Center of STP Theory and Its Applications
LiaoCheng University, LiaoCheng, Shandong, P.R. China

Stability, Stabilization and Controllability of Boolean networks

Jianquan Lu (Southeast University)

jqluma@seu.edu.cn

August 15-23, 2020

非常感谢四位同学帮忙准备PPT！

朱诗勇、丁雪莹、孙靓洁、李博文

Thank you very much for preparing the PPT!

Shiyong Zhu、Xueying Ding、Liangjie Sun、Bowen Li

IET Control Theory & Applications

Special Issue: Recent Developments in Logical Networks and its Applications

IET Journals
The Institution of Engineering and Technology

Survey on semi-tensor product method with its applications in logical networks and other finite-valued systems

ISSN 1751-8644
Received on 22nd January 2017
Revised 22nd March 2017
Accepted on 28th April 2017
E-First on 25th May 2017
doi: 10.1049/iet-cta.2016.1659
www.ietdl.org

Jianquan Lu[1] ✉, Haitao Li[2], Yang Liu[1,3], Fangfei Li[4]

[1]*School of Mathematics, Southeast University, Nanjing 210096, People's Republic of China*
[2]*School of Mathematics and Statistics, Shandong Normal University, Jinan 250014, People's Republic of China*
[3]*Department of Mathematics, Zhejiang Normal University, Jinhua 321004, People's Republic of China*
[4]*Department of Mathematics, East China University of Science and Technology, Shanghai 200237, People's Republic of China*
✉ *E-mail: jqluma@seu.edu.cn*

This is also a survey introduction about stability, stabilization and controllability!

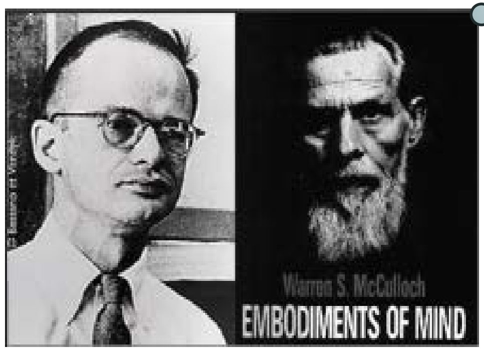
Boolean networks

- ✓ **The purpose of BNs**
- ✓ **Model description of BNs**

The propose of Boolean networks

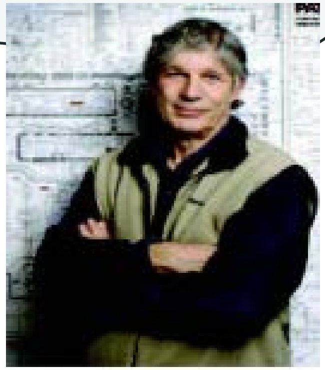
The relations among neural activities in brain can be treated by means of propositional logic [1].

The regulatory genes in every cell can active or inactive other genes like a switch [2].



Mcculloch & Pitts

Jacob & Monod



Kauffman

1943
M-P model of neuron

1961
Genetic regulatory networks
(**be awarded the Nobel Prize**)

1969
Boolean networks [3]

[1] W. S. McCulloch and W. Pitts, A logical calculus of the ideas immanent in nervous activity, The Bulletin of Mathematical Biophysics, 5(4):115-133, 1943.
[2] F. Jacob and J. Monod, Genetic regulatory mechanisms in the synthesis of proteins, Journal of molecular biology, 3(3):318-356,1961.
[3] S.A. Kauffman, Metabolic stability and epigenesis in randomly constructed genetic nets, Journal of Theoretical Biology, 22(3):437-467, 1969.

Model description of Boolean networks

Boolean networks (BNs) [3]

$$\begin{cases} x_1(t+1) = f_1(x_1(t), \ldots, x_n(t)), \\ x_2(t+1) = f_2(x_1(t), \ldots, x_n(t)), \\ \vdots \\ x_n(t+1) = f_n(x_1(t), \ldots, x_n(t)), \end{cases}$$

The feature of BNs:

- **Discrete-time models**
- **State** $x_i(t)$: ON (1) or OFF (0)
- f_i: Boolean function

External control inputs

Boolean control networks (BCNs) [4]

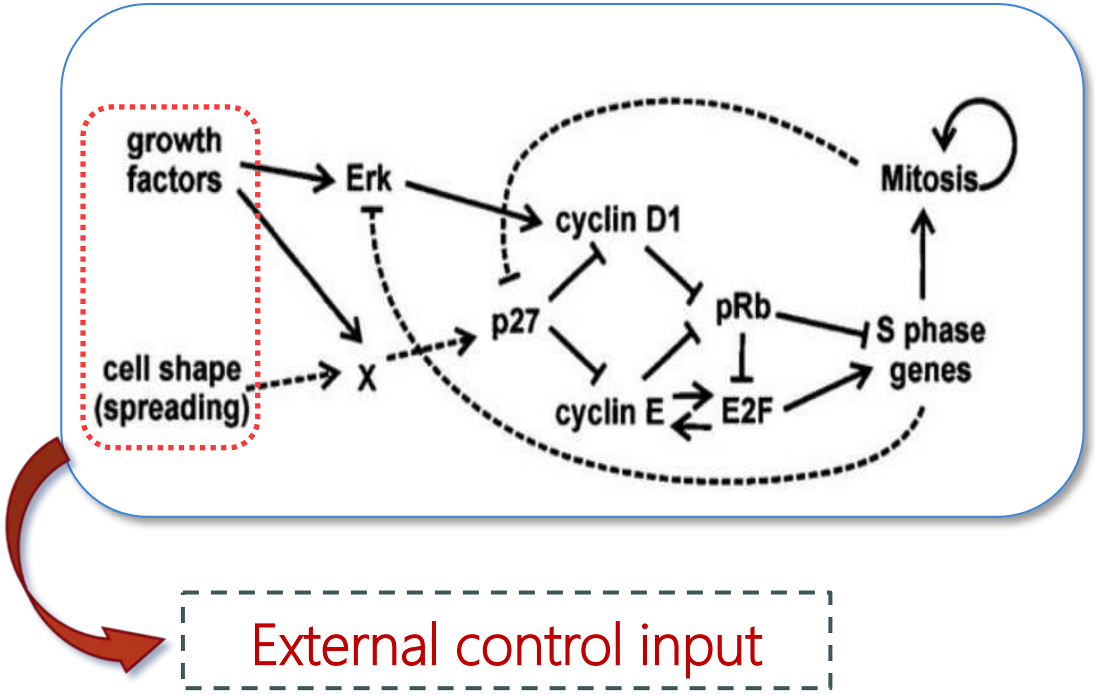
$$\begin{cases} x_1(t+1) = f_1'(x_1(t), \ldots, x_n(t), u_1(t), \ldots, u_m(t)), \\ x_2(t+1) = f_2'(x_1(t), \ldots, x_n(t), u_1(t), \ldots, u_m(t)), \\ \vdots \\ x_n(t+1) = f_n'(x_1(t), \ldots, x_n(t), u_1(t), \ldots, u_m(t)), \end{cases}$$

The control inputs $u_i(t)$ are added.

[3] S.A. Kauffman, Metabolic stability and epigenesis in randomly constructed genetic nets, Journal of Theoretical Biology, 22(3):437-467, 1969.
[4] T. Akutsu, *et al.*, Control of Boolean networks: Hardness results and algorithms for tree structured networks, Journal of Theoretical Biology, 244(4): 670–679, 2007.

Model description of Boolean networks

A simple BCN: The signaling system within capillary endothelial cells [5]



$$
\begin{cases}
Erk(t+1) & = \neg(GF(t) \rightarrow S(t)), \\
D1(t+1) & = \neg(Erk(t) \rightarrow p27(t)), \\
p27(t+1) & = M(t) \rightarrow X(t), \\
E(t+1) & = E2F(t) \rightarrow p27(t), \\
E2F(t+1) & = \neg(E(t) \rightarrow pRb(t)), \\
pRb(t+1) & = \neg(D1(t) \wedge E(t)), \\
S(t+1) & = \neg(pRb(t) \rightarrow E2F(t)), \\
M(t+1) & = \neg(M(t) \rightarrow S(T)), \\
X(t+1) & = GF(t) \wedge CS(t),
\end{cases}
$$

[5] S. Huang and D.E. Ingber, Shape-dependent control of cell growth, differentiation, and apoptosis: switching between attractors in cell regulatory networks, Experimental Cell Research, 261(1):91-103, 2000.

Semi-tensor Product

- ✓ **Definition of STP**
- ✓ **Some Applications**
- ✓ **Algebraic Expression of BNs**

Definition of STP

When m $\neq p$, $A_{n\times m} \times B_{p\times q}$ =?

A generalization of conventional matrix product

Definition 1 [6]**:** Given two matrices $A \in \mathbb{R}^{n\times m}$ and $B \in \mathbb{R}^{p\times q}$, the STP of A and B, denoted by $A \ltimes B$, is defined as:

$$A \ltimes B = (A \otimes I_{l/m})(B \otimes I_{l/p}).$$

Here, l is the least common multiple of m and p, $\otimes$ is Kronecker product of matrices.

[6] D. Cheng, *et al*. Analysis and Control of Boolean Networks: A Semi-tensor Product Approach, Springer, 2010.

Definition of STP

A calculating example:

Let $A = \begin{bmatrix} 1 & -2 \\ 2 & -1 \end{bmatrix}$ and $B = \begin{bmatrix} 1 & 2 \\ 2 & 1 \\ 2 & 3 \\ 3 & 2 \end{bmatrix}$, then we have

$$A \ltimes B = \begin{bmatrix} 1 \times \begin{bmatrix} 1 \\ 2 \end{bmatrix} + (-2) \times \begin{bmatrix} 2 \\ 3 \end{bmatrix}, 1 \times \begin{bmatrix} 2 \\ 1 \end{bmatrix} + (-2) \times \begin{bmatrix} 3 \\ 2 \end{bmatrix} \\ 2 \times \begin{bmatrix} 1 \\ 2 \end{bmatrix} + (-1) \times \begin{bmatrix} 2 \\ 3 \end{bmatrix}, 2 \times \begin{bmatrix} 2 \\ 1 \end{bmatrix} + (-1) \times \begin{bmatrix} 3 \\ 2 \end{bmatrix} \end{bmatrix}$$

$$= \begin{bmatrix} -3 & -4 \\ -4 & -3 \\ 0 & 1 \\ 1 & 0 \end{bmatrix}.$$

Definition of STP

Pseudo-commutative law of STP:

Definition 2 [6]**:** An $mn \times mn$ matrix $W_{[m,n]}$ is called ***a swap matrix***, if it is constructed in the following way: label its columns by $(11,12,\cdots,1n,\cdots,m1,m2,\cdots,mn)$ and similarly label its rows by $(11,21,\cdots,n1,\cdots,1n,2n,\cdots,mn)$. Then its element in the position $((I,J);(i,j))$ is assigned as

$$w_{((I,J);(i,j))} = \begin{cases} 1, & I = i \text{ and } J = j, \\ 0, & otherwise. \end{cases}$$

If $\sigma_1 \in \Delta_m$ and $\sigma_2 \in \Delta_n$, then $W_{[m,n]}\sigma_1 \ltimes \sigma_2 = \sigma_2 \ltimes \sigma_1$.

$$\Delta_n := \{\delta_n^i, i = 1,2,\cdots,n\}$$

$$\delta_n^i := Col_i(I_n)$$

[6] D. Cheng, *et al*. Analysis and Control of Boolean Networks: A Semi-tensor Product Approach, Springer, 2010.

Algebraic Expression of BNs

$\mathcal{D} = \{1, 0\} \sim \Delta_2 = \{\delta_2^1, \delta_2^2\}$

$1 \sim \begin{bmatrix} 1 \\ 0 \end{bmatrix} = \delta_2^1, 0 \sim \begin{bmatrix} 0 \\ 1 \end{bmatrix} = \delta_2^2.$

Proposition 1 [6]**:** Let $f(a_1, \dots, a_n)$: $\{1,0\}^n \to \{1,0\}$ be a logical function. Then there exists a unique matrix $F \in \mathcal{L}_{2\times 2^n}$, such that

$$f(a_1, \dots, a_n) = F \ltimes a_1 \ltimes \cdots \ltimes a_n,$$

for every $(a_1, \dots, a_n) \in (\Delta_2)^n$. Here, F is called the ***structure matrix*** of f.

[6] D. Cheng, *et al.* Analysis and Control of Boolean Networks: A Semi-tensor Product Approach, Springer, 2010.

Algebraic Expression of BNs

$$\begin{cases} x_1(t+1) &= f_1([x_j(t)]_{j\in\mathbf{N}_1}), \\ \quad\vdots & \\ x_n(t+1) &= f_n([x_j(t)]_{j\in\mathbf{N}_n}). \end{cases} \quad (9)$$

$$x_i(t+1) = A_i \ltimes_{j\in\mathbf{N}_i} x_j(t), i \in [1:n].$$

$A_i \in \mathcal{L}_{2\times 2^{|N_i|}}$

$x(t) = \ltimes_{j=1}^{n} x_j(t)$

Algebraic expression of BNs:

$$x(t+1) = Lx(t)$$

$L \in \mathcal{L}_{2^n\times 2^n}$

[6] D. Cheng, *et al*. Analysis and Control of Boolean Networks: A Semi-tensor Product Approach, Springer, 2010.

Algebraic Expression of BNs

Example 1:

There are three persons. A says ``B is a liar", B says ``C is a liar", C says ``Both A and B are liars."

Who is a liar?

- p: A is honest,
- q: B is honest,
- r: C is honest.

$$p \Leftrightarrow \neg q, \quad q \Leftrightarrow \neg r, \quad r \Leftrightarrow \neg p \wedge \neg q.$$

$$\begin{cases} M_e p M_n q = c, \\ M_e q M_n r = c, \\ M_e r M_c M_n p M_n q = c. \end{cases}$$

$$Lx = b, x = pqr, b = c^3 = \delta_8^1,$$

$$L = \delta_8[8, 5, 2, 3, 4, 1, 5, 8].$$

$$x = \delta_8^6$$

$$\boldsymbol{p = 0, q = 1, r = 0}$$

Only B is honest!

Algebraic Expression of BNs

Example 2:

A competition between five players took place in a simple-rotating way, which means each player has to play all others.

What is the result ?

- C beat E,
- A won three games,
- E won one game,
- among B, C and D, there is one player who beat the other two,
- each of B, C, and D won two games,
- each of A, C, D, and E won some and lost some.

$p = AB$, $q = AC$, $r = AD$,

$s = AE$, $t = BC$, $u = BD$,

$v = BE$, $\alpha = CD$, $\beta = DE$.

p	q	r	s	t	u	v	α	β
1	1	0	1	1	1	0	1	1
1	0	1	1	1	1	0	0	1
1	0	1	1	1	0	1	0	0
0	1	1	1	0	0	1	0	0

Algebraic Expression of BNs

Example 3: A Boolean model of epigenetic system [7].

Find fixed points and cycles.

