INTERNATIONAL JOURNAL OF ROBUST AND NONLINEAR CONTROL
Int. J. Robust. Nonlinear Control 2011; **21**:134–156
DOI: 10.1002/rnc.1581

Stability and stabilization of Boolean networks

Daizhan Cheng[1], Hongsheng Qi[1], Zhiqiang Li[1] and Jiang B. Liu[2,3,*,†]

[1]*Laboratory of Systems and Control, AMSS, Chinese Academy of Sciences, Beijing 100190, People's Republic of China*
[2]*State Key Laboratory of Software Engineering, Wuhan University, Wuhan 430072, People's Republic of China*
[3]*Department of Computer Science and Information Systems, Bradley University, Peoria, IL, U.S.A.*

SUMMARY

The stability of Boolean networks and the stabilization of Boolean control networks are investigated. Using semi-tensor product of matrices and the matrix expression of logic, the dynamics of a Boolean (control) network can be converted to a discrete time linear (bilinear) dynamics, called the algebraic form of the Boolean (control) network. Then the stability can be revealed by analyzing the transition matrix of the corresponding discrete time system. Main results consist of two parts: (i) Using logic coordinate transformation, the known sufficient condition based on incidence matrix has been improved. It can also be used in stabilizer design. (ii) Based on algebraic form, necessary and sufficient conditions for stability and stabilization, respectively, are obtained.

Received 22 October 2009; Accepted 6 January 2010

KEY WORDS: Boolean (control) system; stability; stabilization; semi-tensor product of matrices

1. INTRODUCTION

Accompanying the rising of systems biology, the study of Boolean network has attracted a great attention of biologists, physicists, and systems scientists, because it is a proper model to describe cellular networks and genomic regulatory networks [1, 2]. The control of Boolean network is a challenging problem. So far, there are only few results because we are short of systematic tool to deal with logical dynamic systems [3, 4].

Recently, a new matrix product, namely, the semi-tensor product (STP) of matrices, was proposed, which generalizes the conventional matrix product, AB, $A\in M_{p\times q}$ and $B\in M_{m\times n}$ for the case $q\neq m$, where $M_{m\times n}$ stands for the set of $m\times n$ matrices. Using STP, a logical equation can be expressed as an algebraic equation and the dynamics of a Boolean (control) network can be converted into a linear (bilinear) discrete-time (control) system [5]. Then investigating the corresponding discrete time (control) systems, some interesting progresses have been achieved, which are briefly described as follows: (1) Formulas for calculating fixed points and cycles have been obtained [6]; (ii) steering gears' structure of attractors in Boolean network has been revealed [7], which explains why tiny attractors decide the vast order, as described in [8]; (iii) the controllability

---

[*] Correspondence to: Jiang B. Liu, Department of Computer Science and Information Systems, Bradley University, Peoria, IL, U.S.A.

[†] E-mail: jiangbo@bumail.bradley.edu, jiangbo@bradley.edu

Contract/grant sponsor: NNSF; contract/grant numbers: 60674022, 60736022, 60821091

and observability of Boolean control networks are investigated [9]; (iv) coordinate transformation of Boolean networks and the realization of Boolean control networks are presented [10].

Stability of Boolean networks is another interesting topic [11, 12]. In fact, the stability of a Boolean network is essentially equivalent to the convergence of a discrete iteration [13]. As for the stabilization of a Boolean control network, to the best of authors' knowledge, it is a new topic.

This paper considers the stability of Boolean networks and the stabilization of Boolean control networks. The key tool used in this paper is the STP of matrices and the matrix expression of logic, which have been used in the sequence of works [6, 7, 9, 10].

The paper is organized as follows. Section 2 provides necessary preliminaries, including (i) matrix expression of logic; (ii) Boolean (control) network; (iii) logical coordinate transformation; and (iv) incidence matrix. Section 3 discusses the concepts and basic properties of Boolean matrices. The vector distance of Boolean matrices is also introduced. The global stability of Boolean networks is investigated in Section 4. Based on the incidence matrix and the vector distance of Boolean matrices, a convenient sufficient condition is obtained. Moreover, the necessary and sufficient condition is also presented. Section 5 considers the stabilization of Boolean control networks. Some examples are included to depict the technique. Section 6 is the conclusion.

2. PRELIMINARIES

2.1. Matrix expression of logic

We first recall the definition of STP of matrices.

Definition 2.1 (*Cheng* [5])

1. Let X be a row vector of dimension np and Y be a column vector with dimension p. Then we split X into p equal-sized blocks as $X^1, \dots, X^p$, which are $1\times n$ rows. Define the STP, denoted by $\ltimes$, as

$$\begin{aligned} X \ltimes Y &= \sum_{i=1}^{p} X^i y_i \in \mathbb{R}^n, \\ Y^{\mathrm{T}} \ltimes X^{\mathrm{T}} &= \sum_{i=1}^{p} y_i (X^i)^{\mathrm{T}} \in \mathbb{R}^n. \end{aligned} \tag{1}$$

2. Let $A \in M_{m\times n}$ and $B \in M_{p\times q}$. If either n is a factor of p, say $nt=p$ and denote it as $A \prec_t B$, or p is a factor of n, say $n=pt$ and denote it as $A \succ_t B$, then we define the STP of A and B, denoted by $C = A \ltimes B$, as the following: C consists of $m\times q$ blocks as $C=(C^{ij})$ and each block is

$$C^{ij} = A^i \ltimes B_j, \quad i=1,\dots,m, \quad j=1,\dots,q,$$

where A^i is the ith row of A and B_j is the jth column of B.

STP of matrices is a generalization of conventional matrix product. Hence, we can omit the symbol $\ltimes$. All fundamental properties of conventional matrix product remain true [5]. Throughout this paper the matrix products are assumed to be STP. As $n=p$ they becomes conventional matrix product automatically.

We briefly define some concepts and/or notations first.

- Let δ_n^i be the ith column of the identity matrix I_n, and $\Delta_n := \{\delta_n^1, \delta_n^2, \dots, \delta_n^n\}$. When $n=2$ we simply use $\Delta := \Delta_2$.
- The set of logical values: True ($T \sim 1$) and False ($F \sim 0$), is denoted by $\mathscr{D} = \{0, 1\}$. We identify each with a vector as $T \sim \delta_2^1$ and $F \sim \delta_2^2$. Hence, in vector form the set of logical values becomes Δ. In this sense we have the equivalence as $\mathscr{D} \sim \Delta$.

DOI: 10.1002/rnc

- Assume a matrix $M=[\delta_n^{i_1}\delta_n^{i_2}\dots\delta_n^{i_s}]\in M_{n\times s}$, i.e. its columns, $\mathrm{Col}(M)\subset\Delta_n$. We call M a logical matrix, and simply denote it as:

$$M=\delta_n[i_1 i_2 \dots i_s].$$

- The set of $n\times s$ logical matrices is denoted by $\mathscr{L}_{n\times s}$.
- A matrix $B\in M_{n\times s}$ is called a Boolean matrix, if its entries $b_{ij}\in\mathscr{D}$, $\forall i,j$.
- The set of $n\times s$ Boolean matrices is denoted by $\mathscr{B}_{n\times s}$.

In vector form, we have

Proposition 2.2 (*Cheng* [5])
Let $x_1,\dots,x_n\in\mathscr{D}$ be n logical variables, and $f(x_1,\dots,x_n)$ a logical function. Then there exists a unique matrix $M_f\in\mathscr{L}_{2\times 2^n}$, called the structure matrix of f, such that in vector form we have

$$f(x_1,\dots,x_n)=M_f\ltimes_{i=1}^n x_i,\quad x_i\in\Delta.$$

Logical operators are fundamental logical functions. Their structure matrices are particularly useful. In the following table we list the structure matrices for some basic logical operators (Negation: $\neg$; Conjunction: $\wedge$; Disjunction: $\vee$; Conditional: $\rightarrow$; Biconditional: $\leftrightarrow$; Exclusive Or: $\bar{\vee}$ [14]), which are used in the sequel.

2.2. *Boolean* (*control*) *network*

A Boolean network, N, of n elements may be described by its network graph, $G(N)$. $G(N)$ consists of n nodes, denoted by $\mathscr{N}=\{x_1,\dots,x_n\}$, and a set of directed edges $\mathscr{E}\subset\mathscr{N}\times\mathscr{N}$. $\overline{x_j,x_i}\in\mathscr{E}$ (i.e. there is an edge from x_j to x_i) means x_i is affected by x_j. We refer to Figure 1 for a network graph (within the rectangular box).

