and $E = E_1 * E_2 * \cdots * E_p \in \mathcal{L}_{\beta \times \kappa}$.

3. Attractor and basin of attraction

Consider the ASSR of a logical network (14). Since $M \in \mathcal{L}_{2^n \times 2^n}$, using short-hand expression, one sees easily that a mix-valued logical network can be expressed as a κ-dimensional vector, which is called the structure vector of network (10).

Next, we consider how to use this structure vector to calculate the attractors of (10). We give an example to depict this.

Example 3.1. *Assume $X_1(t), X_3(t) \in \mathcal{D}_2$, $X_2(t) \in \mathcal{D}_3$, and the network dynamics is as follows.*

$$\begin{cases} X_1(t+1) = X_1(t) \vee X_3(t) \\ X_2(t+1) = X_2(t) \triangledown X_3(t) \\ X_3(t+1) = \neg X_1(t), \end{cases} \tag{16}$$

where $\triangledown : \mathcal{D}_3 \times \mathcal{D}_2 \to \mathcal{D}_3$, $\vee : \mathcal{D}_2 \times \mathcal{D}_2 \to \mathcal{D}_2$, $\neg : \mathcal{D}_2 \to \mathcal{D}_2$ with their structure matrices as

$$\begin{aligned} &M_{\triangledown} = \delta_3[1, 3, 2, 2, 3, 1], \\ &M_{\vee} = \delta_2[1, 1, 1, 2], \\ &M_{\neg} = \delta_2[2, 1]. \end{aligned}$$

It is easy to calculate the structure matrices of M_i $i = 1, 2, 3$ as follows:

$$\begin{aligned} M_1 &= M_{\vee}[I_2 \otimes \mathbf{1}_3^T \otimes I_2] \\ &= \delta_2[1, 1, 1, 1, 1, 1, 1, 2, 1, 2, 1, 2]. \end{aligned}$$

$$\begin{aligned} M_2 &= M_{\triangledown}[\mathbf{1}_2^T \otimes I_6] \\ &= \delta_3[1, 3, 2, 2, 3, 1, 1, 3, 2, 2, 3, 1]. \end{aligned}$$

$$\begin{aligned} M_3 &= M_{\neg}[I_2 \otimes \mathbf{1}_6^T] \\ &= \delta_2[2, 2, 2, 2, 2, 2, 1, 1, 1, 1, 1, 1]. \end{aligned}$$

It follows that

$$\begin{aligned} M &= M_1 * M_2 * M_3 \\ &= \delta_{12}[2, 6, 4, 4, 6, 2, 1, 11, 3, 9, 5, 7]. \end{aligned} \tag{17}$$

Hence, the ASSR of (16) is

$$x(t+1) = Mx(t), \tag{18}$$

where M is shown in (17). Hence, the network is uniquely determined by the vector

$$V_M = [2, 6, 4, 4, 6, 2, 1, 11, 3, 9, 5, 7],$$

which is called the structure vector of network (16).

The physical meaning of V_M is: It determines a mapping $\pi_M : \Delta_{2^n} \to \Delta_{2^n}$. Say, the first element in V_M is 2 meaning that $\pi_M(\delta_{12}^1) = \delta_{12}^2$, which can also be briefly denoted by $\pi_M(1) = 2$. Hence, we have the mapping as shown in Table 1.

Table 1. π_M of (16).

x	1	2	3	4	5	6	7	8	9	10	11	12
$\pi_M(x)$	2	6	4	4	6	2	1	11	3	9	5	7

δ_{12}^1, briefly 1, is maps to 2, denoted by $1 \to 2$, and

$$2 \to 6, \quad 3 \to 4, \cdots, 12 \to 7.$$

Using this mapping, it is easy to calculate the attractors of network (16) by the following Table 1.

Using this mapping, we can start from any state to calculate whole trajectory as described in Table 2. Note that for each trajectory, if an old state appears again, we use a $*$ to stop the trajectory. That means the trajectory start to go on the second round of cycle.