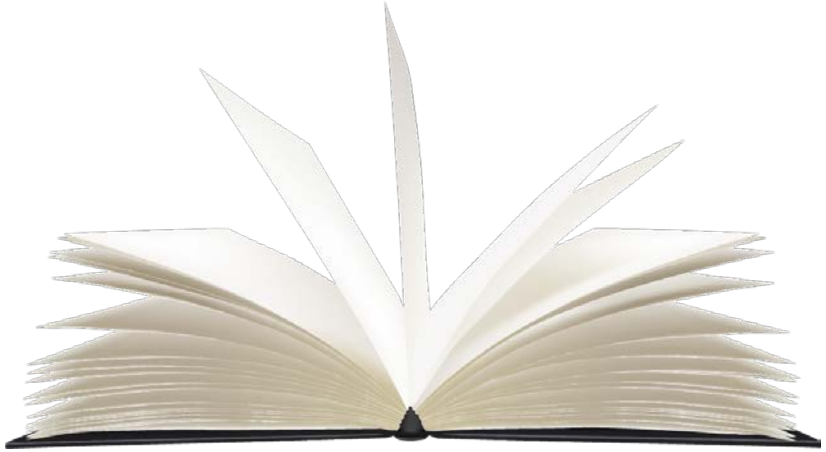
$$
\begin{cases}
A(t+1) = 1\bar{\vee}C(t)\bar{\vee}F(t)\bar{\vee}C(t)\wedge F(t),\\
B(t+1) = A(t),\\
C(t+1) = B(t),\\
D(t+1) = 1\bar{\vee}C(t)\bar{\vee}F(t)\bar{\vee}I(t)\bar{\vee}C(t)\wedge F(t)\\
\quad\quad \bar{\vee}C(t)\wedge I(t)\bar{\vee}F(t)\wedge I(t)\bar{\vee}C(t)\wedge F(t)\wedge I(t),\\
E(t+1) = D(t),\\
F(t+1) = E(t),\\
G(t+1) = 1\bar{\vee}F(t)\bar{\vee}I(t)\bar{\vee}F(t)\wedge E(t),\\
H(t+1) = G(t),\\
I(t+1) = H(t),
\end{cases}
$$

Only 6 cycles of length 6 by [7] !

→ Exactly 10 cycles of length 6 by STP

[7] J. Heidel, J. Maloney, C. Farrow, and J. Rogers, Finding cycles in synchronous Boolean networks with applications to biochemical systems. Int. J. Bifurcation Chaos, 13(3):535-552, 2003

CONTENTS

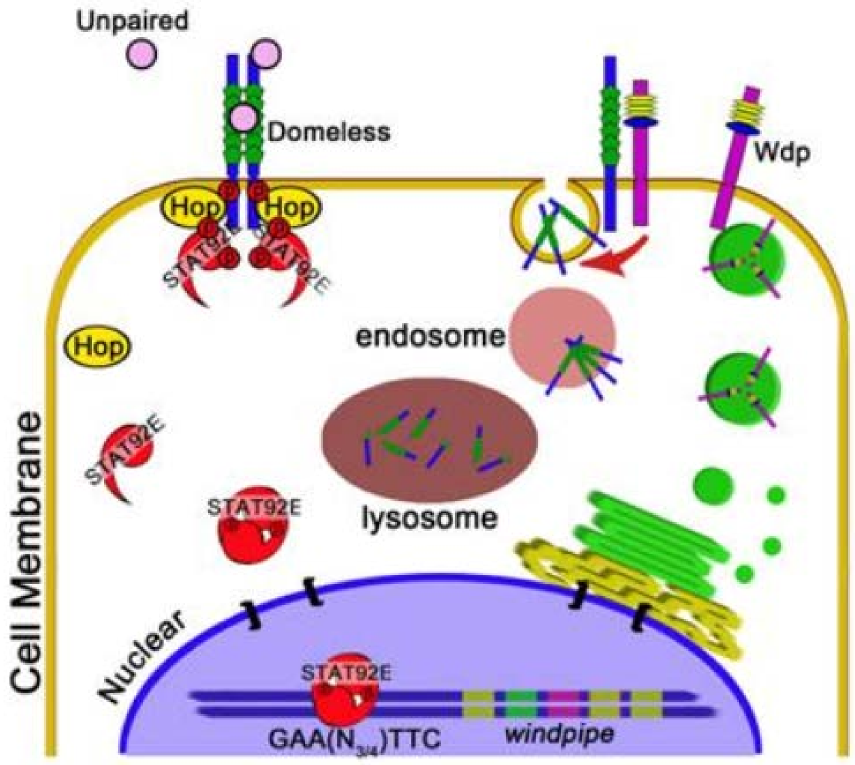


Part 1:
Stability

Stability

- Stability of Boolean networks is important and interesting topic. Especially, in **biological systems** or **genetic networks**.

- It is important to analyze whether systems can **reach** a desirable state, such as the healthy one, and **maintain** this state afterward.



Intestinal homeostasis in Drosophila

The definitions of stability in linear systems

$$\frac{\mathbf{d}\boldsymbol{x}}{\mathbf{d}\boldsymbol{t}} = \boldsymbol{f}(\boldsymbol{t},\boldsymbol{x}) \qquad \textbf{(1.1)}$$

Definition: The zero solution of (1.1) is said to be ***stable***, if $\forall \varepsilon > 0, \forall t_0 \in I, \exists \delta > 0$ such that $\forall x_0, \|x_0\| < \delta(\varepsilon, t_0)$ implies $\|x(t, t_0, x_0)\| < \varepsilon$ for $t \geq t_0$.

Definition: The zero solution of (1.1) is said to be ***attractive***, if $\forall t_0 \in I, \forall \varepsilon > 0, \exists\ \delta(t_0) > 0, \exists\ T(\varepsilon, t_0, x_0) > 0, \|x_0\| < \delta(t_0)$ implies $\|x(t, t_0, x_0)\| < \varepsilon$ for $t \geq t_0 + T$.

Definition: The zero solution of (1.1) is said to be ***asymptotically stable***, if it is stable and attractive.

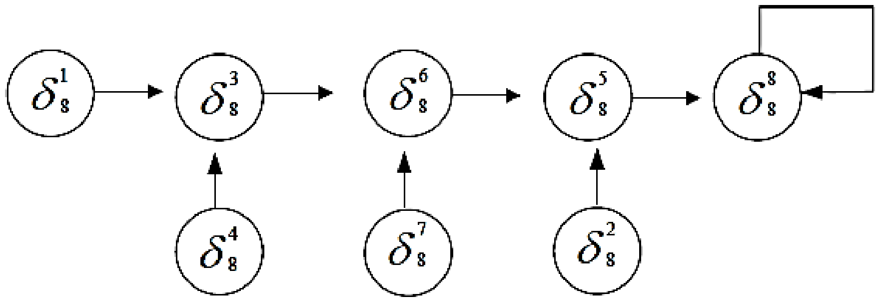
- Common stability analysis methods
  - ✓ All eigenvalues have a negative real part （homogeneous equation）
  - ✓ Routh-Hurwitz stability criterion （algebraic method）
  - ✓ Evans root locus plot
  - ✓ Nyquist stability criterion
  - ✓ Lyapunov's first method
  - ✓ Lyapunov direct method
  - ✓ LaSalle’s invariant principle
  - ✓ Comparison principle

Two definitions of stability in BNs

Definition[1]: A Boolean network is globally stable (or called asymptotically stable) if there exists a unique fixed point as the attractor with no other cycles.

Definition[2]: A Boolean network is said to be globally stable to a state $x^* \in \Delta_{2^n}$, if for any initial state $x_0 \in \Delta_{2^n}$, we have

$$\lim_{t\to\infty} x(t, t_0, x_0) = x^*.$$



[1] D. Cheng, et al., Stability and stabilization of Boolean networks , IJRNC, 21:134-156, 2011.
[2] F. Li, et al., Stability and stabilization of Boolean networks with impulsive effects, System & control letters, 61:1-5,2012.

Three common stability analysis methods in BNs

- **Incidence-matrix-based** stability analysis method
- **Transition-matrix-based** stability analysis method
- **Lyapunov-based** stability analysis method

Incidence-matrix-based stability analysis method[1]

➤ The introduce of **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

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

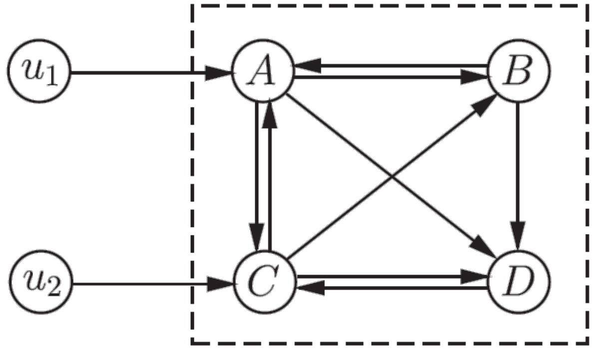


Figure 1. A Boolean (control) network.

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

[1] D. Cheng, et al., “Stability and stabilization of Boolean networks” , IJRNC, 21:134-156, 2011.

Incidence-matrix-based stability analysis method

➢ Logical operations on Boolean matrices

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,

$$\alpha\wedge X=(\alpha\wedge x_{ij});\quad X\wedge\alpha=(x_{ij}\wedge\alpha).$$

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

Incidence-matrix-based stability analysis method

➤ 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:
$$\alpha +_B \beta := \alpha \vee \beta,$$
$$^{B}\sum_{i=1}^{n} \alpha_i := \alpha_1 \vee \alpha_2 \vee \cdots \vee \alpha_n. \tag{20}$$

Incidence-matrix-based stability analysis method

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

$$z_{ij} =^B \sum_{k=1}^{n} x_{ik}\cdot y_{kj}, \quad i=1,\dots,m, \quad j=1,\dots,p.$$

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

$$A\ltimes_B B := (A\otimes I_t)\ltimes_B B. \quad (A\ltimes_B B := A\ltimes_B (B\otimes I_t).)$$

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

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

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

$\longrightarrow$

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

Incidence-matrix-based stability analysis method

- 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

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

For $x=(0\ \ 0\ \ 1\ \ 1)$ and $y=(0\ \ 1\ \ 0\ \ 1)$ one obtains

$$d(x, y)=\begin{bmatrix}0\\1\\1\\0\end{bmatrix}.$$

Incidence-matrix-based stability analysis method

- Now, we consider the **global stability** of BNs. The dynamics of a BN is expressed as

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

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

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

Incidence-matrix-based stability analysis method

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

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

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

Proof

$$\begin{aligned}
\delta_i(f_i(x_1, \ldots, x_n), f_i(y_1, \ldots, y_n)) \leq\ & \delta_i(f_i(x_1, \ldots, x_n), f_i(y_1, x_2, \ldots, x_n)) \\
& + \delta_i(f_i(y_1, x_2, \ldots, x_n), f_i(y_1, y_2, x_3, \ldots, x_n)) \\
& + \ldots \\
& + \delta_i(f_i(y_1, \ldots, y_{n-1}, x_n), f_i(y_1, \ldots, y_{n-1}, y_n)) \\
\leq\ & b_{i1}\delta_1(x_1, y_1) + b_{i2}\delta_2(x_2, y_2) + \ldots + b_{in}\delta_n(x_n, y_n) \\
\leqslant\ & \mathscr{I}(F) \ltimes_B d(X, Y).
\end{aligned}$$

[13] Robert F. Discrete Iterations, A Metric Study. Translated by J. Rokne. Springer: Berlin, 1986.

Incidence-matrix-based stability analysis method

Example：

Let $x=(0\ \ 0\ \ 1)$ and $y=(0\ \ 1\ \ 1)$ then

$$F(x)=(1\ \ 0\ \ 0) \quad \text{and} \quad F(y)=(1\ \ 0\ \ 1). \quad B(F)=\begin{bmatrix}1 & 1 & 1\\ 1 & 0 & 1\\ 0 & 1 & 1\end{bmatrix}$$

The inequality of Theorem 1 is now written as

$$d(F(x),F(y))=\begin{bmatrix}0\\0\\1\end{bmatrix}\leq\begin{bmatrix}1 & 1 & 1\\ 1 & 0 & 1\\ 0 & 1 & 1\end{bmatrix}\begin{bmatrix}0\\1\\0\end{bmatrix}=B(F)\,d(x,y)=\begin{bmatrix}1\\0\\1\end{bmatrix}.$$

Incidence-matrix-based stability analysis method

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

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

An immediate application of the above theorem is

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

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

If **0** is a fixed point of ***F*** and there exists an integer ***k***>**0** such that $[\mathscr{I}(F)]^{(k)} = 0$. Then, the system **globally converges** to 0 .

Incidence-matrix-based stability analysis method

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: Consider the follow system:

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

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

$$\mathscr{I}(F)=\begin{bmatrix}0&1&1&0\\0&0&0&1\\0&0&0&0\\0&0&1&0\end{bmatrix}. \Rightarrow [\mathscr{I}(F)]^{(4)}=0,$$

Assume that 0 is a fixed point of system (32), then it **globally converges** to 0.

Incidence-matrix-based stability analysis method

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.

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

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

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.

Incidence-matrix-based stability analysis method

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.

Just a sufficient condition!

Three common stability analysis methods in BNs

Incidence-matrix-based stability analysis method

Transition-matrix-based stability analysis method

Lyapunov-based stability analysis method

Transition-matrix-based stability analysis method

- Reconsider system (2):

$$x_1(t+1) = f_1(x_1, \ldots, x_n)$$
$$x_2(t+1) = f_2(x_1, \ldots, x_n)$$
$$\vdots$$
$$x_n(t+1) = f_n(x_1, \ldots, x_n), \quad x_i \in \mathscr{D},$$

$$\Rightarrow \quad x(t+1) = Lx(t), \quad x \in \Delta_{2^n}$$

where $x(t) := \ltimes_{i=1}^{n} x_i(t) \in \Delta_{2^n}$ and $L \in \mathscr{L}_{2^n \times 2^n}$ is the transition matrix

Lemma [3]: Consider system (2). Then,

1) $\boldsymbol{x}=\delta_{2^n}^{p}$ is switching reachable from $\boldsymbol{x}(0)=\delta_{2^n}^{q}$ at time $\boldsymbol{k}$, if and only if

$$(L^k)_{pq} > 0$$

2) $\boldsymbol{x}=\delta_{2^n}^{p}$ is switching reachable from $\boldsymbol{x}(0)=\delta_{2^n}^{q}$, if and only if

$$\Big(\sum_{i=1}^{2^n} L^i\Big)_{pq} > 0$$

[3] D. Laschov, et al., Controllability of Boolean control networks via the Perron-Frobenius theory, *Automatica*, 48(6):1218-1223, 2012.

Transition-matrix-based stability analysis method

Definition: An $s \times s$ logic matrix, $\boldsymbol{M}$, is said to be a matrix of constant mapping if there exists a δ_s^j such that

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

Theorem: System (2) is globally stable to state $\boldsymbol{x}_0$ ***if and only if*** satisfies

$$\boldsymbol{L}\boldsymbol{x}_0 = \boldsymbol{x}_0$$

and there exists an integer $\boldsymbol{k} > \mathbf{0}$, such that $\boldsymbol{L}^k$ is a constant mapping.

Transition-matrix-based stability analysis method

Example: Consider the following system:

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

It is easy to calculate that

$$L = \delta_4[1\ 3\ 4\ 1].$$

We have

$$L\delta_4^1 = \delta_4^1$$

and

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

Therefore, the system is globally stable to $x = \delta_4^1$.