A network graph is not enough to describe a Boolean network completely. A logical dynamical equation is necessary to determine a Boolean network. The dynamics of a Boolean network is expressed as

$$\begin{aligned}
x_1(t+1)&=f_1(x_1,\dots,x_n)\\
x_2(t+1)&=f_2(x_1,\dots,x_n)\\
&\vdots\\
x_n(t+1)&=f_n(x_1,\dots,x_n),\quad x_i\in\mathscr{D},
\end{aligned}\tag{2}$$

where f_i, $i=1,\dots,n$ are logical functions. Let $X=(x_1,\dots,x_n)^{\mathrm{T}}$ and $F=(f_1,\dots,f_n)^{\mathrm{T}}$. Then (2) can briefly denoted as:

$$X(t+1)=F(X(t)).\tag{3}$$

Using the vector form and setting $x(t):=\ltimes_{i=1}^n x_i(t)\in\Delta^n$, and denoting by M_i the structure matrix of f_i, we can express each equation in (2) in a vector form as:

$$x_i(t+1)=M_i x(t),\quad x_i\in\Delta, i=1,\dots,n.\tag{4}$$

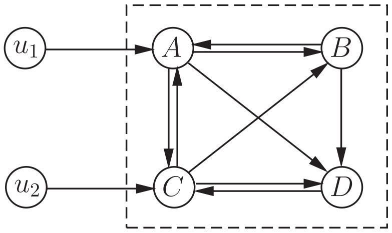

Figure 1. A Boolean (control) network.

Finally, we can obtain an algebraic form of (2) (equivalently, (3)) as [6]

$$x(t+1) = Lx(t), \quad x \in \Delta^n, \tag{5}$$

where $L \in \mathscr{L}_{2^n \times 2^n}$ is the transition matrix.

The dynamics of a Boolean control network is expressed as

$$\begin{aligned} x_1(t+1) &= f_1(x_1, \ldots, x_n, u_1, \ldots, u_m) \\ x_2(t+1) &= f_2(x_1, \ldots, x_n, u_1, \ldots, u_m) \\ &\vdots \\ x_n(t+1) &= f_n(x_1, \ldots, x_n, u_1, \ldots, u_m), \end{aligned} \tag{6}$$

where f_i, $i = 1, \ldots, n$, are logical functions. Similarly, we denote it briefly as:

$$X(t+1) = F(X(t), U(t)), \quad X(t) \in \mathscr{D}^n, \ U(t) \in \mathscr{D}^m. \tag{7}$$

Let $x(t) := \ltimes_{i=1}^n x_i(t)$ and $u(t) := \ltimes_{i=1}^m u_i(t)$. We can equivalently express (6) (equivalently, (7)) in an algebraic form as [7, 9]

$$x(t+1) = Lu(t)x(t), \quad x \in \Delta^n, \ u \in \Delta^m, \tag{8}$$

where $L \in \mathscr{L}_{2^n \times 2^{n+m}}$ is also called the transition matrix.

We give an example to illustrate them.

Example 2.3

1. In Figure 1 within the rectangular box we have a Boolean network of four nodes. Assume that its dynamics is described by the following equation:

$$\begin{aligned} A(t+1) &= B(t) \vee C(t) \\ B(t+1) &= A(t) \leftrightarrow C(t) \\ C(t+1) &= A(t) \wedge D(t) \\ D(t+1) &= (A(t) \rightarrow B(t)) \bar{\vee} C(t). \end{aligned} \tag{9}$$

   Define $x(t) = A(t)B(t)C(t)D(t)$, it is easy to check that its algebraic form is $x(t+1) = Lx(t)$ with

$$L = \delta_{16}[2\ 4\ 5\ 7\ 1\ 3\ 14\ 16\ 8\ 8\ 3\ 3\ 8\ 8\ 11\ 11].$$

   (Please refer to the Acknowledgment for a Toolbox, which provides a simple function to calculate this.)

2. Consider the overall Boolean control network in Figure 1. Say, its dynamics is

$$\begin{aligned} A(t+1) &= (B(t) \vee C(t)) \wedge u_1 \\ B(t+1) &= A(t) \leftrightarrow C(t) \\ C(t+1) &= (A(t) \wedge D(t)) \vee u_2 \\ D(t+1) &= (A(t) \rightarrow B(t)) \bar{\vee} C(t). \end{aligned} \tag{10}$$

   Then its algebraic form is $x(t+1) = Lu(t)x(t)$, where L can be calculated as

$$\begin{array}{rcccccccccccccccc} L = \delta_{16}[2 & 2 & 5 & 5 & 1 & 1 & 14 & 14 & 6 & 6 & 1 & 1 & 6 & 6 & 9 & 9 \\ 2 & 4 & 5 & 7 & 1 & 3 & 14 & 16 & 8 & 8 & 3 & 3 & 8 & 8 & 11 & 11 \\ 10 & 10 & 13 & 13 & 9 & 9 & 14 & 14 & 14 & 14 & 9 & 9 & 14 & 14 & 9 & 9 \\ 10 & 12 & 13 & 15 & 9 & 11 & 14 & 16 & 16 & 16 & 11 & 11 & 16 & 16 & 11 & 11]. \end{array}$$

2.3. *Logical coordinate transformation*

Definition 2.4
Let $x_1, x_2, \ldots, x_n \in \mathscr{D}$ be a set of independent logical variables. A mapping $\varphi : \mathscr{D}^n \to \mathscr{D}^n$, denoted by $Y = \varphi(X)$ and defined by

$$\varphi : \begin{cases} y_1 = g_1(x_1, \ldots, x_n) \\ \vdots \\ y_n = g_n(x_1, \ldots, x_n), \end{cases} \tag{11}$$

is called a coordinate transformation, if (11) is a one-to-one and onto mapping.

In vector form we have $x_i, y_i \in \Delta$, $\forall i$. Setting $x = \ltimes_{i=1}^n x_i \in \Delta^n$ and $y = \ltimes_{i=1}^n y_i \in \Delta^n$, then the algebraic form of (11) is expressed as

$$y = Tx, \tag{12}$$

where $T \in \mathscr{L}_{2^n \times 2^n}$.

Proposition 2.5 (*Cheng et al.* [*10*])
Equation (11) is a coordinate transformation, iff the transformation matrix T in (12) is nonsingular.

Note that since $T \in \mathscr{L}_{2^n \times 2^n}$, when T is nonsingular it becomes an orthogonal matrix [10].

Proposition 2.6 (*Cheng and Qi* [*9*])

1. Consider system (2). Let $z = Tx$ be a coordinate transformation. Then the algebraic form (5) of (2) becomes

$$z(t+1) = TLT^{\mathrm{T}} z(t). \tag{13}$$

2. Consider system (5). Let $z = Tx$ be a coordinate transformation. The algebraic form (8) of (5) becomes

$$z(t+1) = TL(I_{2^m} \otimes T^{\mathrm{T}}) u(t) z(t). \tag{14}$$

2.4. *Incidence matrix*

Assume a Boolean network, N is given, with its nodes $\mathscr{N} = \{x_1, \ldots, x_n\}$ and edge set $\mathscr{E}$.

Definition 2.7
An $n \times n$ matrix $\mathscr{I}(N) = (b_{ij})$ is called the incidence matrix of N, if

$$b_{ij} = \begin{cases} 1, & \overline{x_j, x_i} \in \mathscr{E}, \\ 0 & \text{otherwise.} \end{cases}$$

Example 2.8
Consider the network (9), refer to the graph within the rectangular box of Figure 1. Its incidence matrix is

$$\mathscr{I}|_{(9)} = \begin{bmatrix} 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 \\ 1 & 1 & 1 & 0 \end{bmatrix}.$$

Remark 2.9

1. If a Boolean network N has n nodes, its incidence matrix $\mathscr{I}(N)\in\mathscr{B}_{n\times n}$.
2. The incidence matrix of a Boolean network can also be constructed from its dynamical equations. But you have to be very careful that in dynamic equations there may have some fabricated variables, which should be removed. Cheng and Qi [9] provided a mechanical procedure to remove all fabricated variables from a logic expression.

3. LOGICAL OPERATIONS ON BOOLEAN MATRICES

The main purpose is to define a distance on $\mathscr{B}_{m\times n}$. We first define the logical operators on $\mathscr{B}_{m\times n}$.