Table 2. Trajectory table of (16).

x_0	1	2	3	4	5	6	7	8	9	10	11	12
$\pi_M(x)$	2	6	4	4	6	2	1	11	3	9	5	7
$\pi_M^2(x)$	6	2	4	$*$	2	6	2	5	4	3	6	2
$\pi_M^3(x)$	2	$*$	$*$		6	$*$	6	6	4	4	2	6
$\pi_M^4(x)$	$*$				$*$		2	2	$*$	4	6	2
$\pi_M^5(x)$							$*$	6		$*$	$*$	$*$
$\pi_M^6(x)$								$*$				

From Table 2, one sees easily that there is a fixed point, which is $P = \delta_{12}^4 \sim (1,2,2)$, and a cycle, which is $C : \delta_{12}^2 \leftrightarrow \delta_{12}^6$ or $C : (1,1,2) \leftrightarrow (1,3,2)$. The basin of attraction of each attractor is also clear. That is,

$$\begin{aligned} B_P &= \left\{\delta_{12}^3 \sim (1,2,1), \delta_{12}^9 \sim (2,2,1), \delta_{12}^{10} \sim (2,2,2)\right\}; \\ B_C &= \left\{\delta_{12}^1 \sim (1,1,1), \delta_{12}^5 \sim (1,3,1), \delta_{12}^7 \sim (2,1,1),\right. \\ &\quad \left.\delta_{12}^8 \sim (2,1,2), \delta_{12}^{11} \sim (2,3,1), \delta_{12}^{12} \sim (2,3,2)\right\}. \end{aligned}$$

Remark 3.2. *Assume $X_i \in \mathcal{D}_{k_i}$, $i \in [1,n]$. The states have two expressions:*

(i) Component-wise expression:

$$X = (X_1, X_2, \cdots, X_n).$$

(ii) Vector form: Assume $x_i = \vec{X}_i$, $i \in [1,n]$, then the state X can be expressed into its vector form as

$$x = \ltimes_{i=1}^n x_i.$$

Now there are two ways to convert them from one form to the other.

(i) Tabulator method:
Arrange $(X_1, X_2, \cdots, X_n)$ in alphabetic order, then their corresponding vector expressions are in order. For instance, recall Example 3.1. If (X_1, X_2, X_3) are in alphabetic order, then their vector forms are already in order. See Table 3 for this.

Table 3. Conversion between two forms.

X	(1,1,1)	(1,1,2)	(1,2,1)	(1,2,2)	(1,3,1)	(1,3,2)
$x = \vec{X}$	δ_{12}^1	δ_{12}^2	δ_{12}^3	δ_{12}^4	δ_{12}^5	δ_{12}^6
X	(2,1,1)	(2,1,2)	(2,2,1)	(2,2,2)	(2,3,1)	(2,3,2)
$x = \vec{X}$	δ_{12}^7	δ_{12}^8	δ_{12}^9	δ_{12}^{10}	δ_{12}^{11}	δ_{12}^{12}

(ii) Formula method:

$$(X_1, X_2, \cdots, X_n) \Rightarrow (x_1, x_2, \cdots, x_n) \Rightarrow x = \ltimes_{i=1}^n x_i$$

is easily calculated. We consider how to convert x back to $(X_1, X_2, \cdots, X_n)$*. Define*

$$\Phi_i = \begin{cases} I_{k_1} \otimes \mathbf{1}^T_{k_2 k_3 \cdots k_n}, & i = 1, \\ \mathbf{1}^T_{k_1 k_2 \cdots k_{i-1}} \otimes I_i \otimes \mathbf{1}^T_{k_{i+1} \cdots k_n}, & 1 < i < n, \\ \mathbf{1}^T_{k_1 k_2 \cdots k_{n-1}} \otimes I_{k_n} & i = n. \end{cases}$$

Then

$$x_i = \Phi_i x, \quad i \in [1, n].$$

4. Controllability

To describe the transition mapping of a control network, we first introduce a multi-valued transition matrix.

Definition 4.1. *Let* $N = \{1, 2, \cdots, \kappa\}$ *be a given finite set. A multi-valued transition matrix is an* $\kappa \times \kappa$ *matrix* $M = [M_{i,j}]$*, where* $M_{i,j} \in N$.

$M_{i,j} \in N$ means that each element of the matrix M belongs to set N, which is also the origin of the name of multi-valued transition matrix.

Definition 4.2. *For system (11), let* $X(0) \in \mathcal{D}_\kappa$ *be the initial state and* $k > 0$.