Three common stability analysis methods in BNs

Incidence-matrix-based stability analysis method

Transition-matrix-based stability analysis method

Lyapunov-based stability analysis method

Lyapunov-based stability analysis method

- ✓ How to **define** a Lyapunov function for BNs?
- ✓ How to **establish** a new framework of Lyapunov stability theory for BNs?
- ✓ How to **construct** a Lyapunov function for BNs?

Stability analysis in linear systems

Theorem[1] : The zero solution of (1.1) is ***uniformly asymptotically stable*** if and only if there exists a positive definite function with infinitesimal upper bounded, $V(t,x) \in C^1$, such that along the solution of (1.1), $\frac{dV}{dt}|_{(1.1)}$ is negative definite.

- ***Example:*** $\dot{x}(t) = -2x + \sin x \qquad (1.2)$

We construct a Lyapunov function $V(x) = x^2$.

$$
\begin{aligned}
\frac{dV}{dt}\Big|_{(1.2)} &= 2x(-2x+\sin x) \\
&= -4x^2 + 2x\sin x \\
&\leq -4x^2 + 2|x||\sin x| \\
&\leq -4x^2 + 2x^2 \\
&= -2V(t)
\end{aligned}
$$

How to find V(x)? LMI et al.

[1] X.X. Liao, et al., Stability of dynamical systems, Monograph, 2007, 5(1):115.

Lyapunov-based stability analysis method

✓ How to **define** a Lyapunov function for BNs?

➤ The **pseudo-Boolean functions** was introduced in [14], which has wide applications in graph theory, game theory, and so on.

DEFINITION 2.6 (see [15]). *An n-ary pseudo-Boolean function $f(x_1, x_2, \ldots, x_n)$ is a mapping from $\mathcal{D}^n$ to $\mathbb{R}$.*

LEMMA 2.7 (see [15]). *A pseudo-Boolean function $f(x_1, x_2, \ldots, x_n)$ can be uniquely represented in the multilinear polynomial form of*

$$f(x_1, x_2, \ldots, x_n) = c_0 + \sum_{k=1}^{m} c_k \prod_{i \in A_k} x_i, \tag{2.3}$$

where c_0, c_1, ..., c_m are real coefficients, A_1, A_2, ..., A_m are nonempty subsets of $N = \{1, 2, \ldots, n\}$, and the product is the conventional one.

Remark 2.8. For x, $y \in \mathcal{D}$, it is easy to see that $x \wedge y = xy$ and $\neg x = \bar{x} = 1 - x$.

[14] P. Hammer, et al, On the determination of the minima of pseudo-Boolean functions, *Stud. Cerc. Mat.*, 14:359-364,1963.

[15] P. Hammer, et al. *Boolean Methods in Operations Research and Related Areas*,Springer, Berlin, 1968.

Lyapunov-based stability analysis method

➤ Algebraic expression of pseudo-Boolean functions

THEOREM 3.1. *Assume that* $f(x_1, x_2, \ldots, x_n) : \Delta^n \mapsto \mathbb{R}$ *is a pseudo-Boolean function. Then there exists a matrix* $M_f \in \mathbb{R}^{2\times 2^n}$ *such that*

$$f(x_1, x_2, \ldots, x_n) = J_1 M_f \ltimes_{i=1}^{n} x_i, \quad x_i \in \Delta, \tag{3.1}$$

where $J_1 = [1\ 0]$ *is called the selection matrix to be used to obtain the first row of* M_f*, and* $J_1 M_f$ *is unique.*

Li H, Wang Y. Lyapunov-based stability and construction of Lyapunov functions for Boolean networks[J]. SIAM Journal on Control and Optimization, 2017, 55(6): 3437-3457.

Lyapunov-based stability analysis method

Algebraic expression of pseudo-Boolean functions

Proof

$$
\begin{aligned}
&f(x_1, x_2, \ldots, x_n) \\
&\quad = c_0 + \sum_{k=1}^{m} c_k \prod_{i \in A_k} x_i \\
&\quad = J_1 \left[c_0 (E_d)^n (I_{2^n} \otimes \delta_2^1) \ltimes_{i=1}^{n} x_i + \sum_{k=1}^{m} c_k M_k \ltimes_{i=1}^{n} x_i \right] \\
&\quad = J_1 \left(c_0 (E_d)^n (I_{2^n} \otimes \delta_2^1) + \sum_{k=1}^{m} c_k M_k \right) \ltimes_{i=1}^{n} x_i \\
&\quad := J_1 M_f \ltimes_{i=1}^{n} x_i,
\end{aligned}
$$

where M_k is the structural matrix of term $\prod_{i \in A_k} x_i$, which can be uniquely determined by Lemma 2.3, $k = 1, \ldots, m$.

Lyapunov-based stability analysis method

Algebraic expression of pseudo-Boolean functions

Proof

Next, we prove that $J_1 M_f \in \mathbb{R}^{1\times 2^n}$ is unique.

In fact, if there exists another $J_1 M'_f \in \mathbb{R}^{1\times 2^n}$ such that (3.1) holds, then, for any $(x_1, x_2, \ldots, x_n) \in \mathcal{D}^n$ with $\ltimes_{i=1}^n x_i = \delta_{2^n}^k$, on one hand,

$$f(x_1, x_2, \ldots, x_n) = J_1 M_f \ltimes \delta_{2^n}^k = Col_k(J_1 M_f),$$

and on the other hand,

$$f(x_1, x_2, \ldots, x_n) = J_1 M'_f \ltimes \delta_{2^n}^k = Col_k(J_1 M'_f).$$

Thus, $Col_k(J_1 M'_f) = Col_k(J_1 M_f)\ \forall\ k = 1, 2, \ldots, 2^n$, which implies that $J_1 M'_f = J_1 M_f$, and the proof is completed. □

Lyapunov-based stability analysis method

✓ **Lyapunov functions** of Boolean networks

Consider the following BN:

$$\begin{cases} x_1(t+1) = f_1(x_1(t), x_2(t), \ldots, x_n(t)), \\ x_2(t+1) = f_2(x_1(t), x_2(t), \ldots, x_n(t)), \\ \vdots \\ x_n(t+1) = f_n(x_1(t), x_2(t), \ldots, x_n(t)), \end{cases} \tag{3.2}$$

⇒ $x(t+1) = Lx(t),$

where $x_i \in \mathcal{D}$, $i = 1, 2, \ldots, n$, are logical variables, and $f_i : \mathcal{D}^n \mapsto \mathcal{D}$, $i = 1, 2, \ldots, n$, are logical functions.

In general, the system (3.2) has a few attractors, including fixed points and/or cycles. We denote by $\mathcal{O}_e$ the set of fixed points and by $\mathcal{S}$ the set of points located in both fixed points and cycles. Obviously, $\mathcal{O}_e \subseteq \mathcal{S}$.

Lyapunov-based stability analysis method

✓ **Lyapunov functions** of Boolean networks

What is the form of Lyapunov function of BN (3.2)?

➢ Pseudo-Boolean function

$$
\begin{aligned}
V(x_1, x_2, \ldots, x_n) &= c_0 + c_1 x_1 + c_2 x_2 + \cdots + c_n x_n \\
&+ c_{n+1} x_1 x_2 + \cdots + c_{2^n - 1} x_1 x_2 \cdots x_n,
\end{aligned} \tag{3.3}
$$

where the total number of terms is $C_n^0 + C_n^1 + \cdots + C_n^n = 2^n$; c_i, $i = 0, 1, \ldots, 2^n - 1$

Lyapunov-based stability analysis method

✓ **Lyapunov functions** of Boolean networks

THEOREM 3.3. *The BN (3.2) is asymptotically stable at x_e if there exists a pseudo-Boolean function in the form of (3.3) satisfying*

(i) $V(x_1, x_2, \ldots, x_n) > 0$ *for* $\forall\ (x_1, x_2, \ldots, x_n) \neq x_e \in \mathcal{D}^n$, *and* $V(x_e) = 0$;

(ii) *along the trajectories of the system* (3.2), $\Delta V(x_1(t), \ldots, x_n(t)) := V(x_1(t+1), \ldots, x_n(t+1)) - V(x_1(t), \ldots, x_n(t)) < 0$ *holds for* $(x_1(t), \ldots, x_n(t)) \neq x_e$, *and* $\Delta V(x_1(t), \ldots, x_n(t)) = 0$ *holds for* $(x_1(t), \ldots, x_n(t)) = x_e$.

Proof. Assume that $V(x_1, x_2, \ldots, x_n)$ in the form of (3.3) satisfies (i) and (ii). Using the vector form of logical variables and setting $x = \ltimes_{i=1}^n x_i$, it can be seen from Theorem 3.1 and [4] that $V(x_1, x_2, \ldots, x_n)$ can be expressed as

$$V(x) = J_1 M_V \ltimes_{i=1}^n x_i \qquad (3.4)$$

where $J_1 M_V \in \mathbb{R}^{1\times 2^n}$, $x(t) \in \Delta_{2^n}$

Li H, Wang Y. Lyapunov-based stability and construction of Lyapunov functions for Boolean networks[J]. SIAM Journal on Control and Optimization, 2017, 55(6): 3437-3457.

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

Lyapunov-based stability analysis method

✓ **Lyapunov functions** of Boolean networks

Thus, along the trajectories of the BN (3.2), we have

$$\begin{aligned}
&\Delta V(x(t)) \\
&\quad = V(x_1(t+1), \ldots, x_n(t+1)) - V(x_1(t), \ldots, x_n(t)) \\
&\quad = J_1 M_V x(t+1) - J_1 M_V x(t) \\
&\quad = J_1 M_V L x(t) - J_1 M_V x(t) \\
&\quad = J_1 M_V (L - I_{2^n}) x(t),
\end{aligned}$$

Conditions (i) and (ii) →

$$\begin{cases}
c_0 = Col_{2^n}(J_1 M_V) = 0; \\
Col_i(J_1 M_V) > 0 \ \forall\ 1 \le i \le 2^n - 1; \\
Col_i(J_1 M_V (L - I_{2^n})) < 0 \ \forall\ 1 \le i \le 2^n - 1; \\
Col_{2^n}(J_1 M_V (L - I_{2^n})) = 0.
\end{cases} \quad (3.6)$$

Now, we show that x_e in the vector form of $\delta_{2^n}^{2^n}$ is a fixed point of the BN (3.2)

Lyapunov-based stability analysis method

✓ **Lyapunov functions** of Boolean networks

In fact, if $\delta_{2^n}^{2^n}$ is not a fixed point of the system (3.2), then $L\delta_{2^n}^{2^n} = \delta_{2^n}^{i}$, $i \neq 2^n$. Thus, we have

$$Col_{2^n}(L - I_{2^n}) = [0, \ldots, 0, \underbrace{1}_{i\text{th}}, 0, \ldots, 0, -1]^T.$$

Using (3.6), we obtain

$$\begin{aligned} & Col_{2^n}(J_1 M_V(L - I_{2^n})) \\ & = J_1 M_V Col_{2^n}(L - I_{2^n}) \\ & = Col_i(J_1 M_V) - Col_{2^n}(J_1 M_V) \\ & = Col_i(J_1 M_V) > 0, \end{aligned}$$

which is a contradiction with $Col_{2^n}(J_1 M_V(L - I_{2^n})) = 0$ (see (3.6)). Therefore, $\delta_{2^n}^{2^n}$ is a fixed point of the system (3.2).

Lyapunov-based stability analysis method

✓ **Lyapunov functions** of Boolean networks

Next, we prove that the system (3.2) is asymptotically stable at x_e.

Let $x(0) = \delta_{2^n}^{i_0}$ be any initial point. If $\delta_{2^n}^{i_0} \neq \delta_{2^n}^{2^n}$, then we obtain $x(1) = Lx(0) = \delta_{2^n}^{i_1}$. If $\delta_{2^n}^{i_1} \neq \delta_{2^n}^{2^n}$, then we have $x(2) = Lx(1) = \delta_{2^n}^{i_2} \ldots$. Keeping going, we obtain $x(k) = Lx(k-1) = \delta_{2^n}^{i_k}, \ldots$. Thus, we get the sequence

$$x(0) \to x(1) \to x(2) \to \cdots \to x(k) \to \cdots . \tag{3.7}$$

Since $x(k) = \delta_{2^n}^{i_k} \in \Delta_{2^n}$, $k \in \mathbb{N}$, and Δ_{2^n} is a finite set, we conclude that there exists an integer k_0 $(0 \leq k_0 \leq 2^n - 1)$ such that $x(k_0) = \delta_{2^n}^{2^n}$, which implies the sequence (3.7) converges to x_e. From the arbitrariness of $x(0)$, the system (3.2) is globally asymptotically stable at x_e. □

Lyapunov-based stability analysis method

- ✓ **Lyapunov functions** of Boolean networks

Based on Theorem 3.3, we give the following definition:

DEFINITION 3.4. *A pseudo-Boolean function* $V(x_1, \ldots, x_n)\colon \mathcal{D}^n \mapsto \mathbb{R}$ *in the form of* (3.3) *is called a strict-Lyapunov function of the BN* (3.2) *if it satisfies conditions* (i) *and* (ii) *in Theorem* 3.3.

DEFINITION 3.5. *A pseudo-Boolean function* $V(x_1, \ldots, x_n)\colon \mathcal{D}^n \mapsto \mathbb{R}$ *in the form of* (3.3) *is called a Lyapunov function of the BN* (3.2) *if*

(i) $V(x_1, \ldots, x_n) > 0$, $\forall (x_1, \ldots, x_n) \in \mathcal{D}^n \setminus \mathcal{O}_e$, *and* $V(x_1, \ldots, x_n) = 0$ *holds* $\forall (x_1, \ldots, x_n) \in \mathcal{O}_e$;

(ii) *along the trajectories of the system* (3.2), $\Delta V(x_1(t), \ldots, x_n(t)) < 0$ *holds for* $(x_1(t), \ldots, x_n(t)) \notin \mathcal{S}$, *and* $\Delta V(x_1(t), \ldots, x_n(t)) = 0$ *holds for* $(x_1(t), \ldots, x_n(t)) \in \mathcal{S}$.

With Definition 3.4 and Theorem 3.3, we have the following corollary.

COROLLARY 3.7. *If the BN* (3.2) *has a strict-Lyapunov function in the form of* (3.3), *then the system is globally asymptotically stable at* x_e.

Lyapunov-based stability analysis method

✓ **Lyapunov functions** of Boolean networks

Example 3.8. Consider the following BN:

$$\begin{cases} x_1(t+1) = x_1(t)\bar{\nabla}x_2(t), \\ x_2(t+1) = \neg(x_1(t) \rightarrow x_2(t)), \end{cases} \tag{3.8}$$

where $x_i \in \mathcal{D}$, $i = 1, 2$.

Choose $V(x_1, x_2) = 2x_1 + 3x_2 - 4x_1x_2$. Then, it is easy to check that $V(x_1, x_2) > 0$ for $(x_1, x_2) \neq (0, 0) \in \mathcal{D}^2$, and $V(0, 0) = 0$. On the other hand, we can easily check that along the trajectories of the BN (3.8), $\Delta V(x_1(t), x_2(t)) < 0$ holds for $(x_1(t), x_2(t)) \neq (0, 0)$, and $\Delta V(x_1(t), x_2(t)) = 0$ holds for $(x_1(t), x_2(t)) = (0, 0)$. Thus, $V(x_1, x_2)$ is a strict-Lyapunov function of the system (3.8). By Corollary 3.7, the system (3.8) is globally convergent to $(0, 0)$.