Definition 3.1

1. Let $X=(x_{ij})\in\mathscr{B}_{m\times n}$ and σ an unary logical operator. Then $\sigma:\mathscr{B}_{m\times n}\to\mathscr{B}_{m\times n}$ is defined by $\sigma X=(\sigma x_{ij})$. For instance,
$$\neg X:=(\neg x_{ij}). \tag{15}$$
2. Let $X=(x_{ij}), Y=(y_{ij})\in\mathscr{B}_{m\times n}$ and σ a binary logical operator. Then $\sigma:\mathscr{B}_{m\times n}\times\mathscr{B}_{m\times n}\to\mathscr{B}_{m\times n}$ is defined by $X\sigma Y:=(x_{ij}\sigma y_{ij})$. For instance,
$$X\vee Y:=(x_{ij}\vee y_{ij}),\ \text{etc.} \tag{16}$$
3. Let $\alpha\in\mathscr{D}$ and $X=(x_{ij})\in\mathscr{B}_{m\times n}$. σ is a binary logical operator. Then $\sigma:\mathscr{D}\times\mathscr{B}_{m\times n}\to\mathscr{B}_{m\times n}$ is defined by $\alpha\sigma X:=(\alpha\sigma x_{ij})$. Similarly, $\sigma:\mathscr{B}_{m\times n}\times\mathscr{D}\to\mathscr{B}_{m\times n}$ is defined by $X\sigma\alpha:=(x_{ij}\sigma\alpha)$. For instance,
$$\alpha\wedge X=(\alpha\wedge x_{ij});\quad X\wedge\alpha=(x_{ij}\wedge\alpha). \tag{17}$$

Next, we consider the scalar product and the (semi-tensor) product on the set of Boolean matrices.

Definition 3.2

1. Let $\alpha\in\mathscr{D}$. The scalar product of α with $X\in\mathscr{B}_{m\times n}$ is
$$\alpha\cdot X:=\alpha\wedge X,\quad X\cdot\alpha:=X\wedge\alpha. \tag{18}$$
Note that since it coincides with the conventional real number product, we use the same product symbol. For compactness, we may also even omit the symbol in the sequel.
2. Let $X=(x_{ij})\in\mathscr{B}_{m\times n}$ and $Y\in\mathscr{B}_{p\times q}$ be two Boolean matrices. Then the Kronecker product of X, Y is defined as:
$$X\otimes Y=(x_{ij}\cdot Y|i=1,\ldots,m;\, j=1,\ldots,n)\in\mathscr{B}_{mp\times nq}. \tag{19}$$
3. Let $\alpha,\beta,\alpha_i\in\mathscr{D}$, $i=1,2,\ldots,n$. The Boolean plus is defined as follows:
$$\begin{aligned}&\alpha+_B\beta:=\alpha\vee\beta,\\ &{}^{B}\sum_{i=1}^{n}\alpha_i:=\alpha_1\vee\alpha_2\vee\cdots\vee\alpha_n.\end{aligned} \tag{20}$$
4. Let $X=(x_{ij})\in\mathscr{B}_{m\times n}$ and $Y=(y_{ij})\in\mathscr{B}_{n\times p}$. Then the Boolean product of Boolean matrices
$$X\ltimes_B Y:=Z\in\mathscr{B}_{m\times p}, \tag{21}$$
where
$$z_{ij}={}^{B}\sum_{k=1}^{n}x_{ik}\cdot y_{kj},\quad i=1,\ldots,m,\ \ j=1,\ldots,p.$$

5. Let $A \prec_t B$ $(A \succ_t B)$. Then the Boolean product of A, B is defined as:

$$A \ltimes_B B := (A \otimes I_t) \ltimes_B B. \quad (A \ltimes_B B := A \ltimes_B (B \otimes I_t).) \tag{22}$$

6. Assume that $A \ltimes_B A$ is well defined. Then the Boolean power

$$A^{(k)} := \underbrace{A \ltimes_B A \ltimes_B \cdots \ltimes_B A}_{k}.$$

Note that $\ltimes_B$ may be omitted when there is no possible confusion.
We give some simple examples.

Example 3.3
Let

$$A = \begin{bmatrix} 1 & 1 \\ 0 & 1 \\ 0 & 0 \end{bmatrix}, \quad B = \begin{bmatrix} 0 & 1 \\ 1 & 0 \\ 0 & 1 \end{bmatrix}, \quad C = \begin{bmatrix} 0 & 1 \\ 1 & 0 \\ 1 & 0 \\ 0 & 1 \end{bmatrix}.$$

Then

$$\neg A = \begin{bmatrix} 0 & 0 \\ 1 & 0 \\ 1 & 1 \end{bmatrix}, \quad A \wedge B = \begin{bmatrix} 0 & 1 \\ 0 & 0 \\ 0 & 0 \end{bmatrix},$$

$$A +_B B = A \vee B = \begin{bmatrix} 1 & 1 \\ 1 & 1 \\ 0 & 1 \end{bmatrix}, \quad A \rightarrow B = \begin{bmatrix} 0 & 1 \\ 1 & 0 \\ 1 & 1 \end{bmatrix},$$

$$A \leftrightarrow B = \begin{bmatrix} 0 & 1 \\ 0 & 0 \\ 1 & 0 \end{bmatrix}, \quad A \bar{\vee} B = \begin{bmatrix} 1 & 0 \\ 1 & 1 \\ 0 & 1 \end{bmatrix},$$

$$A \ltimes_B C = \begin{bmatrix} 1 & 1 \\ 1 & 1 \\ 1 & 0 \\ 0 & 1 \\ 0 & 0 \\ 0 & 0 \end{bmatrix}, \quad B \ltimes_B C = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 1 & 0 \\ 0 & 1 \\ 1 & 0 \\ 0 & 1 \end{bmatrix}.$$

Next, we define a partial order on $\mathscr{B}_{m\times n}$, and a 'distance', called the vector distance on $\mathscr{B}_{m\times n}$.

Definition 3.4
Let $X = (x_{ij}), Y = (y_{ij}) \in \mathscr{B}_{m\times n}$. We said $X \leqslant Y$ if $x_{ij} \leqslant y_{ij}$, $\forall i, j$.

Definition 3.5
Let $X = (x_{ij}), Y = (y_{ij}) \in \mathscr{B}_{m\times n}$. The vector distance of X and Y, denoted by $d(X, Y)$, is defined as

$$d(A, B) = A \bar{\vee} B. \tag{23}$$

Since both Boolean product and Boolean plus are order-preserving, it is easy to verify the following properties, which are the generalization of the corresponding results (for vector cases) in [13].

Proposition 3.6
Assume $A\geqslant B$ and $C\geqslant D$, then (as long as the product is well defined)

$$A\ltimes_B C\geqslant B\ltimes_B D. \tag{24}$$

Proposition 3.7
Let $X,Y,Z\in\mathscr{B}_{m\times n}$. The vector distance satisfies

$$\begin{aligned} &d(X,Y)=0\Leftrightarrow X=Y,\\ &d(X,Y)=d(Y,X),\\ &d(X,Z)\leqslant d(X,Y)+_B d(Y,Z). \end{aligned} \tag{25}$$

4. GLOBAL STABILITY

This section considers global stability of a Boolean network. That is, when there exists a fixed point as the unique attractor. Equivalently, when a Boolean dynamics will globally converge to a state. A key tool in this investigation is the vector distance.

Denote $\mathscr{X}=\mathscr{D}^n$ as the state space of the Boolean network (2) (or Boolean control network (6)). (In vector form we have $\mathscr{X}=\Delta^n$.) $X\in\mathscr{X}$ is expressed as $X=(x_1,\ldots,x_n)$. we consider a logical mapping $F:\mathscr{X}\to\mathscr{X}$, which is described as:

$$\begin{aligned} z_1 &= f_1(x_1,\ldots,x_n),\\ &\vdots\\ z_n &= f_n(x_1,\ldots,x_n). \end{aligned} \tag{26}$$

It is briefly denoted as:

$$Z=F(X)\quad \text{where } X,Z\in\mathscr{X}. \tag{27}$$

This mapping may come from the Boolean network (2).

Theorem 4.1 (*Robert* [*13*])
Let $X,Y\in\mathscr{X}$. Then

$$d(F(X),F(Y))\leqslant\mathscr{I}(F)\ltimes_B d(X,Y), \tag{28}$$

where $\mathscr{I}(F)$ is the incidence matrix of F.

Theorem 4.2 (*Robert* [*13*])
For mapping (27) if there exists a matrix $M\in\mathscr{B}_{n\times n}$ such that

$$d(F(X),F(Y))\leqslant M\ltimes_B d(X,Y)\quad \forall X,Y\in\mathscr{X}, \tag{29}$$

then

$$\mathscr{I}(F)\leqslant M.$$

Theorem 4.3 (*Robert* [*13*])
Let $E,F:\mathscr{X}\to\mathscr{X}$ be two logical mappings. Then

$$\mathscr{I}(E\circ F)\leqslant\mathscr{I}(E)\ltimes_B\mathscr{I}(F). \tag{30}$$

DOI: 10.1002/rnc

An immediate application of the above theorem is

Corollary 4.4
Let ξ be a fixed point of (2). Then

$$d(X(k),\xi)\leqslant[\mathscr{I}(F)]^{(k)}\ltimes_B d(X(0),\xi). \tag{31}$$

Particularly, if the $j_1,\ldots,j_s$ columns of $[B(F)]^{(k)}$ are zero, and $x_\alpha(0)=\xi_\alpha$, $\forall\alpha\notin\{j_1,\ldots,j_s\}$, then $x(t)=\xi$, $t\geqslant k$.