- $X(k) \in \mathcal{D}_\kappa$ *is said to be reachable from* $X(0)$ *at time k if there exists a control sequence* $U(t)$*,* $t = 1, 2, \ldots, k-1$*, that leads system (11) from* $X(0)$ *to* $X(k)$*. The reachable set from* $X(0)$ *at time k is denoted by* $\mathcal{R}_k(X(0))$*. The overall reachable set from* $X(0)$ *is denoted by* $\mathcal{R}(X(0)) = \mathcal{D}_\kappa$*.*
- *System (11) is said to be controllable at* $X(0)$ *if* $\mathcal{R}(X(0)) = \mathcal{D}_\kappa$*.*
- *System (11) is said to be controllable if it is controllable at every* $X(0) \in \mathcal{D}_\kappa$*.*

It is well known that in game theory bi-matrix is widely used [37]. Multi-valued matrix can be considered as a generalization of bi-matrix. We give an example to depict it.

Example 4.3. *Consider a mix-valued network*

$$\begin{cases} X_1(t+1) = X_2(t) \Delta U(t) \\ X_2(t+1) = X_1(t) \Box X_2(t), \end{cases} \tag{19}$$

where $X_1(t), U(t) \in \mathcal{D}_2$, $X_2(t) \in \mathcal{D}_3$, $\Delta : \mathcal{D}_3 \times \mathcal{D}_2 \to \mathcal{D}_2$ is determined by its structure matrix

$$M_\Delta = \delta_2[2,1,1,1,2,1],$$

and $\Box : \mathcal{D}_2 \times \mathcal{D}_3 \to \mathcal{D}_3$ is determined by its structure matrix

$$M_\Box = \delta_3[3,1,1,2,2,2].$$

The component-wise ASSR of (19) can be calculated as

$$\begin{array}{rcl} x_1(t+1) & = & M_\Delta x_2(t)u(t) \\ & = & M_\Delta W_{[2,3]}u(t)x_2(t) \\ & = & M_\Delta W_{[2,3]}(I_2 \otimes \mathbf{1}_2^T \otimes I_3)u(t)x(t) \\ & := & L_1 u(t)x(t). \end{array}$$

Then

$$\begin{array}{rcl} L_1 & = & M_\Delta W_{[2,3]}(I_2 \otimes \mathbf{1}_2^T \otimes I_3) \\ & = & \delta_2[2,1,2,2,1,2,1,1,1,1,1,1]. \end{array}$$

$$\begin{array}{rcl} x_2(t+1) & = & M_\Box x_1(t)x_2(t) \\ & = & M_\Box(\mathbf{1}_2^T \otimes I_6)u(t)x(t) \\ & := & L_2 u(t)x(t). \end{array}$$

Then

$$\begin{array}{rcl} L_2 & = & M_\Box(\mathbf{1}_2^T \otimes I_6) \\ & = & \delta_3[3,1,1,2,2,2,3,1,1,2,2,2]. \end{array}$$

Finally, we have ASSR of (19) as

$$x(t+1) = Lu(t)x(t), \tag{20}$$

where

$$L = L_1 * L_2 = \delta_6[6,1,4,5,2,5,3,1,1,2,2,2].$$

Since u can be either 0 or 1, as $x(0) = \delta_6^1$ the one step transition mapping, denoted by π_L, maps δ_6^1 into two possible values δ_6^6 and δ_6^3. We briefly denote it by $\pi_L(1) = \{6,3\}$. Similarly for other states, we have the multi-valued transition mapping, described in Table 4.

Table 4. π_L of (19).

x	1	2	3	4	5	6
$\pi_L(x)$	6,3	1	4,1	5,2	2	5,2

Using this mapping, the controllability of (19) can easily be calculated. We construct a set transition matrix. The constructing process is similar to the construction of Table 2. The only difference is now each step the image of π_L is a set. Moreover, in the following we construct each column of entries only by picking out new elements. As the process does not provide new element, we stop this column with $\emptyset$. Then we have Table 5 as follows.

Table 5. Trajectory table of (19).

x	1	2	3	4	5	6
R_1	6,3	1	4,1	5,2	2	5,2
R_2	5,2,4,1	6,3	5,2,6,3	1	1	1
R_3	$\emptyset$	5,2,4	$\emptyset$	6,3	6,3	6,3
R_4		$\emptyset$		4	5,4	4
R_5				$\emptyset$	$\emptyset$	$\emptyset$

By construction the following result is obvious.