Li H, Wang Y. Lyapunov-based stability and construction of Lyapunov functions for Boolean networks[J]. SIAM Journal on Control and Optimization, 2017, 55(6): 3437-3457.

Lyapunov-based stability analysis method

✓ Construction of Lyapunov functions for Boolean networks

- In this subsection, we present two methods to construct a Lyapunov function for a given BN:

- **Definition-based** method
- **Structure-based** method

Li H, Wang Y. Lyapunov-based stability and construction of Lyapunov functions for Boolean networks[J]. SIAM Journal on Control and Optimization, 2017, 55(6): 3437-3457.

Lyapunov-based stability analysis method

✓ Construction of Lyapunov functions for Boolean networks

➤ First, we present the **definition-based** method

Assume that a Lyapunov function $V(x_1, x_2, \ldots, x_n)$ of the system is given in the form of (3.3), where c_i, $i = 0, 1, \ldots, 2^n - 1$, are real coefficients to be determined.

Set

(3.9) $$J_1 M_V = [a_1, a_2, \ldots, a_{2^n}].$$

Since

(3.10) $$\begin{cases} V(1,1,\ldots,1,1) = c_0 + c_1 + \cdots + c_{2^n-1} = a_1, \\ V(1,1,\ldots,1,0) = c_0 + \cdots + c_{n-1} \\ \qquad\qquad + c_{n+1} + \cdots = a_2, \\ \vdots \\ V(0,0,\ldots,0,0) = c_0 = a_{2^n}, \end{cases}$$

Theorem 3.1 ➡ P_n is nonsingular

there exists a $2^n \times 2^n$ matrix P_n such that ➡ $[c_0, c_1, \ldots, c_{2^n-1}]^T = P_n^{-1}[a_1, a_2, \ldots, a_{2^n}]^T.$

$$P_n[c_0, c_1, \ldots, c_{2^n-1}]^T = [a_1, a_2, \ldots, a_{2^n}]^T. \qquad (3.11)$$

Lyapunov-based stability analysis method

✓ **Construction of Lyapunov functions for Boolean networks**

- Denote the index set of all the fixed points in L by $\mathcal{I}_e = \{i_1,\ i_2, \ldots,\ i_k\}$
- $\mathcal{S}$ is the set of points in all the attractors of the BN (3.2)
- Denote by $\mathcal{I}_\mathcal{S}$ the index set of $\mathcal{S}$

$J_1 M_V$ should satisfy

$$\begin{array}{ll} Col_i(J_1 M_V) > 0, & i \in \{1,2,\ldots,2^n\} \setminus \mathcal{I}_e; \\ Col_i(J_1 M_V) = 0, & i \in \mathcal{I}_e; \\ Col_i(J_1 M_V(L - I_{2^n})) < 0, & i \in \{1,2,\ldots,2^n\} \setminus \mathcal{I}_\mathcal{S}; \\ Col_i(J_1 M_V(L - I_{2^n})) = 0, & i \in \mathcal{I}_\mathcal{S}, \end{array}$$

⟺

$$\begin{cases} a_i = 0, \ i \in \mathcal{I}_e, \\ a_i > 0, \ i \in \{1,2,\ldots,2^n\} \setminus \mathcal{I}_e, \\ [a_1, \ldots, a_{2^n}]\, Col_i(L - I_{2^n}) = 0, & i \in \mathcal{I}_\mathcal{S}, \\ [a_1, \ldots, a_{2^n}]\, Col_i(L - I_{2^n}) < 0, & i \in \{1,\ldots,2^n\} \setminus \mathcal{I}_\mathcal{S}. \end{cases} \qquad (3.13)$$

Whether equations (3.13) are solvable?

Lyapunov-based stability analysis method

✓ Construction of Lyapunov functions for Boolean networks

PROPOSITION 3.10. *The set of inequalities/equations (3.13) is always solvable.*

Proof. The proof is given by reduction to absurdity. If the set of inequalities/equations (3.13) is not solvable, then there exists at least one couple of incompatible inequalities in (3.13), which is in the form of

$$(3.14) \qquad \text{(I)}: \begin{cases} a_{i_0} > 0, \\ a_{i_0} < 0 \end{cases} \quad \text{or} \quad \text{(II)}: \begin{cases} a_{i_0} \geq a_{j_0}, \\ a_{i_0} < a_{j_0}, \end{cases}$$

where $a_{i_0} \neq 0$ and $a_{j_0} \neq 0$, $i_0 \neq j_0 \in \{1, 2, \ldots, 2^n\}$.

If the couple is case (I), the inequality $a_{i_0} < 0$ must come from the last one of (3.13), that is, $[a_1, a_2, \ldots, a_{2^n}]Col_i(L - I_{2^n}) < 0$. On the other hand, for any $a_i \neq 0$, it is easy to see from the structure of $L - I_{2^n}$ that the terms related to a_i in $[a_1, a_2, \ldots, a_{2^n}]Col_i(L - I_{2^n}) < 0$ should take the form of $a_i - a_j < 0$ (> 0) or $-a_i < 0$, where $a_j \neq 0$ and $j \neq i$, which implies that the inequality $a_{i_0} < 0$ does not appear in (3.13). This is a contradiction.

Lyapunov-based stability analysis method

✓ Construction of Lyapunov functions for Boolean networks

If the couple is in the form of $\{a_{i_0} = a_{j_0}$ and $a_{i_0} < a_{j_0}\}$, then it is easy to see from (3.13) and the equation $a_{i_0} = a_{j_0}$ that $j_0 \in \mathcal{I}_{\mathcal{S}}$. On the other hand, from the inequality $a_{i_0} < a_{j_0}$, we know that $j_0 \in \{1, 2, \ldots, 2^n\} \setminus \mathcal{I}_{\mathcal{S}}$. This is a contradiction. Thus, case (II) should take the form of

$$(\text{II})': \begin{cases} a_{i_0} > a_{j_0}, \\ a_{i_0} < a_{j_0}. \end{cases}$$

Lyapunov-based stability analysis method

✓ Construction of Lyapunov functions for Boolean networks

If the couple is the case (II)′, then the two incompatible inequalities must come from the last one of (3.13), that is, $[a_1, a_2, \ldots, a_{2^n}]Col_i(L - I_{2^n}) < 0$. Since $a_{i_0} \neq 0$ and $a_{j_0} \neq 0$, it can be seen from the structure of $L - I_{2^n}$ that there exist $\{a_{i_1}, \ldots, a_{i_r}\}$ and $\{a_{j_1}, \ldots, a_{j_s}\}$ such that both

$$\begin{array}{ccccccccc} a_{i_0} & > & a_{i_1} & > & \cdots & > & a_{i_r} & > & a_{j_0} \\ \downarrow & & \downarrow & & & & \downarrow & & \downarrow \\ V(\delta_{2^n}^{i_0}) & > & V(\delta_{2^n}^{i_1}) & > & \cdots & > & V(\delta_{2^n}^{i_r}) & > & V(\delta_{2^n}^{j_0}) \end{array}$$

and

$$\begin{array}{ccccccccc} a_{i_0} & < & a_{j_1} & < & \cdots & < & a_{j_s} & < & a_{j_0} \\ \downarrow & & \downarrow & & & & \downarrow & & \downarrow \\ V(\delta_{2^n}^{i_0}) & < & V(\delta_{2^n}^{j_1}) & < & \cdots & < & V(\delta_{2^n}^{j_s}) & < & V(\delta_{2^n}^{j_0}) \end{array}$$

hold one after the other along the trajectories of the system, which implies that

$$L\delta_{2^n}^{i_0} = \delta_{2^n}^{i_1}, \quad L\delta_{2^n}^{i_1} = \delta_{2^n}^{i_2}, \quad \ldots, \quad L\delta_{2^n}^{i_r} = \delta_{2^n}^{j_0},$$

$$L\delta_{2^n}^{j_0} = \delta_{2^n}^{j_s}, \quad L\delta_{2^n}^{j_s} = \delta_{2^n}^{j_{s-1}}, \quad \ldots, \quad L\delta_{2^n}^{j_1} = \delta_{2^n}^{i_0}.$$

Lyapunov-based stability analysis method

✓ Construction of Lyapunov functions for Boolean networks

Thus,

$$\left\{\delta_{2^n}^{i_0} \to \delta_{2^n}^{i_1} \to \cdots \to \delta_{2^n}^{i_r} \to \delta_{2^n}^{j_0}\right\}$$

and

$$\left\{\delta_{2^n}^{j_0} \to \delta_{2^n}^{j_s} \to \cdots \to \delta_{2^n}^{j_1} \to \delta_{2^n}^{i_0}\right\}$$

are two parts of the trajectories of the BN (3.2). Therefore, $\delta_{2^n}^{i_0}$ and $\delta_{2^n}^{j_0}$ are in the same cycle of the system. From the third equation of (3.13), we obtain $a_{i_0} = a_{j_0}$, which is a contradiction with (II)$'$.

Summarizing the above, we know that the set of inequalities/equations (3.13) is solvable. □

Lyapunov-based stability analysis method

✓ Construction of Lyapunov functions for Boolean networks

- With (3.13) and (3.11), we have the following algorithm to construct the desired Lyapunov function $V(x_1, x_2, \ldots, x_n)$ for the BN (3.2).

ALGORITHM 3.11. *Consider the BN (3.2) and assume that* $V(x_1, \ldots, x_n)$ *in the form of* (3.3) *is a Lyapunov function of the system to be found. To construct* $V(x_1, \ldots, x_n)$, *we can take the following steps:*

1. *Compute the matrix* L , *and find out the index sets* $\mathcal{I}_e$ *and* $\mathcal{I}_S$ *from* L.

2. *Solve the set of inequalities/equations* (3.13) *and obtain a solution* $(a_1, a_2, \ldots, a_{2^n})$.
3. *Compute the matrix* P_n *by* (3.10), *and then find out all the* c_i *by* (3.11) *with the obtained solution* $(a_1, a_2, \ldots, a_{2^n})$. *Then, the desired Lyapunov function is given as*

$$V(x_1, \ldots, x_n) = c_0 + c_1 x_1 + c_2 x_2 + \cdots + c_n x_n + c_{n+1} x_1 x_2 + \cdots + c_{2^n-1} x_1 x_2 \cdots x_n,$$

where $x_i \in \mathcal{D}$, $i = 1, 2, \ldots, n$.

Lyapunov-based stability analysis method

✓ Construction of Lyapunov functions for Boolean networks

- By Proposition 3.10 and Algorithm 3.11, we have the following result.

Proposition 3.12. *The BN* (3.2) *always has a Lyapunov function (the system's energy function) in the form of* (3.3).

- In the following, we give an example to show how to use Algorithm 3.11 to construct a Lyapunov function for a given BN.

Example 3.14. Construct a Lyapunov function for the following BN:

$$\left\{\begin{aligned} x_1(t+1) &= [x_1(t) \wedge (x_2(t) \to x_3(t))] \\ &\quad \vee (\neg x_1(t) \wedge x_3(t)), \\ x_2(t+1) &= [x_1(t) \wedge (x_2(t) \vee x_3(t))] \vee \neg x_1(t), \\ x_3(t+1) &= [x_3(t) \wedge (x_1(t) \leftrightarrow x_2(t))] \vee \neg x_3(t), \end{aligned}\right. \tag{3.15}$$

where $x_i \in \mathcal{D}$, $i = 1, 2, 3$.

Lyapunov-based stability analysis method

✓ Construction of Lyapunov functions for Boolean networks

Assume that the pseudo-Boolean function

$$V(x_1, x_2, x_3) = c_0 + c_1x_1 + c_2x_2 + c_3x_3 + c_4x_1x_2 + c_5x_1x_3 + c_6x_2x_3 + c_7x_1x_2x_3 \quad (3.16)$$

is a Lyapunov function of the system (3.15) to be found, where c_i, $i = 0, 1, \ldots, 7$, are real coefficients to be determined.

Using the vector form of logical variables and letting $x(t) = \ltimes_{i=1}^{3} x_i(t)$, one can easily obtain the algebraic form of the system (3.15) as $x(t+1) = Lx(t)$, where $L = \delta_8[1\ 5\ 2\ 3\ 2\ 5\ 1\ 5]$. Moreover, it is easy to see from L that $\mathcal{I}_e = \{1\}$ and $\mathcal{I}_{\mathcal{S}} = \{1, 2, 5\}$.

Lyapunov-based stability analysis method

✓ Construction of Lyapunov functions for Boolean networks

For this example, the set of ineqalities/equations (3.13) reduces to

$$(3.17)\qquad \begin{cases} a_1 = 0,\ \ a_i > 0,\ \ i = 2, \ldots, 8, \\ a_2 = a_5, \\ a_2 - a_3 < 0,\ \ a_3 - a_4 < 0, \\ a_5 - a_6 < 0,\ \ a_1 - a_7 < 0,\ \ a_5 - a_8 < 0, \end{cases}$$

which has an infinite number of solutions. For example, $(a_1, a_2, a_3, a_4, a_5, a_6, a_7, a_8) = (0, 1, 2, 3,\ 1, 2, 1, 2),\ (0, 2, 3, 4, 2, 6, 7, 8)$, and so on.

Choose a solution, say, $(0, 1, 2, 3, 1, 2, 1, 2)$. From (3.10) and (3.11), it is easy to obtain $[c_0, c_1, c_2,\ c_3, c_4, c_5, c_6, c_7]^T = P_3^{-1}[0, 1, 2, 3, 1, 2, 1, 2]^T = [2, 1, 0, -1, -2, 0, 0, 0]^T$. Thus, a Lyapunov function of the system (3.15) is given as

$$(3.18)\qquad V(x_1, x_2, x_3) = 2 + x_1 - x_3 - 2x_1x_2,$$

The (strict-)Lyapunov function of a BN is not unique. In general, the number of (strict-)Lyapunov functions is finite.

where $x_i \in \mathcal{D}$, $i = 1, 2, 3$. It is easy to check that (3.18) satisfies (i) and (ii) of Definition 3.5.

Lyapunov-based stability analysis method

✓ Construction of Lyapunov functions for Boolean networks

➤ In this subsection, we present two methods to construct a Lyapunov function for a given BN:

- **Definition-based** method
- **Structure-based** method

Li H, Wang Y. Lyapunov-based stability and construction of Lyapunov functions for Boolean networks[J]. SIAM Journal on Control and Optimization, 2017, 55(6): 3437-3457.