It is obvious that if 0 is a fixed point of F and there exists an integer $k>0$ such that $[B(F)]^{(k)}=0$. Then the system globally converges to 0.

Definition 4.5
System (2) is said to be globally stable if it globally converges to a fixed point. In other words, it has a fixed point as the only attractor.

Example 4.6
Consider the follow system:

$$\begin{aligned} x_1(t+1) &= f_1(x_2(t),x_3(t)),\\ x_2(t+1) &= f_2(x_4(t)),\\ x_3(t+1) &= c_0,\\ x_4(t+1) &= f_4(x_3), \end{aligned} \tag{32}$$

where f_1, f_2, and f_3 can be any logical functions, and c_0 is a logical constant. Briefly, we denote (32) as:

$$X(t+1)=F(X(t)),\quad X\in\mathscr{D}^4.$$

The incidence matrix of F is

$$\mathscr{I}(F)=\begin{bmatrix}0&1&1&0\\0&0&0&1\\0&0&0&0\\0&0&1&0\end{bmatrix}.$$

It is easy to check that $[\mathscr{I}(F)]^{(4)}=0$, assume that 0 is a fixed point of the system (32), then it globally converges to 0.

Summarizing the above arguments, we have

Proposition 4.7
Consider system (2) (equivalently, (3)). Assume that $0\in\mathscr{D}^n$ is a fixed point of F and there exists an integer $k>0$ such that

$$[\mathscr{I}(F)]^{(k)}=0, \tag{33}$$

then 0 is the unique global attractor.

Note that if $x_e=(e_1,e_2,\ldots,e_n)$ is a fixed point of the system (2), the above method is still useful for testing whether x_e is a global attractor. Make a coordinate transformation

$$z_i=\begin{cases}x_i, & e_i=0,\\ \neg x_i, & e_i=1\end{cases} \tag{34}$$

Now it is easy to convert the system (2) into a system of z as:

$$z(t+1)=\tilde{F}(z(t)). \tag{35}$$

If there exists a $k>0$ such that $[\mathscr{I}(\tilde{F})]^{(k)}=0$, then x_e is a global attractor of the system (2).

It is easy to prove that [13] for a Boolean matrix $H\in\mathscr{B}_{n\times n}$ the followings are equivalent:

- there exists a $k>0$ such that $H^{(k)}=0$;
- there exists a permutation matrix P such that $P^{\mathrm{T}}\ltimes_B H\ltimes_B P$ is a strictly lower triangular (or upper triangular) matrix.

In fact, when $H=\mathscr{I}(F)$ is an incidence matrix, P means a re-ordering of variables.

Unfortunately, this method is sufficient but not necessary. Consider the following example

Example 4.8
Consider the following system:

$$\begin{aligned} x_1(t+1) &= x_1(t)\wedge x_2(t)\\ x_2(t+1) &= x_1(t)\wedge(\neg x_2(t)). \end{aligned} \tag{36}$$

It is easy to check that 0 is its unique attractor. But its incidence matrix is

$$\mathscr{I}(F)=\begin{bmatrix}1 & 1\\ 1 & 1\end{bmatrix},$$

and

$$[\mathscr{I}(F)]^{(k)}=\mathscr{I}(F)\neq 0,\quad k\geqslant 1.$$

Now what is the necessary and sufficient condition for a Boolean network to be globally convergent. In fact, we have the following conclusion.

Theorem 4.9
The Boolean network (2) is globally convergent, iff there exists a $k>0$ such that

$$\mathscr{I}(F^k)=0. \tag{37}$$

Proof (Necessity)
If the system is globally convergent, then after T_r steps (where T_r is the transient period [6]) all the states converge to the global attractor ξ. Hence, when $k\geqslant T_r$ (37) is true.

Sufficiency. Now assume that (37) is true. Then for any x we have $F^k(x)$ that is constant, say,

$$F^k(x)=\xi.$$

Then for any number $t\geqslant k$,

$$F^t(x)=F^k(F^{t-k}(x))=\xi.$$

Hence, the system converges to ξ globally. □

Remark 4.10

1. The Proposition 4.7 and the method right following it are practically useful because the size of the incidence matrix is $n\times n$, which is of the order of $O(n)$.
2. In Theorem 4.9, F^k is not directly computable. It can only be calculated by the algebraic form of F, say L_F, which is of size $2^n\times 2^n$. Hence, it is difficult to use it if n is not small.
3. From Theorem 4.3 it is clear that

$$\mathscr{I}(F^k)\leqslant[\mathscr{I}(F)]^{(k)},\quad k\geqslant 1. \tag{38}$$

But in general they are not equal.

DOI: 10.1002/rnc

Recall Proposition 4.7. In fact, the condition '0 is a fixed point' is not necessary for stability. Because from (38) one sees that condition (33) assures that F^s is constant for $s\geqslant k$. Say, $F^s(x)=\xi$, $\forall x$ and $s\geqslant k$. Then the system globally converges to ξ. We write it as a corollary.

Corollary 4.11
Consider the system (2). It is globally stable if the condition (33) holds.

Proposition 4.7 is one of the main tools for the stability analysis and stabilizer design. Hence, some further discussions are necessary.

First, we would like to point out that for a Boolean network its incidence matrix $\mathscr{I}(F)$ is coordinate-depending. The following example shows this.

Example 4.12
Consider the following system:

$$\begin{aligned}
x_1(t+1) &= [x_1(t)\wedge(x_2(t)\bar{\vee}x_3(x))]\vee(\neg x_1(t)\wedge x_3(t)),\\
x_2(t+1) &= [x_1(t)\wedge(\neg x_2(t))]\vee(\neg x_1\wedge x_2),\\
x_3(t+1) &= [x_1(t)\wedge(\neg(x_2(t)\wedge x_3(t)))]\vee[\neg x_1(t)\vee(x_2(t)\vee x_3(t))].
\end{aligned}\tag{39}$$

Briefly denote it as:

$$x(t+1)=F(x(t)).$$

It is easy to check that 0 is a fixed point of (39). For this system its incidence matrix is

$$\mathscr{I}(F)=\begin{bmatrix}1 & 1 & 1\\ 1 & 1 & 0\\ 1 & 1 & 1\end{bmatrix}.$$

There is no way to convert it into a strictly lower triangular form with reordering the variables. In algebraic form, it is easy to calculate that system (39) can be expressed as

$$x(t+1)=Lx(t),\tag{40}$$

where $x(t)=x_1(t)x_2(t)x_3(t)$,

$$L=\delta_8[8\ \ 3\ \ 1\ \ 5\ \ 1\ \ 5\ \ 3\ \ 8].$$

Now we consider a coordinate transformation as:

$$\begin{aligned}
z_1 &= [x_1\wedge\neg(x_3)]\vee[(\neg x_1)\wedge(x_2\bar{\vee}x_3)]\\
z_2 &= [x_1\wedge(x_2\bar{\vee}x_3)]\vee[(\neg x_1)\wedge x_3]\\
z_3 &= x_2.
\end{aligned}\tag{41}$$

In the vector form, we can easily calculated that

$$z=z_1z_2z_3=Tx,$$

where

$$T=\delta_8[7\ 1\ 6\ 4\ 5\ 3\ 2\ 8].$$

Then in z coordinate frame we have

$$z(t+1)=TLT^{\mathrm{T}}z(t):=\tilde{L}z(t),\tag{42}$$

where $\tilde{L}$ is

$$\tilde{L}=\delta_8[6\ 6\ 5\ 5\ 7\ 7\ 8\ 8].$$

A sequence of 2×2^n matrices, called the retrievers, have been defined in [9] as:

$$S_k^n=\delta_2[\underbrace{1\ldots 1}_{2^{n-k}}\underbrace{2\ldots 2}_{2^{n-k}}\ldots\underbrace{1\ldots 1}_{2^{n-k}}\underbrace{2\ldots 2}_{2^{n-k}}],\quad k=1,\ldots,n. \tag{43}$$

Using them, a mechanical procedure has been proposed in [9] to recover a system from the transition matrix of its algebraic form (42). Next, we recover the system from $\tilde{L}$.