Proposition 4.4. *Consider the multi-valued trajectory matrix of a control network. The union of elements of each column of set entries, denoted by $R(i)$, $i \in [1,\kappa]$, is the reachable set of the initial element corresponding to this column. So δ_κ^i is reachable (to any δ_κ^j, $j \in [1,\kappa]$), if and only if, $R(i) = N = \{1, 2, \cdots, \kappa\}$. The network is completely controllable, if and only if, $R(i) = N$, $\forall i \in [1,\kappa]$.*

Proof. The union of elements of each column of set entries is actually the reachable set $\mathcal{R}(X(0))$ mentioned in Definition 4.2, where the first element of each column corresponds to the initial state $X(0)$. $R(i) = N = \{1, 2, \cdots, \kappa\}$ means that $\mathcal{R}(X(0)) = \mathcal{D}_\kappa$, which is exactly the second rule of the definition of controllability. $\forall i \in [1,\kappa]$, $R(i) = N$ means that it is controllable at every $X(0) \in \mathcal{D}_\kappa$, which is exactly the third rule of the definition of controllability. □

Example 4.5. *Consider network (19). Observe its multi-valued trajectory matrix in Table 5. According to Proposition 4.4, (19) is completely controllable.*

5. Observability

Definition 5.1. *System with outputs (11) is said to be observable if for any initial state $X(0)$, there exists at least a control sequence, such that the initial state can be determined by the output sequence.*

Observability of Boolean networks has been discussed for long time [3]. [38] proposed a method to solve the observability of Boolean networks via set controllability. Following the technique of [38], the method of multi-valued trajectory matrix can be used to solve observability of mix-valued control networks.

We use an example to describe this.

Example 5.2. *Recall network (19). Assume the output is*

$$Y(t) = X_1(t)\Delta X_2(t) \in \mathcal{D}_2. \tag{21}$$

Following [38], we construct an auxiliary system as

$$\begin{cases} X_1(t+1) = X_2(t)\Delta U(t) \\ X_2(t+1) = X_1(t)\Box X_2(t) \\ Z_1(t+1) = Z_2(t)\Delta U(t) \\ Z_2(t+1) = Z_1(t)\Box Z_2(t). \end{cases} \tag{22}$$

Set $\Xi(t) = [X_1(t), X_2(t), Z_1(t), Z_2(t)]$. Then the ASSR of Ξ can be obtained as follows:

$$\begin{aligned}
\xi(t+1) &= x(t+1)z(t+1) \\
&= Lu(t)x(t)Lu(t)z(t) \\
&= L(I_{12} \otimes L)u(t)x(t)u(t)z(t) \\
&= L(I_{12} \otimes L)u(t)W_{[2,6]}u(t)x(t)z(t) \\
&= L(I_{12} \otimes L)(I_2 \otimes W_{[2,6]})u^2(t)\xi(t) \\
&= L(I_{12} \otimes L)(I_2 \otimes W_{[2,6]})PR_2u(t)\xi(t) \\
&:= \Psi u(t)\xi(t),
\end{aligned} \tag{23}$$

where

$$\begin{aligned}
\Psi &= L(I_{12} \otimes L)(I_2 \otimes W_{[2,6]})PR_2 \\
&= \delta_{36}[36, 31, 34, 35, 32, 35, 6, 1, 4, \\
&\quad 5, 2, 5, 24, 19, 22, 23, 20, 23, \\
&\quad 30, 25, 28, 29, 26, 29, 12, 7, 10, \\
&\quad 11, 8, 11, 30, 25, 28, 29, 26, 29, \\
&\quad 15, 13, 13, 14, 14, 14, 3, 1, 1, \\
&\quad 2, 2, 2, 3, 1, 1, 2, 2, 2, \\
&\quad 9, 7, 7, 8, 8, 8, 9, 7, 7, \\
&\quad 8, 8, 8, 9, 7, 7, 8, 8, 8].
\end{aligned}$$

The mapping $\pi_\Psi : N \to 2^N$ can then be determined by Ψ as:

$$\begin{aligned}
&\pi_\Psi(1) = \{36, 15\}, \quad \pi_\Psi(2) = \{31, 13\}, \\
&\pi_\Psi(3) = \{34, 13\}, \quad \pi_\Psi(4) = \{35, 14\}, \\
&\cdots, \\
&\pi_\Psi(35) = \{26, 8\}, \quad \pi_\Psi(36) = \{29, 8\}.
\end{aligned}$$

The $\xi(t) = x(t)z(t)$ is called the one-step indistinguishable state, if $x(t) \neq z(t)$ and $y(x(t)) = y(z(t))$. The set of one-step indistinguishable states is denoted by $\mathcal{I}$. If $x(t) \neq z(t)$ and $y(x(t)) \neq y(z(t))$, $\xi(t)$ is called the one-step distinguishable state, and the set of one-step distinguishable states is denoted by $\mathcal{J}$. Then it is clear that [38] a control network is observable, if and only if, each state $\xi \in \mathcal{I}$ can be driven into $\mathcal{J}$.