Lyapunov-based stability analysis method

✓ Construction of Lyapunov functions for Boolean networks

➤ Then, we present the **structure-based** method

Consider the BN (3.2) and let $V(x)$ in the algebraic form (3.4) be a Lyapunov function of the system. It is easy to see that if $J_1 M_V$ is expressed as (3.9), then $V(\delta_{2^n}^i) = a_i$, $i = 1, 2, \dots, n$. By Definition 3.5, $V(x)$ or a_i should have the following properties:

(i) If $\delta_{2^n}^{i_0}$ is a fixed point of the system (3.2) and $\{\delta_{2^n}^{i_1} \to \delta_{2^n}^{i_2} \to \cdots \to \delta_{2^n}^{i_k} \to \delta_{2^n}^{i_0}\}$ is a trajectory which converges to $\delta_{2^n}^{i_0}$, then

$$a_{i_1} > a_{i_2} > \cdots > a_{i_k} > 0 = a_{i_0}. \tag{3.19}$$

(ii) If $\{\delta_{2^n}^{i_1} \to \delta_{2^n}^{i_2} \to \cdots \to \delta_{2^n}^{i_k} \to \delta_{2^n}^{i_1}\}$ is a cycle of the system (3.2), then

$$a_{i_1} = a_{i_2} = \cdots = a_{i_k} > 0. \tag{3.20}$$

Lyapunov-based stability analysis method

✓ Construction of Lyapunov functions for Boolean networks

(iii) For each other point which is not a fixed point or a point in a cycle, it should locate at a trajectory containing one attractor, say, the trajectory is $\{\delta_{2^n}^{i_1} \to \delta_{2^n}^{i_2} \to \cdots \to \delta_{2^n}^{i_k} \to \delta_{2^n}^{i_0}\}$, where $\delta_{2^n}^{i_0}$ belongs to the attractor that the trajectory converges to. Then, we have

$$a_{i_1} > \cdots > a_{i_k} > a_{i_0}. \tag{3.21}$$

It is noted that (3.19)–(3.21) are sets of linear inequalities/equations. On the other hand, the number of all the basins and trajectories of the system (3.2) is finite, and they do not cross each other in backward time. This shows that (3.19)–(3.21) are sets of inequalities/equations of a simple relationship, not a coupled one, which implies that the solution set of one or more equations in the form of (3.19)–(3.21) is nonempty. Thus, with (i)–(iii), we can obtain a solution $(a_1, a_2, \ldots, a_{2^n})$. It should be pointed out that the solution set is generally infinite.

Lyapunov-based stability analysis method

✓ Construction of Lyapunov functions for Boolean networks

ALGORITHM 3.16. *Consider the BN (3.2) and assume that $V(x_1, \ldots, x_n)$ in the form of (3.3) is a Lyapunov function of the system to be found. To construct $V(x_1, \ldots, x_n)$, we can take the following steps:*

1. *Calculate the matrix L, and find out all the attractors and basins of the system (3.2) based on L by using the method of [4].*
2. *Based on the attractors/basins and (3.19)–(3.21), establish and solve a set of linear inequalities/equations, and obtain a solution $(a_1, a_2, \ldots, a_{2^n})$.*
3. *Compute the matrix P_n by (3.10), and then find out all the c_i by (3.11) with the obtained solution. Then, the desired Lyapunov function is given as*

$$V(x_1, \ldots, x_n) = c_0 + c_1 x_1 + c_2 x_2 + \cdots + c_n x_n + c_{n+1} x_1 x_2 + \cdots + c_{2^n-1} x_1 x_2 \cdots x_n,$$

where $x_i \in \mathcal{D}$, $i = 1, 2, \ldots, n$.

[4] D. Cheng, H. Qi, and Z. Li, Analysis and Control of Boolean Networks: A Semi-Tensor Product Approach, Springer, London, 2011.

Lyapunov-based stability analysis method

✓ Construction of Lyapunov functions for Boolean networks

Example 3.17. Consider the BN (3.15) given in Example 3.14.

Suppose that the Lyapunov function to be found is in the form of (3.16). Using the vector form of logical variables and setting $x(t) = \ltimes_{i=1}^{3} x_i(t)$, the system (3.15) can be rewritten as

$$x(t+1) = Lx(t), \quad L = \delta_8[1\ 5\ 2\ 3\ 2\ 5\ 1\ 5].$$

With this L, it is easy to see that the system (3.15) has a fixed point δ_8^1 and a cycle $\{\delta_8^2 \to \delta_8^5 \to \delta_8^2\}$. Meanwhile, we can obtain the entire four different trajectories including the basins of the above attractors, which are $\{\delta_8^7 \to \delta_8^1\}$, $\{\delta_8^4 \to \delta_8^3 \to \delta_8^2\}$, $\{\delta_8^6 \to \delta_8^5\}$, and $\{\delta_8^8 \to \delta_8^5\}$.

Lyapunov-based stability analysis method

✓ **Construction of Lyapunov functions for Boolean networks**

Based on (3.19)–(3.21) and the attractors/basins, we have the following set of linear inequalities/equations that all a_i satisfy:

$$\begin{cases} a_1 = 0,\ a_2 = a_5 > 0, & \longleftarrow\ \ \delta_8^1 \text{ and } \{\delta_8^2 \to \delta_8^5 \to \delta_8^2\}, \\ a_7 > a_1, & \longleftarrow\ \ \{\delta_8^7 \to \delta_8^1\}, \\ a_4 > a_3 > a_2, & \longleftarrow\ \ \{\delta_8^4 \to \delta_8^3 \to \delta_8^2\}, \\ a_6 > a_5, & \longleftarrow\ \ \{\delta_8^6 \to \delta_8^5\}, \\ a_8 > a_5, & \longleftarrow\ \ \{\delta_8^8 \to \delta_8^5\}, \end{cases}$$

which is a set of simple linear inequalities/equations. Obviously, it has an infinite number of solutions, for example, $(a_1, a_2, a_3, a_4, a_5, a_6, a_7, a_8) = (0, 1, 2, 3, 1, 2, 1, 2)$, $(0, 2, 4, 6, 2, 8, 8, 9)$, and so on.

Lyapunov-based stability analysis method

✓ **Construction of Lyapunov functions for Boolean networks**

If we choose

$$(a_1, a_2, a_3, a_4, a_5, a_6, a_7, a_8) = (0, 1, 2, 3, 1, 2, 1, 2),$$

then by (3.10) and (3.11) we obtain

$$\begin{aligned}[c_0, c_1, c_2, c_3, c_4, c_5, c_6, c_7]^T &= P_3^{-1}[0, 1, 2, 3, 1, 2, 1, 2]^T \\ &= [2, 1, 0, -1, -2, 0, 0, 0]^T.\end{aligned}$$

Thus, a Lyapunov function of the system (3.15) is given as

$$V(x_1, x_2, x_3) = 2 + x_1 - x_3 - 2x_1x_2,$$

which is the same as (3.18).

Lyapunov-based stability analysis method

✓ **Construction of Lyapunov functions for Boolean networks**

In Corollary 3.7, a sufficient condition for the asymptotical stability of BNs is obtained. Whether exists **a necessary and sufficient condition** for the **asymptotical stability** of BNs by using Lyapunov-based method ?

THEOREM 3.18 (converse Lyapunov theorem). *Consider the BN* (3.2). *If it is asymptotically stable at* x_e, *then the system has a strict-Lyapunov function* $V(x_1, \ldots, x_n) : \mathcal{D}^n \mapsto \mathbb{R}$ *in the form of* (3.3).

Proof. According to Proposition 3.12, the system (3.2) has a Lyapunov function $V(x_1, \ldots, x_n)$ in the form of (3.3). By Algorithm 3.16, we can find out one for the system. Now, we prove that such a $V(x_1, \ldots, x_n)$ is also a strict-Lyapunov function, i.e., $V(x_1, \ldots, x_n)$ satisfies (i)–(ii) of Definition 3.4.

Lyapunov-based stability analysis method

✓ **Construction of Lyapunov functions for Boolean networks**

Using the vector form of logical variables x_i, $i = 1, 2, \ldots, n$, we can obtain the algebraic form (3.4) of $V(x_1, \ldots, x_n)$ (also see (3.9)). Since the system (3.2) is globally convergent to x_e, x_e is a unique fixed point and the only attractor of the system. From the construction of $V(x)$ or (3.19), we know that $a_{2^n} = 0$ and $a_i > 0$, $1 \leq i \leq 2^n - 1$, which implies that $V(\delta_{2^n}^{2^n}) = 0$ and $V(\delta_{2^n}^{i}) > 0$, $1 \leq i \leq 2^n - 1$. That is, $V(x_e) = 0$ and $V(x_1, \ldots, x_n) > 0$ holds $\forall$ $(x_1, \ldots, x_n) \neq x_e \in \mathcal{D}^n$, which implies that (i) of Definition 3.4 is satisfied.

Lyapunov-based stability analysis method

✓ **Construction of Lyapunov functions for Boolean networks**

Let $\{\delta_{2^n}^{i_1} \to \delta_{2^n}^{i_2} \to \cdots \to \delta_{2^n}^{i_k} \to \delta_{2^n}^{2^n}\}$ be any trajectory converging to $\delta_{2^n}^{2^n}$. From (3.19), it is easy to know that $a_{i_1} > a_{i_2} > \cdots > a_{i_k} > 0 = a_{2^n}$, which implies that

$$V(\delta_{2^n}^{i_1}) > V(\delta_{2^n}^{i_2}) > \cdots > V(\delta_{2^n}^{i_k}) > 0 = V(\delta_{2^n}^{2^n}).$$

Thus, we obtain

$$\Delta V := V(\delta_{2^n}^{i_{j+1}}) - V(\delta_{2^n}^{i_j}) < 0, \quad 1 \le j \le k-1,$$

and

$$V(\delta_{2^n}^{2^n}) - V(\delta_{2^n}^{i_k}) < 0,$$

which implies that along the trajectory,

$$\Delta V(x_1(t), \ldots, x_n(t)) = V(x_1(t+1), \ldots, x_n(t+1)) \\ -V(x_1(t), \ldots, x_n(t)) < 0$$

Lyapunov-based stability analysis method

✓ Construction of Lyapunov functions for Boolean networks

holds for $(x_1(t), \ldots, x_n(t)) \neq x_e$. On the other hand, if $(x_1(t), \ldots, x_n(t)) = x_e$, then for any $\tau \geq t$, $(x_1(\tau), \ldots, x_n(\tau)) \equiv x_e$, which implies that $\Delta V(x_1(t), \ldots, x_n(t)) = 0$ for $(x_1(t), \ldots, x_n(t)) = x_e$. From the arbitrariness of the above trajectory, we know that $V(x)$ satisfies (ii) of Definition 3.4.

By Definition 3.4, $V(x_1, \ldots, x_n)$ is a strict-Lyapunov function, and the proof is completed. □

With Theorems 3.3 and 3.18, we have the following result:

THEOREM 3.19. *The BN (3.2) is asymptotically stable at x_e if and only if the system has a strict-Lyapunov function in the form of (3.3).*

Several types of stability

Partial stability

Global stability

Set stability

Partial stability[5]

- Consider the following system:

$$\begin{cases} \mathbf{x}_1(t+1) = f_1(\mathbf{x}_1(t), \mathbf{x}_2(t), \ldots, \mathbf{x}_n(t)), \\ \mathbf{x}_2(t+1) = f_2(\mathbf{x}_1(t), \mathbf{x}_2(t), \ldots, \mathbf{x}_n(t)), \\ \vdots \\ \mathbf{x}_n(t+1) = f_n(\mathbf{x}_1(t), \mathbf{x}_2(t), \ldots, \mathbf{x}_n(t)), \end{cases}$$

STP(1) →

$$x(t+1) = L_1 x(t), \quad (3)$$

where $L_1 = M_1 \ltimes_{j=2}^{n} [(I_{2^n} \otimes M_j)\Phi_n] \in \mathcal{L}_{2^n}$ alled the structure matrix of the Boolean network (1). Specific

where $\mathbf{x}_i \in \mathcal{D}, i = 1, 2 \ldots, n$ are logical variables, $f_i : \mathcal{D}^n \to \mathcal{D}, i = 1, 2, \ldots, n$ are logical functions.

Definition 3.2: For a given state $x_e^r = \ltimes_{l=1}^{r} x_l^e \in \Delta_{2^r}$, the Boolean network (1) is said to be partially stable to x_e^r, if for any initial state $x_0 = \ltimes_{i=1}^{n} x_i(0) \in \Delta_{2^n}$, there exists a positive integer τ such that $t \geq \tau$ implies that $x^r(t; x_0) = x_e^r$.

Boolean network (1) is stable with respect to just some, and not all of the state variables.

[5] H. Chen, et al, Partial stability and stabilisation of Boolean networks, *International Journal of Systems Science*, 47(9):2119-2127,2016.

Partial stability[5]

Theorem 3.3: *The Boolean network* (1) *is partially stable to* $x_e^r = \delta_{2^r}^{\lambda}, \lambda \in \{1, 2, \ldots, 2^r\}$, *if and only if there exists time* τ, *such that for matrix* $\Omega_\tau = E_d^{n-r} W_{[2^r, 2^{n-r}]} L_1^\tau \in \mathcal{L}_{2^r \times 2^n}$ *having equal columns as* $\delta_{2^r}^{\lambda}$, *i.e.*

Only consider the first *r* subsystems

$$\Omega_\tau = [\underbrace{\delta_{2^r}^{\lambda}, \delta_{2^r}^{\lambda} \ldots \delta_{2^r}^{\lambda}}_{2^n}]. \quad (4)$$

Global stability[6]

Here, we consider the following BN with disturbances:

$$\begin{cases} x_1(t+1) = f_1(\xi_1(t), \dots, \xi_p(t), x_1(t), \dots, x_n(t)), \\ \quad \cdots \\ x_n(t+1) = f_n(\xi_1(t), \dots, \xi_p(t), x_1(t), \dots, x_n(t)), \end{cases} \tag{1}$$

where $x_i \in \mathcal{D}$, $i = 1, \dots, n$, $\xi_j \in \mathcal{D}$, $j = 1, \dots, p$ are states and disturbance inputs, and $f_j : \mathcal{D}^{n+p} \to \mathcal{D}$ are logical functions. To convert system (1) into an algebraic form, define $x(t) = \ltimes_{i=1}^{n} x_i(t) \in \Delta_{2^n}$ and $\xi(t) = \ltimes_{j=1}^{p} \xi_j(t) \in \Delta_{2^p}$. Assume that the structure matrix of f_i is $F_i \in \mathscr{L}_{2^n \times 2^{n+p}}$, then system (1) can be expressed as the following algebraic form:

$$x(t+1) = L\xi(t)x(t), \tag{2}$$

where $L \in \mathscr{L}_{2^n \times 2^{n+p}}$ is called the transition matrix of system (1) and $\mathrm{Col}_i(L) = \ltimes_{j=1}^{n} \mathrm{Col}_i(F_j)$, $i = 1, \dots, 2^{n+p}$.

[6] J. Zhong, et al, Global robust stability and stabilization of Boolean network with disturbances, *Automatica*, 84:142-148,2017.

Global stability[6]

Definition 4. BN (1) with disturbances is globally robust stable w.r.t. a fixed point $X^* = \delta_{2^n}^r$, if for any initial state $x(0) \in \Delta_{2^n}$ and any disturbance inputs $\xi_1 \in \Delta_2, \ldots, \xi_p \in \Delta_2$, there exists an integer k such that $x(t) = \delta_{2^n}^r, t \geq k$.

Definition 5. A state $\mathbb{X}$ is a reachable state from an initial state $x(0)$ at the kth step if there exists a disturbance sequence $\xi(0), \ldots, \xi(k-1)$, such that $\mathbb{X} = x(k)$. Then, for all possible disturbance sequences $\xi(0), \ldots, \xi(k-1)$, the set of all reachable states is denoted by $R_S^k(x(0))$, which is called the set of robustly reachable states from $x(0)$ at the kth step.