Using retriever S_i^3, $i=1,2,3$, we have

$$\begin{aligned} z_1(t+1)&=M_1z(t),\\ z_2(t+1)&=M_2z(t),\\ z_3(t+1)&=M_3z(t), \end{aligned} \tag{44}$$

where

$$\begin{aligned} M_1&=S_1^3\tilde{L}=\delta_2[2\ 2\ 2\ 2\ 2\ 2\ 2\ 2],\\ M_2&=S_2^3\tilde{L}=\delta_2[1\ 1\ 1\ 1\ 2\ 2\ 2\ 2],\\ M_3&=S_3^3\tilde{L}=\delta_2[2\ 2\ 1\ 1\ 1\ 1\ 2\ 2]. \end{aligned}$$

It is easy to convert the matrix form (44) back to logic form, denoted by $z(t+1)=\tilde{F}(z(t))$, as:

$$\begin{aligned} z_1(t+1)&=0\\ z_2(t+1)&=z_1(t)\\ z_3(t+1)&=z_1(t)\bar{\vee}z_2(t). \end{aligned} \tag{45}$$

Now consider system (45) (i.e. system (39) under the coordinates Z), its incidence matrix is

$$\mathscr{I}(\tilde{F})=\begin{bmatrix}0&0&0\\1&0&0\\1&1&0\end{bmatrix},$$

which is strictly lower triangular. We conclude that system (39) globally converges to zero.

Example 4.12 shows that in some cases a coordinate change can help to find a nice incidence matrix to assure the global convergence.

Now a natural question is: If a network is globally stable, can we always find a coordinate transformation such that under new coordinate frame the system has a strict triangular form? Unfortunately, the answer is 'No'. Let us go back to Example 4.8. Since $n=2$ there are $(2^2)!=24$ coordinate transformations. We list them in the increasing order as:

$$T_1=I_2,\ \ T_2=\delta_4[1\ \ 2\ \ 4\ \ 3],\ \ T_3=\delta_4[1\ \ 3\ \ 2\ \ 4],\ldots T_{24}=\delta_4[4\ \ 3\ \ 2\ \ 1].$$

It follows that

$$T_2:\begin{cases}z_1=x_1\\ z_2=x_1\leftrightarrow x_2\end{cases}\Rightarrow\begin{cases}z_1(t+1)=z_1(t)\wedge z_2(t)\\ z_2(t+1)=\neg z_1(t).\end{cases}$$

$$T_3:\begin{cases}z_1=x_2\\ z_2=x_1\leftrightarrow x_2\end{cases}\Rightarrow\begin{cases}z_1(t+1)=z_1(t)\wedge z_2(t)\\ z_2(t+1)=(\neg z_1(t))\wedge z_2(t).\end{cases}$$

$$\vdots$$

$$T_{24}:\begin{cases}z_1=\neg x_1\\ z_2=x_2\end{cases}\Rightarrow\begin{cases}z_1(t+1)=z_1(t)\wedge z_2(t)\\ z_2(t+1)=z_2(t)\rightarrow z_1(t).\end{cases}$$

We have plenty of various forms. But unfortunately, no one has incidence matrix in strictly triangular form.

Hence, when the condition of Proposition 4.7 fails even under possible coordinate transformations, we have to invoke Theorem 4.9.

5. STABILIZATION OF BOOLEAN CONTROL NETWORKS

Consider the Boolean control network (6).

Definition 5.1
The global stabilization problem of system (6) is to find, if possible, $\{u(t)\}$ such that the system becomes globally convergent. If $u(t)=Wx(t)$, $t=1,2,\ldots,$ which is a set of logical functions, then the control is called the state feedback control.

Consider the following example.

Example 5.2
Consider the following system:

$$\begin{aligned} x_1(t+1) &= x_3(t)\vee u, \\ x_2(t+1) &= \neg x_1(t), \\ x_3(t+1) &= x_1(t)\leftrightarrow x_2(t), \end{aligned} \tag{46}$$

It is obvious that as long as we can delete $x_3(t)$ by using control u, the system is globally stable because the incidence matrix becomes strictly low triangular.

This is easy to be done. We may choose an open-loop control as $u(t)=1$, or closed-loop control $u(t)=\neg x_3(t)$.

To get a general design method, we first consider the expression of logical state variables. Let $x_1,\ldots,x_n$ be n logical state variables. There are two ways to express x_i:

(i) *Scalar Form*: $x_i\in\mathscr{D}$, $i=1,\ldots,n$. Putting them together, we denote

$$X=(x_1,\ldots,x_n)^{\mathrm{T}}\in\mathscr{D}^n.$$

(ii) *Vector Form*: $x_i\in\Delta$, $i=1,\ldots,n$. Multiplying them together, we denote

$$x=\ltimes_{i=1}^n x_i\in\Delta_{2^n}.$$

Define a set of vectors as:

$$s_k^n=[\underbrace{1\ldots1}_{2^{n-k}}\underbrace{0\ldots0}_{2^{n-k}}\ldots\underbrace{1\ldots1}_{2^{n-k}}\underbrace{0\ldots0}_{2^{n-k}}],\quad k=1,\ldots,n. \tag{47}$$

Remark 5.3

1. $s_k^n\in\mathscr{B}_{1\times 2^n}$, $k=1,\ldots,n$. Hence, the logical operators are applicable to them.
2. Comparing (47) with (43) one sees that s_k^n can be obtained from the first row of S_k^n by replacing 2 by 0.

Then we define a matrix as:

$$\mathscr{S}^n=\begin{bmatrix} s_1^n\\ s_2^n\\ \vdots\\ s_n^n\end{bmatrix}\in\mathscr{B}_{n\times 2^n}.$$

In fact, they are the first row of the corresponding retriever matrices [9].

The following proposition is easily verifiable.

Proposition 5.4

1. From scalar form to vector form, we have

$$x=[(x_1\leftrightarrow s_1^n)\wedge(x_2\leftrightarrow s_2^n)\wedge\cdots\wedge(x_n\leftrightarrow s_n^n)]^{\mathrm{T}} \quad \forall x_i\in\mathscr{D}. \tag{48}$$

2. From vector form to scalar form, we have

$$X=\mathscr{S}^n x. \tag{49}$$

Example 5.5
Let $x_s=(1,0,1,0)$. Then in vector form we have

$$\begin{aligned} x &= [1\leftrightarrow(1111111100000000)^{\mathrm{T}}]\wedge[0\leftrightarrow(1111000011110000)^{\mathrm{T}}]\\ &\quad \wedge[1\leftrightarrow(1100110011001100)^{\mathrm{T}}]\wedge[0\leftrightarrow(1010101010101010)^{\mathrm{T}}]\\ &=(1111111100000000)^{\mathrm{T}}\wedge(0000111100001111)^{\mathrm{T}}\wedge(1100110011001100)^{\mathrm{T}}\\ &\quad \wedge(0101010101010101)^{\mathrm{T}}\\ &=(0000010000000000)^{\mathrm{T}}. \end{aligned}$$

Let $X=\delta_{16}^9$. Then

$$x_s=\mathscr{S}^4 x_v=(0,1,1,1).$$

Next, we give a systematic analysis on the stabilizer design of Boolean control networks.

First, we define a mapping $\pi:\mathscr{B}_{2^n\times 2^n}\to\mathscr{B}_{n\times n}$ as

$$\begin{aligned} \pi(L) = {}&[[(\mathscr{S}^n L)\bar{\vee}(\mathscr{S}^n L M_n)]\ltimes_B \mathbf{1}_{2^n}, [(\mathscr{S}^n L)\bar{\vee}(\mathscr{S}^n L)(I_2\otimes M_n)]\ltimes_B \mathbf{1}_{2^n},\\ &\ldots, [(\mathscr{S}^n L)\bar{\vee}(\mathscr{S}^n L)(I_{2^{n-1}}\otimes M_n)]\ltimes_B \mathbf{1}_{2^n}\ltimes_B \mathbf{1}_{2^n}], \quad L\in\mathscr{B}_{2^n\times 2^n}. \end{aligned} \tag{50}$$

where M_n is the structure matrix of negation (refer to Table I).

Then we have the following result about how to build the incidence matrix from L.

Theorem 5.6
Consider the Boolean network (2) (equivalently, (3)) with its algebraic form (5). The incidence matrix of F can be obtained from L by the following formula:

$$\mathscr{I}(F)=\pi(L). \tag{51}$$

Proof
From the construction of $\mathscr{S}^n$ it is easy to see that $L_s:=\mathscr{S}^n L$ is the structure matrix of the system, resulting in scalar form. While $L_s M_n$ is the structure matrix with x_1 being replaced by $\neg x_1$. If at the ith row they are the same, it means f_i is independent of x_1. Then the ith row of $[(\mathscr{S}^n L)\bar{\vee}(\mathscr{S}^n L M_n)]$ will be identically zero. Hence the ith element of

Table I. Structure matrices of some logical operators.