Using M_Δ, it is easy to verify that for network (23) we have

$$\begin{aligned}
\mathcal{I} &= \delta_{36}\{5, 9, 10, 12, 16, 18, 24\}, \\
\mathcal{J} &= \delta_{36}\{2, 3, 4, 6, 11, 17, 23, 30\}.
\end{aligned}$$

Now we construct the multi-valued trajectory matrix for $\mathcal{I}$ in Table 6. In each column, when $\mathcal{J}$ is reached $\mathcal{J}$ is used to end the trajectory. If there is no new state appearing, $\emptyset$ is used to end the trajectory, which means the $\xi = xz$ in this column is not distinguishable at all.

Now it is clear that the network (19)–(21) is not observable. The only indistinguishable initial pair is $\delta_{36}^{24} = \delta_6^4\delta_6^6 \sim \{(2, 1), (2, 3)\}$.

Table 6. Trajectory table of auxiliary network (22).

ξ	5	9	10	12	16	18	24
R_1	32,14	4,1	5,2	5,2	23,2	23,2	29,8
R_2	25,7,19,1	$\mathcal{J}$	$\mathcal{J}$	$\mathcal{J}$	$\mathcal{J}$	$\mathcal{J}$	1
R_3	12,9,6,3,30,15,36						36,15
R_4	$\mathcal{J}$						22
R_5							$\emptyset$

Proposition 5.3. *Consider the multi-valued trajectory matrix of a control network. The network is observable if and only if for any state $\delta_{2^n}^i \in I$, it holds that $R(i) \cap \mathcal{J} \neq \emptyset$.*

Proof. $R(i) \cap \mathcal{J} \neq \emptyset$ holds for any $\delta_{2^n}^i \in I$, if and only if each element in the set of one-step indistinguishable states $\mathcal{I}$ can be driven into the set of one-step distinguishable states $\mathcal{J}$ by a proper control, which means that any two initial states can be distinguished, i.e., any initial state can be determined by the output sequence. □

Remark 5.4. *Assume a mix-valued logical network with n nodes, m inputs and p outputs. Assume $X_i \in \mathcal{D}_{k_i}$, $U_s \in \mathcal{D}_{\alpha_s}$, $Y_l \in \mathcal{D}_{\beta_l}$, $i = 1, 2, \ldots, n$, $s = 1, 2, \ldots, m$, $l = 1, 2, \ldots, p$. Next we consider the computational complexity of the methods in this paper when the vector form of the system is known. For the method of calculating the attractors in Section 3, since there are at most κ states, each column of the table will end after at most $\kappa - 1$ steps. Hence, the computational complexity of calculating the attractors is $O(\kappa(\kappa - 1))$ in the worse case, where $\kappa = \prod_{i=1}^n k_i$. For the method of determining controllability in Section 4, there are also at most κ states, and the state generated by every step should be compared with the previous states. Each state needs to be compared at most $1 + 2 + \cdots + (\kappa - 1)$ times. Therefore, the computational complexity of determining controllability is $O(\frac{\kappa^2(\kappa-1)}{2})$ in the worse case. For the method of determining observability in Section 5, there are also at most $1+2+\cdots+(\kappa-1)$ state pairs, and each column of the table will end after at most $\kappa - 1$ steps. Hence, the computational complexity of determining observability is $O(\frac{\kappa(\kappa-1)^2}{2})$ in the worse case, where $\kappa = \prod_{i=1}^n k_i$.*

6. Conclusions

Structure vector method for logical (control) networks has been proposed in order to reduce the computational complexity. Using trajectory table, topology structure, controllability and observability has been analyzed. Compared with the existing method, the vector form expression greatly reduces the computational complexity. The method can be used to discuss other problems of logical networks, such as decoupling, delectability, tracking, etc.