Global stability[6]

Split matrix L into 2^p equal blocks as $L = [L_1, \ldots, L_{2^p}]$, where $L_j \in \mathscr{L}_{2^n \times 2^n}$ $(j = 1, \ldots, 2^p)$.

Some results of Global stability

Proposition 2. *System (1) is globally robust stable w.r.t. the fixed point $X^* = \delta_{2^n}^r$ if and only if there is an integer k satisfying $1 \leq k \leq 2^n - 1$, such that $R_S^k(\Delta_{2^n}) = \delta_{2^n}^r$.*

Proposition 3. *Consider BN (1) with disturbances, the column vector form $\mathcal{R}_S^k(\Delta_{2^n})$ is presented as follows: $\mathcal{R}_S^k(\Delta_{2^n}) = \sum_{\mathscr{B}j=1}^{2^n} \mathrm{Col}_j(\mathfrak{L}^{(k)})$, $\mathfrak{L} = \sum_{\mathscr{B}j=1}^{2^p} L_j$.*

↓

Theorem 1. *System (1) is globally robust stable w.r.t. the fixed point $X^* = \delta_{2^n}^r$ if and only if there exists an integer $1 \leq k \leq 2^n - 1$, such that $\sum_{\mathscr{B}j=1}^{2^n} \mathrm{Col}_j(\mathfrak{L}^{(k)}) = \delta_{2^n}^r$.*

[6] J. Zhong, et al, Global robust stability and stabilization of Boolean network with disturbances, *Automatica*, 84:142-148,2017.

Set stability[7]

A BN with n nodes is described as

$$\begin{cases} A_1(t+1) = f_1(A_1(t), \ldots, A_n(t)) \\ \vdots \\ A_n(t+1) = f_n(A_1(t), \ldots, A_n(t)) \end{cases}$$

STP →

$$x(t+1) = Lx(t) \quad (2)$$

Definition 1 (*Set Stability*). Let $\mathcal{M}$ be a subset of Δ_{2^n}. BN (2) is said to be $\mathcal{M}$-stable if, for any initial state $x_0 \in \Delta_{2^n}$, there exists a $T(x_0) \in \mathbb{Z}_{\geq 0}$ such that

$$x(t; x_0) \in \mathcal{M}, \quad \forall t \geq T(x_0). \quad (4)$$

When the subset $\mathcal{M}$ is a singleton, → stability

The denotation $T_{\mathcal{M}}(x_0)$ refers to the smallest integer such that (4) holds, which is called the transient period from x_0. The transient period of BN (2) is defined as $T_{\mathcal{M}} := \max_{x_0 \in \Delta_{2^n}} T_{\mathcal{M}}(x_0)$.

---

[7] Y. Guo, et al, Set stability and set stabilization of Boolean control networks based on invariant subsets, *Automatica*, 61:106-112,2015.

Set stability[7]

The analysis of set stability can be divided into two steps:

1. All of states finally can enter the set $\mathcal{M}$

2. The states which enter the $\mathcal{M}$ always stay the set $\mathcal{M}$

Definition 3 (*Invariant Subset*). A subset $\mathcal{C} \subseteq \Delta_{2^n}$ is called an invariant subset of BN (2) if $x(t; x_0) \in \mathcal{C}, \forall t \in \mathbb{Z}_{\geq 0}, \forall x_0 \in \mathcal{C}$.

Remark The union of any two invariant subsets is still an invariant subset. The union of all the invariant subsets contained in a given set $\mathcal{M}$ is called the largest invariant subset contained in $\mathcal{M}$, denoted by $I(\mathcal{M})$.

[7] Y. Guo, et al, Set stability and set stabilization of Boolean control networks based on invariant subsets, *Automatica*, 61:106-112,2015.

Set stability[7]

- For any $\mathbf{F} \in \mathscr{B}_{m\times n}$, a logical matrix $F \in \mathscr{L}_{m\times n}$ is called a logical sub-matrix of $\mathbf{F}$ if $F \wedge \mathbf{F} = F$. Denote by $\mathscr{S}(\mathbf{F})$ the set of all of the logical sub-matrices of $\mathbf{F}$, i.e., $\mathscr{S}(\mathbf{F}) := \{F \in \mathscr{L}_{m\times n} \mid F \wedge \mathbf{F} = F\}$. Especially, for any nonzero $x \in \mathscr{B}_{m\times 1}$, $\mathscr{S}(x) = \{z \in \Delta_m \mid z \wedge x = z\}$. For convenience, define $\mathscr{S}^T(x) := \mathscr{S}(x^T)$ for any $x \in \mathscr{B}_{1\times n}$.

Proposition 1. *Assume that $q = |\mathcal{M}|$. Define M_0 as*

$$\mathrm{Col}_j(M_0) = \begin{cases} \delta_{2^n}^j, & \delta_{2^n}^j \in \mathcal{M} \\ \delta_{2^n}^0, & \delta_{2^n}^j \notin \mathcal{M}, \end{cases}$$

and a collection of Boolean matrices M_i, $1 \le i \le q$, as

$$M_i := M_0 L M_{i-1} = (M_0 L)^i M_0, \quad 1 \le i \le q.$$

Then, it holds that $I(\mathcal{M}) = \mathscr{S}^T[\mathrm{Row}_\Sigma(M_q)]$.

The largest invariant subset →

[7] Y. Guo, et al, Set stability and set stabilization of Boolean control networks based on invariant subsets, *Automatica*, 61:106-112,2015.

Set stability[7]

Proposition 3. *BN (2) is $\mathcal{M}$-stable if and only if* $\mathrm{Col}(L^{2^n}) \subseteq \mathcal{M}$ *and in addition, there holds* $T_{\mathcal{M}} = \min_{k\in\mathbb{Z}_{\geq 0}}\{k \mid \mathrm{Col}(L^k) \subseteq \mathcal{M}\}$. □

Lemma 3. *Given* $\mathcal{M} \subseteq \Delta_{2^n}$, *BN (2) is $\mathcal{M}$-stable if and only if it is $I(\mathcal{M})$-stable. In addition, for any $\mathcal{M}$-stable BN, there holds* $T_{\mathcal{M}}(x_0) = T_{I(\mathcal{M})}(x_0), \forall x_0 \in \Delta_{2^n}$. □

Proposition 4. *The following statements are true.*

(1) *BN (2) is $\mathcal{M}$-stable if and only if*

$$\mathrm{Row}_{\Sigma}[(M_0L)^q M_0 L^p] = \mathbf{1}_{2^n}^T \tag{23}$$

where $q = |\mathcal{M}|$ *and* $p = 2^n - |I(\mathcal{M})|$.

(2) *There holds*

$$T_{\mathcal{M}}(\delta_{2^n}^j) = \min_{k\in\mathbb{Z}_{\geq 0}}\left\{k \mid (M_0L)^q M_0 \mathrm{Col}_j(L^k) \neq 0\right\}. \tag{24}$$

Application of stability in synchronization

Consider a master BN as

$$x_i(t+1) = f_i(x_1(t), \ldots, x_n(t)), \quad t \geq 0 \tag{2}$$

where $f_i : \mathcal{D}^n \to \mathcal{D}$, $1 \leq i \leq n$ are logical functions. By Lemma 3, letting $x(t) = \ltimes_{i=1}^{n} x_i(t)$, (2) can further be converted into the following discrete-time system:

$$x(t+1) = Fx(t), \quad t \geq 0 \tag{3}$$

where $F = M_1 \ltimes_{i=2}^{n} [(I_{2^n} \otimes M_i)\Phi_n] \in \mathcal{L}_{2^n \times 2^n}$ and M_i is the structure matrix of f_i. For any given $1 \leq h \leq n$, let

$$x'(t) \triangleq \ltimes_{i=1}^{h} x_i(t) = B_1 x(t) \tag{4}$$

where

$$B_1 = \delta_{2^h}[\underbrace{1, 1, \ldots, 1}_{2^{n-h}}, \underbrace{2, 2, \ldots, 2}_{2^{n-h}}, \ldots, \underbrace{2^h, 2^h, \ldots, 2^h}_{2^{n-h}}] \in \mathcal{L}_{2^h \times 2^n}.$$

If $h = n$, then $B_1 = I_{2^n}$ and $x'(t) = x(t)$.

Application of stability in synchronization

A slave BN expressed by

$$y_i(t+1) = l_i(x_1(t), \ldots, x_n(t), y_1(t), \ldots, y_n(t)), \quad t \geq 0 \tag{5}$$

STP →

$$y(t+1) = Lx(t)y(t) \tag{6}$$

$$\text{and } y'(t) \triangleq \ltimes_{i=1}^{h} y_i(t) = B_1 y(t), \quad t \geq 0 \tag{7}$$

Definition 3 [20]: For a given $1 \leq h \leq n$, (2) and (5) achieve h-partial synchronization for any initial states $x(0)$ and $y(0)$, and and an integer $k \geq 0$, such that for all $t \geq k$, $x'(t) = y'(t)$. In particular, if $x(t) = y(t)$ for all $t \geq k$, complete synchronization of (2) and (5) occurs.

Application of stability in synchronization

Let $z'(t) = x'(t)y'(t)$ and $z(t) = x(t)y(t)$. Then, we have

Proposition 1: The h-partial synchronization of (2) and (5) occurs, i.e., there exists an integer $k \geq 0$, such that for $t \geq k$, $x'(t) = y'(t) = \delta_{2^h}^{i_t}$, for some $i_t \in \Omega_h$, if and only if for $t \geq k$, $z'(t) = \delta_{2^{2h}}^{\lambda}$, where $\lambda = (i_t - 1)2^h + i_t$, $\Omega_h = \{1,2,\dots,2^h\}$.

Application of stability in synchronization[8]

We consider the following two kinds of arrays of M delayed coupled BNs, with each BN being an N-nodes system:

$$\begin{cases} X_j^i(t+1) = f_j^i(X_j^1(t-\tau), X_j^2(t-\tau), \ldots, X_j^N(t-\tau) \\ \qquad\qquad y_1(t-\tau), y_2(t-\tau), \ldots, y_M(t-\tau)) \\ y_j(t) = g_j(X_j^1(t), \ldots, X_j^N(t)) \end{cases} \quad (1)$$

and

$$\begin{cases} X_j^i(t+1) = f_j^i(X_j^1(t-\tau), \ldots, X_j^N(t-\tau), y_1(t), \ldots, y_M(t)) \\ y_j(t) = g_j(X_j^1(t), \ldots, X_j^N(t)) \end{cases} \quad (2)$$

where X_j^i is the ith node of the jth BN, y_j is the binary output of the jth BN, $f_j^i : \{1,0\}^{NM} \to \{1,0\}$, $g_j : \{1,0\}^N \to \{1,0\}$ are Boolean functions, $i = 1, \ldots, N$, $j = 1, \ldots, M$, $t = 0, 1, \ldots,$ and τ is a nonnegative integer. We simply denote by $X_j(t) = (X_j^1(t), X_j^2(t), \ldots, X_j^N(t))$ the states of jth BN at time instant t. Within the isolated BN, communication delay is considered in the process of information exchange among its nodes in both models.

Application of stability in synchronization[8]

Definition 2: The array of BNs in (1) and (2) is said to be synchronized if for any initial states $X_j(-\tau), \ldots, X_j(0) \in \{1,0\}^N$, $j = 1, \ldots, M$, there is a positive integer k, such that $t \geq k$ satisfies $X_i(t) = X_j(t)$ for any $1 \leq i, j \leq N$.

By STP, system (1) can be converted into the following form:

$$\begin{cases} X_j(t+1) = F_j X_j(t-\tau) y(t-\tau) \\ y(t) = G \ltimes_{j=1}^{M} X_j(t) \end{cases} \tag{3}$$

where F_j is a $2^N \times 2^{MN}$ matrix and G is a $2^M \times 2^{MN}$ matrix.

[8] J. Zhong, et al, Synchronization in an Array of Output-Coupled Boolean Networks With Time Delay, *IEEE Transaction on neural networks and learning systems*, 25(12):2288-2294,2014.

Application of stability in synchronization[8]

Lemma 3: Let $W = W_{[2^M, 2^N]} \ltimes \{\ltimes_{i=2}^{M}[(I_{2^M} \otimes W_{[2^M, 2^{iN}]})\Phi_M]\}$ and $\Xi = (\otimes_{j=1}^{M} F_j) \ltimes W \ltimes G \ltimes \Phi_{MN}$. Then, we have

$$\ltimes_{j=1}^{M} X_j(t+1) = \Xi[\ltimes_{j=1}^{M} X_j(t-\tau)] \quad (4)$$

and

$$\ltimes_{j=1}^{M} X_j(t) = \Xi^{p+1}[\ltimes_{j=1}^{M} X_j(q-1-\tau)] \quad (5)$$

where $p \geq 0$ and $1 \leq q \leq \tau + 1$ are the unique integers satisfying $t = p(\tau+1) + q$.

Application of stability in synchronization[8]

Now, we present necessary and sufficient synchronization criterion for an array of delay-coupled BNs in form of (1):

Theorem 1: Let (3) be the algebraic representations of the array of delayed BNs (1). Then, synchronization occurs if and only if there exists a positive integer k satisfying $1 \leq k+1 \leq k_0$ such that

$$\mathrm{Col}(\Xi^{k+1}) \subseteq \left\{\delta_{2^{MN}}^{\lambda_i} : \lambda_i = 1 + \frac{(i-1)(2^{MN}-1)}{2^N - 1}, i = 1, 2, \ldots, 2^N\right\} \quad (6)$$

where $k_0 = \min\{i : i \geq 1, \Xi^i = \Xi^j \text{ for some } j > i\}$.

[8] J. Zhong, et al, Synchronization in an Array of Output-Coupled Boolean Networks With Time Delay, *IEEE Transaction on neural networks and learning systems*, 25(12):2288-2294,2014.

Some references about stability analysis

[1] S. Zhu, et al. “Asymptotical stability of probabilistic Boolean networks with state delays”. *IEEE Transaction on Automatic Control*, 65(4): 1779-1784, 2020.

[2] B. Li, et al. “Fast-time stability of temporal Boolean networks”. *IEEE Transactions on Neural Networks and Learning*, 30(8): 2285-2294, 2019.

[3] M. Meng, et al. “Stability and *l*_{1} gain analysis of Boolean networks with markovian jump parameters”. *IEEE Transactions on Neural Networks and Learning*, 30(8): 2285-2294, 2019.

[4] M. Meng, et al. “Stability and Stabilization of Boolean Networks With Stochastic Delays”. *IEEE Transaction on Automatic Control*, 65(4): 790-796, 2019.

[5] C. Huang, et al.” Stability and stabilization in probability of probabilistic Boolean networks”. *IEEE Transactions on Neural Networks and Learning*, Doi: 10.1109/TNNLS.2020.2978345, 2020.

[6] Y. Guo, et al. “Invariant subset and set stability of Boolean networks under arbitrary switching signals”. *IEEE Transaction on Automatic Control*, 62(8): 1779-1784, 2020.

[7] H. Li, et al.” Robustness for stability and stabilization of Boolean networks with stochastic function perturbations”. *IEEE Transaction on Automatic Control*, Doi:10.1109/TAC.2020.2997282, 2020.

…………………

Part 2:
Stabilization

Boolean Control Network (BCN)

- When ***control inputs*** are added, the concept of BNs extends naturally to that of **BCNs**.