Operator	Structure matrix	Operator	Structure matrix
$\neg$	$M_n=\delta_2[2\ \ 1]$	$\vee$	$M_d=\delta_2[1\ \ 1\ \ 1\ \ 2]$
$\rightarrow$	$M_i=\delta_2[1\ \ 2\ \ 1\ \ 1]$	$\leftrightarrow$	$M_e=\delta_2[1\ \ 2\ \ 2\ \ 1]$
$\wedge$	$M_c=\delta_2[1\ \ 2\ \ 2\ \ 2]$	$\bar{\vee}$	$M_p=\delta_2[2\ \ 1\ \ 1\ \ 2]$

DOI: 10.1002/rnc

$[(\mathscr{S}^n L)\bar{\vee}(\mathscr{S}^n LM_n)]\ltimes_B \mathbf{1}_{2^n}\ltimes_B \mathbf{1}_{2^n}$ is zero. Otherwise, at least one element in this row is 1, and hence the ith element of $[(\mathscr{S}^n L)\bar{\vee}(\mathscr{S}^n LM_n)]\ltimes_B \mathbf{1}_{2^n}\ltimes_B \mathbf{1}_{2^n}$ is one.

Same argument is applicable to other variables. The only difference is, the negation structure matrix needs to be moved from the front of x_i to the front of all variables. Then (51) follows. □

We give an example to depict it.

Example 5.7
Assume that system (2) has its network transition matrix L as:

$$L=\delta_{16}[1\ 9\ 9\ 13\ 4\ 12\ 12\ 16\ 2\ 1\ 10\ 14\ 1\ 9\ 9\ 13]. \tag{52}$$

A simple routine can calculate that

$$\mathscr{I}(F)=\pi(L)=\begin{bmatrix}0&0&1&1\\0&0&1&1\\1&1&0&0\\1&1&0&0\end{bmatrix}$$

In fact, using the mechanical procedure provided in [9], we can uniquely recover the system from L as:

$$\begin{aligned}x_1(t+1)&=x_3(t)\wedge x_4(t),\\x_2(t+1)&=x_3(t)\vee x_4(t),\\x_3(t+1)&=x_1(t)\rightarrow x_2(t),\\x_4(t+1)&=x_1(t)\leftrightarrow x_2(t).\end{aligned} \tag{53}$$

This verifies the obtained $\mathscr{I}(F)$.

Next, we consider the stabilization problem. Consider system (6) with its algebraic form (8).

Using Theorem 5.6 and Corollary 4.11, a sufficient condition for stabilization with open-loop control is

Lemma 5.8
System (6) is stabilizable by an open-loop control u, if $\pi(Lu)$ has a strictly lower (or upper) triangular form.

Note that since the incidence matrix is coordinate depended, coordinate transformations have to be taken into consideration.

Using formula (14), we have the following

Theorem 5.9
System (6) (or its algebraic form (8)) is stabilizable by an open-loop control u, if there is a coordinate transformation $z=Tx$ such that $\pi(TL(I_{2^m}\otimes T^{\mathrm{T}})u)$ has a strictly lower triangular form.

Note that both possible values of u and coordinate transformations are finite (precisely, 2^m and $(2^n)!$ respectively), hence, theoretically both Lemma 5.8 and Theorem 5.9 are verifiable.

Example 5.10
Consider the following system:

$$\begin{aligned}x_1(t+1)&=\neg x_2(t),\\x_2(t+1)&=\neg x_4(t)\leftrightarrow((x_4(t)\wedge(x_2(t)\bar{\vee}x_3(t)))\vee u(t)),\\x_3(t+1)&=\neg((x_4(t)\wedge(x_2(t)\bar{\vee}x_3(t)))\vee u(t)),\\x_4(t+1)&=(x_4(t)\vee(x_2(t)\bar{\vee}x_3(t)))\wedge u(t).\end{aligned} \tag{54}$$

After ‘try and error’ for simplifying the system, we use the following coordinate transformation:

$$\begin{aligned} z_1 &= x_4, \\ z_2 &= x_2 \bar{\vee} x_3, \\ z_3 &= \neg x_3, \\ z_4 &= \neg x_1. \end{aligned} \tag{55}$$

Its inverse can be easily calculated as:

$$\begin{aligned} x_1 &= \neg z_4, \\ x_2 &= z_2 \leftrightarrow z_3, \\ x_3 &= \neg z_3, \\ x_4 &= z_1. \end{aligned} \tag{56}$$

Then the system becomes

$$\begin{aligned} z_1(t+1) &= (z_1(t) \vee z_2(t)) \wedge u(t), \\ z_2(t+1) &= \neg z_1(t), \\ z_3(t+1) &= (z_1(t) \wedge z_2(t)) \vee u(t), \\ z_4(t+1) &= z_2(t) \leftrightarrow z_3(t). \end{aligned} \tag{57}$$

Now it is clear that if we choose

$$u(t)=0, \tag{58}$$

the system becomes

$$\begin{aligned} z_1(t+1) &= 0, \\ z_2(t+1) &= \neg z_1(t), \\ z_3(t+1) &= (z_1(t) \wedge z_2(t)), \\ z_4(t+1) &= z_2(t) \leftrightarrow z_3(t). \end{aligned} \tag{59}$$

It is obvious that the incidence matrix of system (59) is

$$\mathscr{I}(F)=\begin{bmatrix} 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 \end{bmatrix},$$

which is of the strictly lower triangular form. We conclude that the constant control $u(t)=0$ stabilizes system (54).

Next, we consider the closed-loop control. Let $u(t)$ be a set of logical functions of $x(t)$. Then we can always express it as

$$u(t)=Gx(t), \tag{60}$$

where $G \in \mathscr{L}_{2^m \times 2^n}$. Plugging it into (8) yields

$$x(t+1)=LGx^2(t)=LG\Phi_n x(t), \tag{61}$$

DOI: 10.1002/rnc

where Φ_n is defined in [6] as

$$\Phi_n=\prod_{i=1}^{n} I_{2^{n-1}}\otimes[(I_2\otimes W_{[2,2^{n-i}]})M_r], \tag{62}$$

with $M_r=\delta_4[14]$.

Now the following result is obvious.

Theorem 5.11
System (4) is stabilizable by a closed-loop control $u=Gx$, if $\pi(LG\Phi_n)$ has a strictly lower (or upper) triangular form. Moreover, if there exists a coordinate transformation $z=Tx$, such that $\pi(TLG\Phi_n T^{\mathrm{T}})$ has a strictly lower (or upper) triangular form, then the control also stabilizes the system.

Example 5.12
Consider the system

$$\begin{aligned}
x_1(t+1)&=\neg x_2(t),\\
x_2(t+1)&=\neg x_4(t)\leftrightarrow((x_4(t)\wedge(x_2(t)\bar{\vee}x_3(t)))\vee u(t)),\\
x_3(t+1)&=\neg((x_4(t)\wedge(x_2(t)\bar{\vee}x_3(t)))\vee u(t)),\\
x_4(t+1)&=(x_4(t)\vee(x_2(t)\bar{\vee}x_3(t)))\vee u(t).
\end{aligned} \tag{63}$$

In fact, it is obtained from (53) by changing the ways of inputs. Using the same coordinate transformation as in Example 5.10, we have

$$\begin{aligned}
z_1(t+1)&=(z_1(t)\vee z_2(t))\vee u(t),\\
z_2(t+1)&=\neg z_1(t),\\
z_3(t+1)&=(z_1(t)\wedge z_2(t))\wedge u(t),\\
z_4(t+1)&=z_2(t)\leftrightarrow z_3(t).
\end{aligned} \tag{64}$$

One can check that constant (open-loop) controls cannot stabilize the system. If we use a closed-loop control

$$u(t)=\neg z_1(t)\wedge\neg z_2(t),$$

the system becomes

$$\begin{aligned}
z_1(t+1)&=1,\\
z_2(t+1)&=\neg z_1(t),\\
z_3(t+1)&=0,\\
z_4(t+1)&=z_2(t)\leftrightarrow z_3(t).
\end{aligned} \tag{65}$$

Obviously, it is globally stable. Converting the control back to the original coordinate frame, we conclude that

$$u(t)=\neg x_4(t)\wedge\neg(x_2(t)\bar{\vee}x_3(t))$$

stabilizes system (63).

The advantage of using metric-based analysis is that the size of the involved matrix is small. The disadvantage is that the condition is only a sufficient one. Next, we search for necessary and sufficient condition.

As discussed in Section 2, the global stability of a free boolean network is equivalent to that where a $k>0$ exists such that $F^k=$constant. In vector form it is equivalent to L^k having equal columns, which is the global attractor. An $s\times s$ logic matrix, M, is said to be a matrix of constant mapping if there exists a δ_s^j, such that

$$\mathrm{Col}(M)=\{\delta_s^j\}.$$

Now consider the stabilization by a constant control u. Then the control-dependent transition matrix is Lu.

Using the properties of STP, it is easy to calculate that

$$(Lu)^k = L[(I_{2^m}\otimes L)\Phi_m]^{m-1}u. \tag{66}$$

Since k should be less than or equal to the transient time, i.e. $k\leqslant T_r\leqslant 2^n$, we can get an easily verifiable necessary and sufficient condition as follows: Note that $L[(I_{2^m}\otimes L)\Phi_m]^{m-1}$ is a $2^n\times 2^{n+m}$ matrix. We split it into 2^m square blocks as:

$$L[(I_{2^m}\otimes L)\Phi_m]^{m-1} := [L_1^k L_2^k \ldots L_{2^m}^k]. \tag{67}$$

Using this notation and according to the above argument, we have the following necessary and sufficient condition.