Acknowledgments

This work is supported by the National Natural Science Foundation of China (62103716), and the Natural Science Foundation of Shandong Province (ZR2019BF023).

Conflict of interest

All authors declare no conflicts of interest in this paper.

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. S. Wang, J. E. Feng, Y. Yu, J. Zhao, Further results on dynamic-algebraic Boolean control networks, *Sci. China Inform. Sci.*, **62** (2019), 1–14. https://doi.org/10.1007/s11432-018-9447-4
3. D. Cheng, H. Qi, Controllability and observability of Boolean control networks, *Automatica*, **45** (2009), 1659–1667. https://doi.org/10.1016/j.automatica.2009.03.006
4. D. Cheng, Input-state approach to Boolean networks, *IEEE T. Neural Networ.*, **20** (2009), 512–521. https://doi.org/10.1109/TNN.2008.2011359
5. D. Cheng, H. Qi, Z. Li, *Analysis and control of Boolean networks: A semi-tensor product approach*, London: Springer, 2011.
6. J. Zhong, D. W. C. Ho, J. Lu, A new approach to pinning control of Boolean networks, *IEEE T. Control Netw.*, 2021. https://doi.org/10.1109/TCNS.2021.3106453
7. M. R. Rafimanzelat, F. Bahrami, Attactor controllability of Boolean networks by fipping a subset of their nodes, *Chaos*, **28** (2018), 043120. https://doi.org/10.1063/1.4999950
8. F. Li, Y. Tang, Pinning controllability for a Boolean network with arbitrary disturbance inputs, *IEEE T. Cybernetics*, **51** (2021), 3338–3347. https://doi.org/10.1109/TCYB.2019.293 0734
9. Y. Yu, M. Meng, J. Feng, G. Chen, Observability criteria for Boolean networks, *IEEE T. Automat. Contr.*, 2021. https://doi.org/10.1109/TAC.2021.3131436
10. Y. Yu, M. Meng, J. Feng, Observability of Boolean networks via matrix equations, *Automatica*, **111** (2020), 108621. https://doi.org/10.1016/j.automatica.2019.108621
11. J. Yang, W. Qian, Z. Li, Redefined reconstructibility and state estimation for Boolean networks, *IEEE T. Control Netw.*, **7** (2020), 1882–1890. https://doi.org/10.1109/TCNS.2020.3007820
12. H. Li, X. Yang, S. Wang, Robustness for stability and stabilization of Boolean networks with stochastic function perturbations, *IEEE T. Automat. Contr.*, **66** (2021), 1231–1237. https://doi.org/10.1109/TAC.2020.2997282
13. Q. Zhang, J. Feng, Y. Zhao, J. Zhao, Stabilization and set stabilization of switched Boolean control networks via flipping mechanism, *Nonlinear Anal.: Hybrid Syst.*, **41** (2021), 101055. https://doi.org/10.1016/j.nahs.2021.101055
14. B. Chen, X. Yang, Y. Liu, J. Liu, Controllability and stabilization of Boolean control networks by the auxiliary function of flipping, *Int. J. Robust Nonlinear Control*, **30** (2020), 5529–5541. https://doi.org/10.1002/rnc.5091
15. Y. Wang, P. Guo, Optimal control of singular Boolean control networks via Ledley solution method, *J. Franklin Inst.*, **358** (2021), 6161–6173. https://doi.org/10.1016/j.jfranklin.2021.06.006

16. X. Ding, H. Li, Optimal control of random evolutionary Boolean games, *Int. J. Control*, **94** (2021), 144–152. https://doi.org/10.1080/00207179.2019.1585957

17. Y. Zheng, J. Feng, Output tracking of delayed logical control networks with multi-constraints, *Front. Inform. Technol. Electron. Eng.*, **21** (2020), 316–323. https://doi.org/10.1631/FITEE.1900376

18. Q. Zhang, J. Feng, T. Jiao, Finite horizon tracking control of probabilistic Boolean control networks, *J. Franklin Inst.*, **358** (2021), 9909–9928. https://doi.org/10.1016/j.jfranklin.2021.10.003

19. Y. Li, J. Zhu, B. Li, Y. Liu, J. Lu, A necessary and sufficient graphic condition for the original disturbance decoupling of Boolean networks, *IEEE T. Automat. Contr.*, **66** (2021), 3765–3772. https://doi.org/10.1109/TAC.2020.3025507