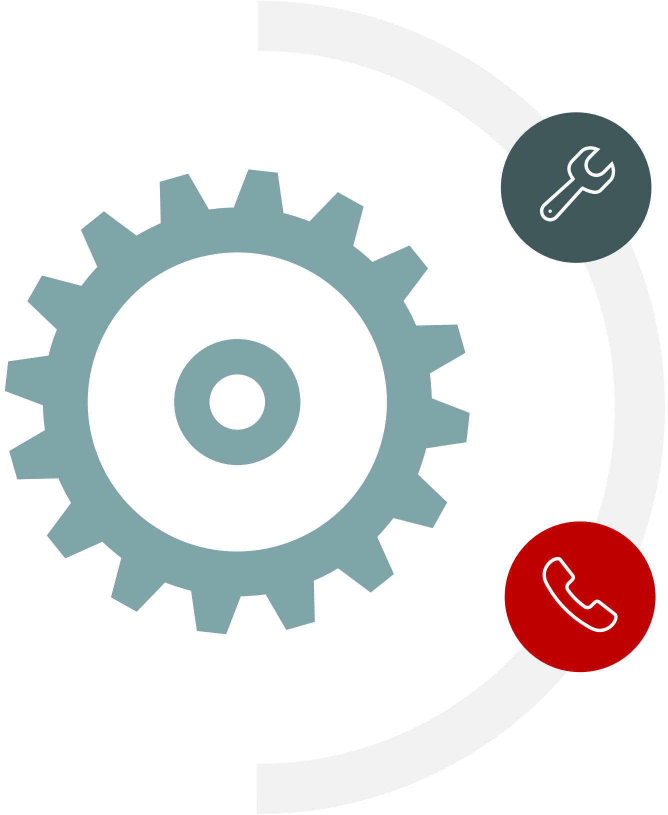
- $$\begin{cases} x_i(t+1) = f_i(X(t), U(t)), \ i = 1, 2, \ldots, n, \\ y_j(t) = h_j(X(t)), \ j = 1, \ldots, p, \end{cases} \quad (1)$$

where $X(t)=(x_1(t), x_2(t), \ldots, x_n(t)) \in \mathcal{D}^n$, $U(t)=(u_1(t), u_2(t), \ldots, u_m(t)) \in \mathcal{D}^m$ and $Y(t)=(y_1(t), y_2(t), \ldots, y_p(t)) \in \mathcal{D}^p$ are states, ***control inputs*** and ***outputs*** at time ***t*** of BCN (1), and f_i: $\mathcal{D}^{m+n} \to \mathcal{D}, i = 1, \ldots, n$.

- **Algebraic representation**:

$$\begin{cases} x(t+1) = Lu(t)x(t), \ x(t) \in \Delta_{2^n}, \ u(t) \in \Delta_{2^m}, \ L \in \mathcal{L}_{2^n \times 2^{n+m}} \\ y(t) = Hx(t), \ y(t) \in \Delta_{2^p}, \ H \in \mathcal{L}_{2^p \times 2^n} \end{cases} \quad (2)$$

Boolean control networks

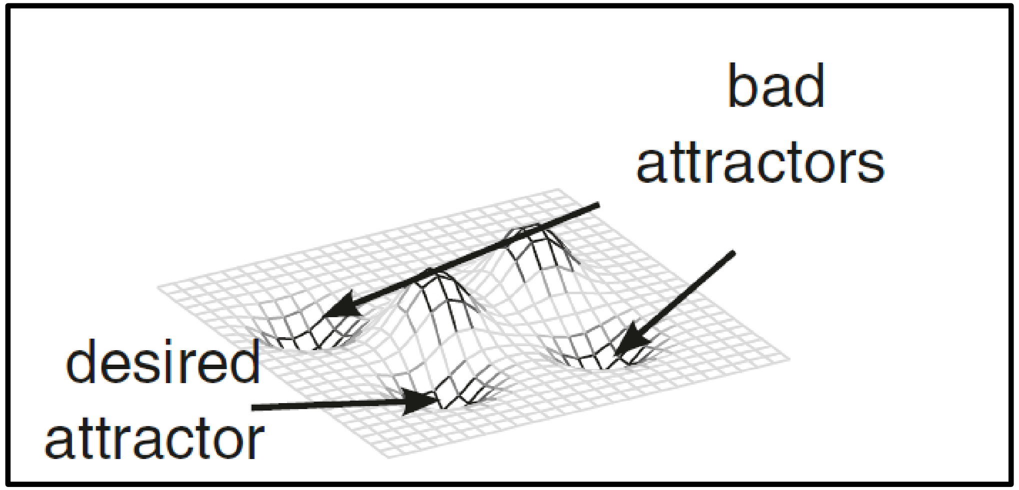


Stabilization

Set stabilization

Sabilization

- The **global stabilization** problem of system (1) is to find, if possible, $\{\boldsymbol{u}(\boldsymbol{t})\}$ such that the system becomes **globally convergent**.



- In the case of **disease treatment**, one may want to **design** therapeutic interventions that steer the patient to the **desirable state**, such as the healthy one, and **maintain** this state afterward, that is the **stabilization**.

Definition 1: For a given state $\mathbf{X}^* \in \{1,0\}^n$, the BCN (1) is said to be *globally stabilizable* to $\mathbf{X}^*$, if for every $\mathbf{X}_0 \in \{1,0\}^n$ there exist a control sequence $\mathbf{U} : \{0,1,2,\ldots\} \to \{1,0\}^m$ and a positive integer N such that $t \geq N$ implies that $\mathbf{X}(t;\mathbf{X}_0;\mathbf{U}) = \mathbf{X}^*$.

Stabilization → **Existing methods**

- **Open-loop control design technique**
- **State feedback control design technique**
  - **Reachability set**
  - **Control Lyapunov function**
  - **Pinning control**
  - **Event-triggered control**
  - **Sampled-data control**
- **Output feedback control design technique**

Stabilization → **Existing methods**

Open-loop control design technique

State feedback control design technique

- **Reachability set**
- **Control Lyapunov function**
- **Pinning control**
- **Event-triggered control**
- **Sampled-data control**

Output feedback control design technique

Open-loop control design technique[1]

- ***Objective***: find a control sequence $\{u(t)\} \subseteq \mathcal{D}^m$ such that BCN(1) can achieve **global stabilization.**

- Define: $$s_k^n = [\underbrace{1\ldots1}_{2^{n-k}}\underbrace{0\ldots0}_{2^{n-k}}\ldots\underbrace{1\ldots1}_{2^{n-k}}\underbrace{0\ldots0}_{2^{n-k}}] \in \mathscr{B}_{1\times 2^n},\ k=1,\ldots,n.$$

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

[1] D. Cheng, H. Qi, Z. Li and J. Liu, Stability and stabilization of Boolean networks, *International Journal of Robust and Nonlinear Control*, 2011, 21(2): 134-156.

Open-loop control design technique[1]

- From scalar form to vector form, we have

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

- From vector form to scalar form, we have $X=\mathscr{S}^n x$.

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

Open-loop control design technique[1]

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

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

where M_n is the structure matrix of negation.

Theorem 1: Consider the Boolean network with its algebraic form. The incidence matrix of F can be obtained from L by the following formula:

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

Open-loop control design technique[1]

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

Open-loop control design technique[1]

Lemma 1: System (1) is **stabilizable** by an **open-loop control** $\boldsymbol{u}$, **if** $\boldsymbol{\pi}(\boldsymbol{Lu})$ has a strictly lower (or upper) triangular form.

Theorem 2: System (1) (or its algebraic form (2)) is **stabilizable** by an **open-loop control** $\boldsymbol{u}$, **if** there is a coordinate transformation $\boldsymbol{z}=\boldsymbol{Tx}$ such that $\boldsymbol{\pi}(\boldsymbol{TL}(\boldsymbol{I}_{2m}\otimes \boldsymbol{T}^{\mathrm{T}})\boldsymbol{u})$ has a strictly lower triangular form.

Open-loop control design technique[1]

Example:

$$x_1(t+1) = \neg x_2(t),$$

$$x_2(t+1) = \neg x_4(t) \leftrightarrow ((x_4(t) \wedge (x_2(t) \bar{\nabla} x_3(t))) \vee u(t)),$$

$$x_3(t+1) = \neg((x_4(t) \wedge (x_2(t) \bar{\nabla} x_3(t))) \vee u(t)),$$

$$x_4(t+1) = (x_4(t) \vee (x_2(t) \bar{\nabla} x_3(t))) \wedge u(t).$$

- The **coordinate transformation**:

$$z_1 = x_4, \quad z_2 = x_2 \bar{\nabla} x_3, \quad z_3 = \neg x_3, \quad z_4 = \neg x_1.$$

$$\Rightarrow$$

$$x_1 = \neg z_4, \quad x_2 = z_2 \leftrightarrow z_3, \quad x_3 = \neg z_3, \quad x_4 = z_1.$$

$$\Rightarrow$$

$$z_1(t+1) = (z_1(t) \vee z_2(t)) \wedge u(t),$$

$$z_2(t+1) = \neg z_1(t),$$

$$z_3(t+1) = (z_1(t) \wedge z_2(t)) \vee u(t),$$

$$z_4(t+1) = z_2(t) \leftrightarrow z_3(t).$$

Open-loop control design technique[1]

- Choose $\boldsymbol{u(t)}=0$

$$
\begin{aligned}
z_1(t+1) &= 0,\\
z_2(t+1) &= \neg z_1(t),\\
z_3(t+1) &= (z_1(t)\wedge z_2(t)),\\
z_4(t+1) &= z_2(t)\leftrightarrow z_3(t).
\end{aligned}
\;\Rightarrow\;
\mathscr{I}(F)=\begin{bmatrix}0&0&0&0\\1&0&0&0\\1&1&0&0\\0&1&1&0\end{bmatrix}
\;\overset{\text{Th2}}{\Rightarrow}
$$

System is **stabilizable**.

Advantage: the size of the involved matrix is ***small*** via metric-based analysis.

Disadvantage: the condition is ***only a sufficient one***.

Open-loop control design technique[1]

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

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

- Split $L[(I_{2^m} \otimes L)\Phi_m]^{m-1} := [L_1^k L_2^k \dots L_{2^m}^k]$

Theorem 3: System (1) 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 $u = \delta_{2^m}^j$.

Open-loop control design technique[1]

- We briefly discuss the case when the system is required to ***converge to a particular state x*.***
- In addition to above stability requirements, we need to ***assure that x* is a fixed point*** of the control system.

Theorem 4: System (2) is globally stabilized to ***x**** by an open-loop control ***u***(***t***), *t* =1,2, . . . , ***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 an $u_e \in \Delta_{2^m}$ such that $\boldsymbol{L u_e x^* = x^*}$ holds.

Stabilization → **Existing methods**

- **Open-loop control design technique**
- **State feedback control design technique**
  - **Reachability set**
  - **Control Lyapunov function**
  - **Pinning control**
  - **Event-triggered control**
  - **Sampled-data control**
- **Output feedback control design technique**

State feedback control design technique[1]

- If $u(t)=Gx(t)$, $t=1,2,\ldots,$ $G \in \mathcal{L}_{2^m \times 2^n}$, then the control is called the **state feedback control**.

- $$x(t+1)=LGx^2(t)=LG\Phi_n x(t), \qquad (3)$$

where G is **state feedback gain matrix** and Φ_n is the **power-reducing matrix**.

State feedback control design technique[1]

Theorem 5: System (1) (or its algebraic form (2)) is **stabilizable** by a **closed-loop control** $\boldsymbol{u}=\boldsymbol{Gx}$, ***if*** $\boldsymbol{\pi}(\boldsymbol{LG\Phi_n})$ has a strictly lower (or upper) triangular form. Moreover, **if** there exists a coordinate transformation $\boldsymbol{z}=\boldsymbol{Tx}$ such that $\boldsymbol{\pi}(\boldsymbol{TLG\Phi_n T^{\mathrm{T}}})$ has a strictly lower triangular form, then the control also stabilizes the system.

Theorem 6: System (2) is **stabilizable** by a **closed-loop control** $\boldsymbol{u}=\boldsymbol{Gx}$, ***iff*** there exists a $2^m\times 2^n$ logical matrix G and an integer $1\leq k\leq 2n$ such that $(LG\Phi_n)^k$ is a matrix of constant mapping.

State feedback control design technique→**Reachable set**

- The control design technique based on reachable set was proposed by Li et al.[2], Fornasini and Valcher[3].
- Let $\boldsymbol{E_k(r)}$ denote the set consisting of all the initial states that can be steered to $\delta_{2^n}^{r}$ in $\boldsymbol{k}$ steps by a control input sequence $\boldsymbol{u}(0)$, $\boldsymbol{u}(1)$, …, $\boldsymbol{u}(k\text{-}1)$.

$$E_k(r) = \{\mathbf{x}_0 \in \Delta_{2^n} : \text{there are } \mathbf{u}(0), \ldots, \mathbf{u}(k-1) \in \Delta_{2^m} \text{ such that } \mathbf{x}(k; \mathbf{x}_0; \mathbf{u}(0), \ldots, \mathbf{u}(k-1)) = \delta_{2^n}(r)\}$$

---

[2] R. Li, M. Yang, and T. Chu, State feedback stabilization for Boolean control networks, *IEEE Transactions on Automatic Control*, 2017, 58: 1853-1857.
[3] E. Fornasini and M. E. Valcher, On the periodic trajectories of Boolean control networks, *Automatica*, 2013, 49: 1506-1509.

State feedback control design technique→**Reachable set**[2]

Lemma 1: If $\delta_{2^n}(r) \in E_1(r)$, then $E_k(r) \subseteq E_{k+1}(r)$ for all $k \geq 1$.

Lemma 2:

1) If $E_1(r) = \{\delta_{2^n}(r)\}$, then $E_k(r) = \{\delta_{2^n}(r)\}$ for all $k \geq 1$.
2) If $E_{j+1}(r) = E_j(r)$ for some $j \geq 1$, then $E_k(r) = E_j(r)$ for all $k \geq j$.

Theorem 1: Let $\mathbf{X}^* = (\sigma_1, \ldots, \sigma_n) \in \{1, 0\}^n$ and let r be the integer such that $\boldsymbol{\sigma}_1 \ltimes \cdots \ltimes \boldsymbol{\sigma}_n = \delta_{2^n}(r)$. If the BCN (1) can be globally stabilized to $\mathbf{X}^*$ by a state feedback controller of the form (2), then:

1) $\delta_{2^n}(r) \in E_1(r)$;
2) there exists an integer $1 \leq N \leq 2^n - 1$ such that $E_N(r) = \Delta_{2^n}$.