Theorem 5.13
System (4) is stabilizable by a constant control u, iff there exists a matrix of constant mapping

$$L_j^k,\quad 1\leqslant k\leqslant 2^n,\ \ 1\leqslant j\leqslant 2^m.$$

Moreover, corresponding to each matrix of constant mapping L_j^k the stabilizing control is $u=\delta_{2^m}^j$.

The following example illustrates this result.

Example 5.14
Consider the following system:

$$\begin{aligned} x_1(t+1) &= (x_1(t)\vee x_2(t))\wedge u\\ x_2(t+1) &= (x_2(t)\wedge u)\rightarrow x_1. \end{aligned} \tag{68}$$

It is easy to calculate that

$$L=\delta_4[1\ 1\ 2\ 3\ 3\ 3\ 3\ 3].$$

Since $\Phi_1 = M_r$, according to Theorem 5.13 we have to calculate

$$L[(I_2\otimes L)M_r]^k,\quad k\geqslant 1$$

to see if we can find a constant mapping block. In fact when $k=2$ we have

$$L[(I_2\otimes L)M_r]^2=\delta_4[1\ \ 1\ \ 1\ \ 1\ \ 3\ \ 3\ \ 3\ \ 3].$$

We conclude that if we use control $u=1$, the system is stabilized at $x=\delta_4^1$ (i.e. $x_1=1$ and $x_2=1$), when $u=0$ the system is stabilized at $x=\delta_4^3$ (i.e. $x_1=0$, $x_2=1$).

Consider the state feedback control as in (60). Using the expression (61) and the above argument, the following result is obvious.

Theorem 5.15
System (4) is stabilizable by a closed-loop control $u=Gx$, iff there exists a $2^m\times 2^n$ logical matrix G and an integer $1\leqslant k\leqslant 2^n$ such that $(LG\Phi_n)^k$ is a matrix of constant mapping.

We give an example.

DOI: 10.1002/rnc

Example 5.16
Consider the following system:

$$\begin{aligned}
x_1(t+1) &= [x_2(t)\vee(\neg x_2(t)\wedge(x_3(t)\vee x_4(t)))]\wedge u\\
x_2(t+1) &= (x_2(t)\wedge(x_3(t)\vee x_4(t)))\vee[x_1(t)\wedge(\neg x_2(t)\wedge\neg(x_3(t)\vee x_4(t)))]\\
x_3(t+1) &= (x_1(t)\wedge(x_3(t)\leftrightarrow x_4(t)))\vee[\neg x_1(t)\wedge((x_2(t)\\
&\qquad \wedge(x_3(t)\leftrightarrow x_4(t)))\wedge(\neg x_2(t)\wedge(x_3(t)\wedge x_4(t))))]\\
x_4(t+1) &= x_1(t)\wedge\neg x_4(t)\vee[\neg x_1(t)\wedge((x_2(t)\wedge\neg x_4(t))\vee(\neg x_2(t)\wedge\neg(x_3(t)\rightarrow x_4(t))))].
\end{aligned}\tag{69}$$

It is easy to verify that if we choose

$$G=\begin{bmatrix}1 & 1 & 0 & 0\\ 0 & 0 & 1 & 1\end{bmatrix}^3$$

then

$$(LG\Phi_4)^{14}=\delta_{16}[\underbrace{16\ldots16}_{16}].$$

Note that

$$u(t)=Gx(t)=x_1(t),$$

which globally stabilizes system (69) to $x_1=0$, $x_2=0$, $x_3=0$, $x_4=0$.

Next, we briefly discuss the case when the system is required to converge to a particular state x_0. In this case the problem is slightly simpler. In addition to above stability requirements, we need to assure that x_0 is a fixed point of the control system. We may write it down as a corollary.

Corollary 5.17

1. System (4) is globally stabilized to x_0 by a constant control u, iff u satisfies

$$Lux_0=x_0,\tag{70}$$

   and there exists an integer $k>0$, such that $(Lu)^k$ is a constant mapping.
2. System (4) is globally stabilized to x_0 by a constant control $u=Gx$, iff G satisfies

$$LG\Phi_n x_0=x_0,\tag{71}$$

   and there exists an integer $k>0$, such that $(LG\Phi_n)^k$ is a constant mapping.

Finally, we consider the stabilization by an open-loop control, $u(t)=\ltimes_{i=1}^m u_i(t)$, $t=1,2,\ldots$. Assume we want to stabilize it to x_0.

First of all, it is obvious that a necessary condition is, there is a control $u_e\in\Delta_{2^m}$ such that

$$Lu_e x_0=x_0.\tag{72}$$

Second, note that

$$x(t_0+k+1)=Lu(t_0+k)Lu(t_0+k-1)\ldots Lu(t_0+1)x(t_0).$$

To make all trajectories converge to x_0, there must be a $k>0$ such that

$$Lu(k)Lu(k-1)\ldots Lu(1)x\equiv x_0\quad \forall x\in\Delta_{2^n}.$$

This is equivalent to

$$\mathrm{Col}(Lu(k)Lu(k-1)\ldots Lu(1))=\{x_0\}.\tag{73}$$

Observe that

$$\begin{aligned} Lu(k)Lu(k-1)\ldots Lu(1) &= L(I_{2^m}\otimes L)(I_{2^{2m}}\otimes L)\cdots(I_{2^{(k-1)m}}\otimes L)\ltimes_{i=k}^{1}u(i)\\ &:=[L_1^k, L_2^k, \ldots, L_{2^{km}}^k]\ltimes_{i=k}^{1}u(i). \end{aligned} \tag{74}$$

Now it is clear that if there is a $1\leqslant j\leqslant 2^{km}$ such that L_j^k corresponds to the constant mapping $\psi(x)\equiv x_0$, we can choose the control

$$\ltimes_{i=k}^{1}u(i)=\delta_{2^{mn}}^{j},$$

such that (73) holds.

Summarizing the above arguments, we have

Theorem 5.18
System (6) is globally stabilized to x_0 by an open-loop control $u(t), t=1,2,\ldots,$ iff

(i) there are an integer $k>0$ and an L_j^k, $1\leqslant j\leqslant 2^{km}$, such that

$$\mathrm{Col}(L_j^k)=\{x_0\};$$

(ii) there is a $u_e\in\Delta_{2^m}$ such that (72) holds.

We give an example for this.

Example 5.19
Consider the following system:

$$\begin{aligned} x_1(t+1)&=x_1(t)\vee u_1(t),\\ x_2(t+1)&=(x_2(t)\vee x_3(t))\leftrightarrow u_1(t),\\ x_3(t+1)&=(u_1(t)\rightarrow x_2(t))\vee x_3(t),\\ x_4(t+1)&=(x_3(t)\wedge u_2(t))\rightarrow x_4(t). \end{aligned} \tag{75}$$

Set $x(t)=\ltimes_{i=1}^{4}x_i(t)$ and $u(t)=\ltimes_{i=1}^{2}u_i(t)$. Using vector form, (75) can be expressed as

$$x(t+1)=Lu(t)x(t), \tag{76}$$

where

$$\begin{array}{rcccccccccccccccc} L=\delta_{16}[1 & 2 & 1 & 1 & 1 & 2 & 7 & 7 & 1 & 2 & 1 & 1 & 1 & 2 & 7 & 7\\ 1 & 1 & 1 & 1 & 1 & 1 & 7 & 7 & 1 & 1 & 1 & 1 & 1 & 1 & 7 & 7\\ 5 & 6 & 5 & 5 & 5 & 6 & 1 & 1 & 13 & 14 & 13 & 13 & 13 & 14 & 9 & 9\\ 5 & 5 & 5 & 5 & 5 & 5 & 1 & 1 & 13 & 13 & 13 & 13 & 13 & 13 & 9 & 9]. \end{array}$$

According to (74), we calculate

$$L(I_{2^2}\otimes L)(I_{2^{2\times 2}}\otimes L)\ldots(I_{2^{2(k-1)}}\otimes L)$$

to see whether we can find a constant mapping block. In fact when $k=2$ we have

$$\begin{aligned} M &= L(I_{2^2}\otimes L)\\ &:=[M_1, M_2, \ldots, M_{16}], \end{aligned}$$