20. Y. Yu, J. Feng, J. Pan, D. Cheng, Block decoupling of Boolean control networks, *IEEE T. Automat. Contr.*, **64** (2019), 3129–3140. https://doi.org/10.1109/TAC.2018.2880411

21. M. Meng, J. Feng, Z. Hou, Synchronization of interconnected multi-valued logical networks, *Asian J. Control*, **16** (2014), 1659–1669. https://doi.org/10.1002/asjc.835

22. Y. Li, J. Feng, S. Zhu, Controllability and reachability of periodically time-variant mixed-valued logical control networks, *Circuits, Syst., Signal Process.*, **40** (2021), 3639–3654. https://doi.org/10.1007/s00034-021-01648-2

23. Y. Li, H. Li, X. Ding, Set stability of switched delayed logical networks with application to finite-field consensus, *Automatica*, **113** (2021), 108768. https://doi.org/10.1016/j.automatica.2019.108768

24. Y. Wu, S. Le, K. Zhang, X. Sun, Ex-ante agent transformation of Bayesian games, *IEEE T. Automat. Contr.*, 2021. https://doi.org/10.1109/TAC.2021.3122372

25. H. Qi, B. Mu, I. R. Petersen, G. Shi, Measurement-induced Boolean dynamics and controllability for closed quantum networks, *Automatica*, **114** (2020), 108816. https://doi.org/10.1016/j.automatica.2020.108816

26. J. Zhong, D. Lin, Decomposition of nonlinear feedback shift registers based on Boolean networks, *Sci.China Inform. Sci.*, **62** (2019), 1–3. https://doi.org/10.1007/s11432-017-9460-4

27. J. Lu, B. Li, J. Zhong, A novel synthesis method for reliable feedback shift registers via Boolean networks, *Sci. China Inform. Sci.*, **64** (2021), 1–14. https://doi.org/10.1007/s11432-020-2981-4

28. X. Xu, Y. Hong, Observability analysis and observer design for finite automata via matrix approach, *IET Control Theory Appl.*, **7** (2013), 1609–1615.

29. X. Han, Z. Chen, Z. Liu, Q. Zhang, Calculation of siphons and minimal siphons in petri nets based on semi-tensor product of matrices, *IEEE T. Syst., Man, Cybern.: Syst.*, **47** (2017), 531–536. https://doi.org/10.1109/TSMC.2015.2507162

30. H. Lyu, W. Wang, X. Liu, Universal approximation of multi-variable fuzzy systems by semi-tensor product, *IEEE T. Fuzzy Syst.*, **28** (2020), 2972–2981.

31. D. Cheng, Y. Li, J. Feng, J. Zhao, On numerical/non-numerical algebra: Semi-tensor product method, *Math. Model. Control*, **1** (2021), 1–11. https://doi.org/10.3934/mmc.2021001

32. S. Fu, D. Cheng, J. Feng, J. Zhao, Matrix expression of finite Boolean-type algebras, *Appl. Math. Comput.*, **395** (2021), 125880. https://doi.org/10.1016/j.amc.2020.125880

33. Z. Liu, J. Zhong, Y. Liu, W. Gui, Weak stabilization of Boolean networks under state-flipped control, *IEEE T. Neural Netw. Learn. Syst.*, 2021. https://doi.org/10.1109/TNNLS.2021.3106918

34. A. Acernese, A. Yerudkar, L. Glielmo, C. Del Vecchio, Model-free self-triggered control co-design for probabilistic Boolean control networks, *IEEE Control Syst. Lett.*, **5** (2021), 1639–1644. https://doi.org/10.1109/LCSYS.2020.3042394

35. A. Acernese, A. Yerudkar, L. Glielmo, C. Del Vecchio, Double deep-Q learning-based output tracking of probabilistic Boolean control networks, *IEEE Access*, **8** (2020), 199254–199265. https://doi.org/10.1109/ACCESS.2020.3035152

36. D. Cheng, H. Qi, Y. Zhao, *An introduction to semi-tensor product of matrices and its applications*, Singapore: World Scientific, 2012.

37. R. Gibbons, *A primer in game theory*, New York: Printice Hall, 1992.

38. D. Cheng, C. Li, F. He, Observability of Boolean networks via set controllability, *Syst. Control Lett.*, **115** (2018), 22–25. https://doi.org/10.1016/j.sysconle.2018.03.004

AIMS Press