DOI: 10.1002/rnc

where

$$M=\delta_{16}[\begin{array}{cccccccccccccccc}
1 & 2 & 1 & 1 & 1 & 2 & 7 & 7 & 1 & 2 & 1 & 1 & 1 & 2 & 7 & 7\\
1 & 1 & 1 & 1 & 1 & 1 & 7 & 7 & 1 & 1 & 1 & 1 & 1 & 1 & 7 & 7\\
1 & 2 & 1 & 1 & 1 & 2 & 1 & 1 & 1 & 2 & 1 & 1 & 1 & 2 & 1 & 1\\
1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1\\
1 & 1 & 1 & 1 & 1 & 1 & 7 & 7 & 1 & 1 & 1 & 1 & 1 & 1 & 7 & 7\\
1 & 1 & 1 & 1 & 1 & 1 & 7 & 7 & 1 & 1 & 1 & 1 & 1 & 1 & 7 & 7\\
1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1\\
1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1\\
5 & 6 & 5 & 5 & 5 & 6 & 1 & 1 & 5 & 6 & 5 & 5 & 5 & 6 & 1 & 1\\
5 & 5 & 5 & 5 & 5 & 5 & 1 & 1 & 5 & 5 & 5 & 5 & 5 & 5 & 1 & 1\\
5 & 6 & 5 & 5 & 5 & 6 & 5 & 5 & 13 & 14 & 13 & 13 & 13 & 14 & 13 & 13\\
5 & 5 & 5 & 5 & 5 & 5 & 5 & 5 & 13 & 13 & 13 & 13 & 13 & 13 & 13 & 13\\
5 & 5 & 5 & 5 & 5 & 5 & 1 & 1 & 5 & 5 & 5 & 5 & 5 & 5 & 1 & 1\\
5 & 5 & 5 & 5 & 5 & 5 & 1 & 1 & 5 & 5 & 5 & 5 & 5 & 5 & 1 & 1\\
5 & 5 & 5 & 5 & 5 & 5 & 5 & 5 & 13 & 13 & 13 & 13 & 13 & 13 & 13 & 13\\
5 & 5 & 5 & 5 & 5 & 5 & 5 & 5 & 13 & 13 & 13 & 13 & 13 & 13 & 13 & 13
\end{array}], \tag{77}$$

and $M_i \in \mathscr{L}_{16\times 16}, i=1,\ldots,16$. From (77), we know that

$$M_4=M_7=M_8=\delta_{16}[1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1].$$

Equivalently,

$$\mathrm{Col}(M_4)=\mathrm{Col}(M_7)=\mathrm{Col}(M_8)=\{\delta_{16}^1\}.$$

On the other hand, choosing $u_e=\delta_4^1$ (or $u_e=\delta_4^2$), we have

$$Lu_e\delta_{16}^1=\delta_{16}^1.$$

From Theorem 5.18, system (75) is globally stabilized to $x_0=\delta_{16}^1\sim(1\ \ 1\ \ 1\ \ 1)$ by an open-loop control $u(t)$ (or $\bar{u}(t)$ and $\tilde{u}(t)$), where

$$u(t)=\begin{cases}\delta_4^4\sim(00), & t=1\\ \delta_4^1\sim(11), & t=2\\ u_e, & t\geqslant 3,\end{cases} \quad \text{and} \quad \bar{u}(t)=\begin{cases}\delta_4^3\sim(01), & t=1\\ \delta_4^2\sim(10), & t=2\\ u_e, & t\geqslant 3,\end{cases} \quad \text{and} \quad \tilde{u}(t)=\begin{cases}\delta_4^4\sim(00), & t=1\\ \delta_4^2\sim(10), & t=2\\ u_e, & t\geqslant 3.\end{cases}$$

Before ending this section we should like to discuss briefly about the conditions in Theorem 5.18. The condition (i) says that all the trajectories can reach the preassigned fixed point x_0. One may doubt whether the condition (ii) is necessary. In fact, condition (ii) means that x_0 is an equilibrium state under certain control. The following example shows that condition (ii) is also necessary.

Example 5.20
Consider the following system:

$$\begin{aligned} x_1(t+1) &= \neg(x_1(t)\wedge u(t)),\\ x_2(t+1) &= (u(t)\wedge(x_2(t)\to x_1(t)))\vee(\neg u(t)\wedge(x_1(t)\leftrightarrow x_2(t))). \end{aligned} \tag{78}$$

Set $x(t) = \ltimes_{i=1}^{2} x_i(t)$. Using the vector form, (75) can be expressed as:

$$\begin{aligned} x(t+1) &= Lu(t)x(t) \\ &= \delta_4[3\ 3\ 2\ 1\ 1\ 2\ 2\ 1]u(t)x(t). \end{aligned} \tag{79}$$

For any initial state $\xi \in \Delta_4$, if we choose $u(1) = \delta_2^2$, then

$$\begin{aligned} x(2) &= Lu(1)\xi \\ &= \delta_4[1\ 2\ 2\ 1]\xi. \end{aligned}$$

Next, choose $u(2) = \delta_2^1$, then

$$\begin{aligned} x(3) &= Lu(2)x(2) \\ &= (\delta_4[3\ 3\ 2\ 1])(\delta_4[1\ 2\ 2\ 1])\xi \\ &= \delta_4[3\ 3\ 3\ 3]\xi \\ &= \delta_4^3 \quad \forall \xi \in \Delta_4. \end{aligned}$$

From (76), it is obvious that there does not exist u_e, such that

$$Lu_e\delta_4^3 = \delta_4^3.$$

In step 3, no matter what is the value of $u(3)$, ($u(3) = \delta_2^1$ or $u(3) = \delta_2^2$), the dynamic of the Boolean network will leave the state $x_0 = \delta_4^3$. Hence, system (75) cannot be globally stabilized to x_0 by an open-loop control $u(t), t = 1, 2, \ldots$.

6. CONCLUSION

The stability of Boolean networks and the stabilization of Boolean control networks were investigated in this paper. The STP of matrices and the matrix expression of logic provided a powerful tool for this work. The main results consist of two parts. (i) The known results of convergence provided in [13] have been improved by using logical coordinate transformations. Then it has been used to the design of stabilizers for Boolean control networks. (ii) Based on the algebraic form of Boolean (control) networks, necessary and sufficient conditions for stability of Boolean networks were obtained. Then the stabilization of Boolean control networks by constant controls, open-loop controls, and closed-loop controls were investigated and the respective necessary and sufficient conditions was obtained. The stabilizer design techniques were also provided.

Though the results in part (i) are only sufficient, their main advantage is that the related computations are less complicated because the incidence matrices used there are of much smaller dimensions ($n \times n$). The weakness of part (ii) is the computation complexity because they use transition matrices, which are of much larger dimensions ($2^n \times 2^n$).

ACKNOWLEDGEMENTS

An STP toolbox is provided in `http://lsc.amss.ac.cn/~dcheng` for the related computations.

REFERENCES

1. Kauffman SA. *The Origins of Order*: *Self-organization and Selection in Evolution*. Oxford University Press: New York, 1993.
2. Davison EH *et al.* A genomic regulatory network for development. *Science* 2002; **295**:1669–1678.
3. Akutsu T, Hayashida M, Ching W, Ng MK. Control of boolean networks: hardness results and algorithms for tree structured networks. *Journal of Theoretical Biology* 2007; **244**:670–679.

DOI: 10.1002/rnc

4. Huang S, Ingber I. Shape-dependent control of cell growth, differentiation, and apotosis: switching between attractors in cell regulatory networks. *Experimental Cell Research* 2000; **261**:91–103.
5. Cheng D. Semi-tensor product of matrices and its applications—a survey. *ICCM 2007*, vol. 3, 2007; 641–668.
6. Cheng D, Qi H. A linear representation of dynamics of Boolean networks. *IEEE Transactions on Automatic Control* 2010; **55**(10). DOI: 10.1109/TAC.2010.2043294.
7. Cheng D. Input-state approach to boolean networks. *IEEE Transactions on Neural Networks* 2009; **20**(3):512–521.
8. Kauffman SA. *At Home in the Universe*. Oxford University Press: New York, 1995.
9. Cheng D, Qi H. Controllability and observability of boolean networks. *Automatica* 2009; **45**(7):1659–1667.
10. Cheng D, Li Z, Qi H. Realization of Boolean control networks. *Automatica* 2010; **46**(1):62–69.
11. Kauffman SA. Metabolic stability and epigenesis in randomly constructed genetic nets. *Journal of Theoretical Biology* 1969; **22**:437–467.
12. Onami S, Kyoda KM, Morohashi M, Kitano H. The DBRF method for inferring a gene network from large-scale steady-state gene expression date. In *Foundations of Systems Biology*, Kitano H (ed.). MIT Press: Cambridge, 2001; 59–75.
13. Robert F. *Discrete Iterations*, *A Metric Study*. Translated by J. Rokne. Springer: Berlin, 1986.
14. Rade L, Westergren B. *Mathematics Handbook for Science and Engineering* (4th edn). Studentlitteratur: Sweden, 1998.

DOI: 10.1002/rnc