State feedback control design technique→**Reachable set**[2]

Lemma 3: Let $\alpha_1, \alpha_2, \ldots, \alpha_{2m+n} \in \{1, 2, \ldots, 2^n\}$ be such that

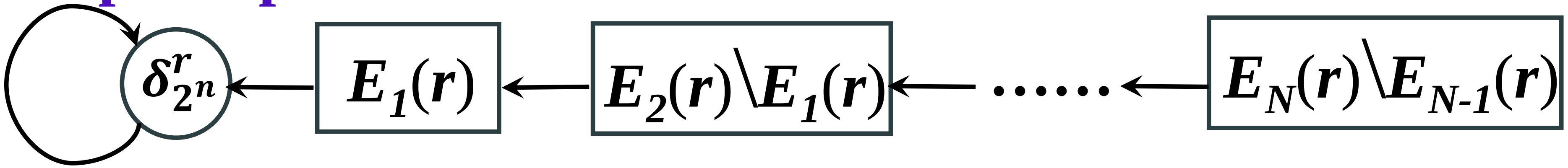
$$\boldsymbol{L} = \delta_{2^n}(\alpha_1, \alpha_2, \ldots, \alpha_{2m+n}).$$

Then for every $1 \leq r \leq 2^n$:

1) $E_1(r) = \{\delta_{2^n}(q) : 1 \leq q \leq 2^n, \alpha_{(p-1)2^n+q} = r$ for some $1 \leq p \leq 2^m\}$;
2) $E_{k+1}(r) = \bigcup\{E_1(r') : 1 \leq r' \leq 2^n, \delta_{2^n}(r') \in E_k(r)\}$ for $k = 1, 2, 3, \ldots$.

State feedback control design technique→**Reachable set**[2]

- The **principle** of reachable set:



Theorem 2: Let $\mathbf{X}^* = (\sigma_1, \ldots, \sigma_n) \in \{1,0\}^n$ and let r be the integer such that $\boldsymbol{\sigma}_1 \ltimes \cdots \ltimes \boldsymbol{\sigma}_n = \delta_{2^n}(r)$. Suppose that conditions 1) and 2) in Theorem 1 hold. To every $1 \le i \le 2^n$ corresponds a unique integer $1 \le l_i \le N$ such that $\delta_{2^n}(i) \in E_{l_i}(r)\backslash E_{l_i-1}(r)$, where $E_0(r) = \emptyset$, and let $1 \le p_i \le 2^m$ be such that $\alpha_{(p_i-1)2^n+i} = r$ if $l_i = 1$, and $\delta_{2^n}(\alpha_{(p_i-1)2^n+i}) \in E_{l_i-1}(r)$ if $l_i \ge 2$. Then the feedback law (2) with the state feedback matrix K given by

$$K = \delta_{2^m}(p_1, p_2, \ldots, p_{2^n})$$

State feedback control design technique→**Reachable set**[2]

Example:

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

- ***Objective***: designing state feedback gain matrices G which makes system globally stabilizable to X*=(1,0,1)~ $\boldsymbol{\delta_8^3}$.
- ***Algebraic form***: $\mathbf{x}(t+1) = F \ltimes \mathbf{u}(t) \ltimes \mathbf{x}(t)$

$$F = \delta_8(8,8,8,8,8,8,8,8,8,8,8,8,8,8,8,8,8,8,8,8,8,\\ 8,8,8,8,8,8,8,8,8,8,8,1,1,1,5,3,3,3,7,1,1,1,\\ 5,3,3,3,7,3,3,3,7,4,4,4,8,4,4,4,8,4,4,4,8).$$

State feedback control design technique→**Reachable set**[2]

- $$E_1(3) = \{\delta_8(i) : i = 1, 2, 3, 5, 6, 7\}.$$

  $$E_2(3) \supseteq (E_1(3) \cup E_1(7)) = \Delta_8.$$

  Hence, conditions 1) and 2) in Theorem 1 are satisfied.

- $$l_1 = l_2 = l_3 = l_5 = l_6 = l_7 = 1, \qquad l_4 = l_8 = 2$$

  $$p_1 = p_2 = p_3 = 7, \qquad p_4 = p_5 = p_6 = p_7 = p_8 = 5.$$

Then the feedback law with the state feedback matrix ***K*** given by

$$K = \delta_8(7, 7, 7, 5, 5, 5, 5, 5)$$

globally stabilizes the BCN to ***X****.

State feedback control design technique→**Reachable set**[3]

Definition [4] A BCN (2) is *stabilizable* to the elementary cycle $\mathcal{C} = (\delta_{2^n}^{i_1}, \delta_{2^n}^{i_2}, \ldots, \delta_{2^n}^{i_k})$ if for every $\mathbf{x}(0) \in \mathcal{L}_{2^n}$ there exist $\mathbf{u}(t)$, $t \in \mathbb{Z}_+$, and $\tau \in \mathbb{Z}_+$ such that $\mathbf{x}(t) = \delta_{2^n}^{i_j}$ for every $t \geq \tau$, where $j \in [1, k]$ and $j \equiv (t - \tau + 1) \bmod k$.

[3] E. Fornasini and M. E. Valcher, On the periodic trajectories of Boolean control networks, *Automatica*, 2013, 49: 1506-1509.

State feedback control design technique→**Reachable set** [3]

Proposition *A BCN is stabilizable to the elementary cycle* $\mathfrak{C} = (\delta_{2^n}^{i_1}, \delta_{2^n}^{i_2}, \ldots, \delta_{2^n}^{i_k})$ *if and only if the following two conditions hold*

(1) *for every* $(i_\ell, i_{\ell+1}), \ell \in [1, k]$, *(with* $i_{k+1} = i_1$*) there exists* $\delta_{2^m}^{j_\ell}$ *such that* $\delta_{2^n}^{i_{\ell+1}} = L \ltimes \delta_{2^m}^{j_\ell} \ltimes \delta_{2^n}^{i_\ell} = L_{j_\ell}\delta_{2^n}^{i_\ell}$;

(2) $\delta_{2^n}^{i_1}$ *is reachable from every initial state* $\mathbf{x}(0)$, *which amounts to saying that*

$$\delta_{2^n}^{i_1} \in \bigcap_{\mathbf{x}(0)\in\mathcal{L}_{2^n}} \mathcal{R}(\mathbf{x}(0)).$$

State feedback control design technique→**Control Lyapunov function**[4]

- ***Objective***: design **all possible state feedback gain matrices *G*** which makes system (1) globally stabilizable to ***X****.

Definition 1:

Given an equilibrium $x^* = \delta_{k^n}^r \in \Delta_{k^n}$. $V(x) : \Delta_{k^n} \to \mathbb{R}$ is called a control Lyapunov function of system (2), if

(i) there exists $u^* \in \Delta_{k^m}$ such that $V(Lu^*x^*) - V(x^*) = 0$;

(ii) for any $x \in \Delta_{k^n}$ satisfying $x \neq x^*$, there exists $u_x \in \Delta_{k^m}$ such that $V(Lu_x x) - V(x) > 0$.

[4] H. Li and X. Ding, A control Lyapunov function approach to feedback stabilization of logical control networks, *SIAM Journal on Control and Optimization*, 2019, 57(2): 810-831.

State feedback control design technique→**Control Lyapunov function**[4]

Proposition 1:

Condition (***i***) of ***Definition*** 1 is equivalent to $\mathbf{L}\boldsymbol{u}^*\boldsymbol{x}^* = \boldsymbol{x}^*$.

Proof. Obviously, one can see from $Lu^*x^* = x^*$ that $V(Lu^*x^*) - V(x^*) = 0$. Now, we prove that $Lu^*x^* = x^*$ holds when $V(Lu^*x^*) - V(x^*) = 0$.

In fact, if $Lu^*x^* \neq x^*$, letting $x^1 = Lu^*x^* \neq x^*$, by condition (ii) of Definition 1, there exists $u_{x^1} \in \Delta_{k^m}$ such that $V(Lu_{x^1}x^1) - V(x^1) > 0$, which together with $V(x^1) = V(x^*)$ shows that $V(Lu_{x^1}x^1) > V(x^*)$. Now, letting $x^2 = Lu_{x^1}x^1$, one can see that $x^2 \neq x^1$, $x^2 \neq x^*$, and there exists $u_{x^2} \in \Delta_{k^m}$ such that $V(Lu_{x^2}x^2) > V(x^2) > V(x^*)$. To keep this procedure going, we obtain a sequence of states $\{x^i : i \in \mathbb{Z}_+\}$ satisfying $x^i \neq x^j \ \forall \ i \neq j$. This is a contradiction to the fact system (2) has k^n different states. Hence, $Lu^*x^* = x^*$ holds when $V(Lu^*x^*) - V(x^*) = 0$. □

State feedback control design technique→**Control Lyapunov function**[4]

Theorem 1:
System (2) is globally stabilizable to $x^* = \delta_{k^n}^r$ by a state feedback control in the form of $u(t) = Gx(t)$, if and only if system (2) has a control Lyapunov function $V(x) = M_V x$, where $M_V := [a_1 \ \ a_2 \ \ \cdots \ \ a_{k^n}] \in \mathbb{R}^{1\times k^n}$ is the structural matrix of $V(x)$.

- The ***closed-loop*** system: $$\begin{cases} x(t+1) = \widehat{L} \ltimes x(t), \\ y(t) = Hx(t), \end{cases}$$

where $\widehat{L} = LGM_{r,k^n}$.

For any $x_0 = \delta_{k^n}^i \in \Delta_{k^n}$, let $c_i = \min\{t : \widehat{L}^t x_0 = x^*, t \in \mathbb{N}\}$.

⇒***CLF***: $V(x) = [a_1 \ \ a_2 \ \ \cdots \ \ a_{k^n}]x$, $a_i = -c_i$, $i = 1, 2, \ldots, k^n$.

State feedback control design technique→**Control Lyapunov function**[4]

➢ **Notations**:

- For any $\alpha, \beta \in \{1, 2, \ldots, k^n\}$ with $\alpha \neq \beta$, $\{a_\beta > a_\alpha\}|_{U_{\alpha,\beta}}$ means that the inequality $a_\beta > a_\alpha$ holds only when $u \in U_{\alpha,\beta}$, where $U_{\alpha,\beta} = \{u : \delta_{k^n}^{\beta} = Lu\delta_{k^n}^{\alpha}, u \in \Delta_{k^m}\}$ denotes the set of controls under which the state $\delta_{k^n}^{\beta}$ is reachable from the state $\delta_{k^n}^{\alpha}$ in one step.
- $\{a_r = a_r\}|_{U_{r,r}}$ means the equality $a_r = a_r$ holds only when $u \in U_{r,r}$, where $U_{r,r} = \{u : \delta_{k^n}^{r} = Lu\delta_{k^n}^{r}, u \in \Delta_{k^m}\}$.
- For any $\alpha, \beta \in \{1, 2, \ldots, k^n\}$, define

$$Z_{\alpha\to\beta} := \begin{cases} \{a_\beta > a_\alpha\}|_{U_{\alpha,\beta}} \text{ if } \alpha \neq \beta, U_{\alpha,\beta} \neq \emptyset, \\ \{a_r = a_r\}|_{U_{r,r}} \text{ if } \alpha = \beta = r, U_{r,r} \neq \emptyset, \\ \emptyset \text{ otherwise.} \end{cases}$$

  Then, one can see that $Z_{\alpha\to\beta}$ denotes an inequality/equality (maybe empty) determined by the one step reachability from $\delta_{k^n}^{\alpha}$ to $\delta_{k^n}^{\beta}$.
- Let $R_i(x^*)$ be the set of initial states which can reach x^* at the ith step.

State feedback control design technique→**Control Lyapunov function**[4]

- **Define**

$$(S)_{\beta,\alpha} := \begin{cases} 1 \text{ if } Z_{\alpha\to\beta} \neq \emptyset, \\ 0 \text{ if } Z_{\alpha\to\beta} = \emptyset. \end{cases}$$

Proposition 2:

$x_d = \delta_{k^n}^{\beta}$ is τ-step reachable from $x_0 = \delta_{k^n}^{\alpha}$ if and only if

$$(S^{\tau})_{\beta,\alpha} > 0.$$

Theorem 2:

System (2) has a control Lyapunov function, if and only if there exists a positive integer $\tau \leq k^n$ such that

$$Row_r(S^{\tau}) > 0.$$

State feedback control design technique→**Control Lyapunov function** [4]

- **Define**

$$\Pi_1(r) = \{\alpha : Z_{\alpha\to r} \neq \emptyset, \alpha \in \{1, 2, \dots, k^n\}\}.$$

for any $k = 2, 3, \dots,$

$$\Pi_k(r) = \{\alpha : \exists\ \alpha' \in \Pi_{k-1}(r) \text{ such that } Z_{\alpha\to\alpha'} \neq \emptyset, \alpha \in \{1, 2, \dots, k^n\}\}.$$

Proposition 3: $R_t(x^*) = \{\delta_{k^n}^{\alpha} : \alpha \in \Pi_t(r)\}.$

Theorem 3:

System (2) has a control Lyapunov function, if and only if there exists a positive integer $\tau \leq k^n$ such that

$$\begin{cases} r \in \Pi_1(r), \\ \Pi_\tau(r) = \{1, 2, \cdots, k^n\}. \end{cases} \quad (4)$$

State feedback control design technique→**Control Lyapunov function** [4]

- Assume that (4) of Theorem 3 holds. Define $\Lambda_0 := \{Z_{r\to r}\}, \quad \widetilde{\Lambda}_0 = \{r\}$.

- We arbitrarily choose a nonempty set of $\{Z_{\alpha\to r} : \alpha \in \Pi_1(r) \setminus \widetilde{\Lambda}_0\}$ denoted by Λ_1. Set $\widetilde{\Lambda}_1 = \{\alpha : Z_{\alpha\to r} \in \Lambda_1\}$.

- For this $\widetilde{\Lambda}_1$, we arbitrarily choose a nonempty set of

$$\left\{Z_{\alpha\to\alpha'} : \alpha \in \Pi_2(r) \setminus (\widetilde{\Lambda}_0 \bigcup \widetilde{\Lambda}_1), \alpha' \in \widetilde{\Lambda}_1\right\},$$

denoted by Λ_2, which satisfies the following condition: $\alpha_1 \neq \alpha_2$ holds for any $Z_{\alpha_1\to\alpha_1'} \in \Lambda_2$ and any $Z_{\alpha_2\to\alpha_2'} \in \Lambda_2$. Set $\widetilde{\Lambda}_2 = \{\alpha : Z_{\alpha\to\alpha'} \in \Lambda_2\}$.

State feedback control design technique→**Control Lyapunov function**[4]

- Generally, for an arbitrarily chosen nonempty set

$$\Lambda_s \subseteq \left\{ Z_{\alpha\to\alpha'} : \alpha \in \Pi_s(r) \setminus \left(\bigcup_{i=0}^{s-1} \widetilde{\Lambda}_i \right), \alpha' \in \widetilde{\Lambda}_{s-1} \right\}, s \geq 2$$

which satisfies the following condition: $\alpha_1 \neq \alpha_2$ holds for any $Z_{\alpha_1\to\alpha_1'} \in \Lambda_s$ and any $Z_{\alpha_2\to\alpha_2'} \in \Lambda_s$. Set $\widetilde{\Lambda}_s = \{\alpha : Z_{\alpha\to\alpha'} \in \Lambda_s\}$.

- For the aforementioned $\widetilde{\Lambda}_i$, $i = 0, 1, \ldots, s$, we arbitrarily choose a nonempty set

$$\Lambda_{s+1} \subseteq \left\{ Z_{\alpha\to\alpha'} : \alpha \in \Pi_{s+1}(r) \setminus \left(\bigcup_{i=0}^{s} \widetilde{\Lambda}_i \right), \alpha' \in \widetilde{\Lambda}_s \right\}$$

which satisfies the following condition: $\alpha_1 \neq \alpha_2$ holds for any $Z_{\alpha_1\to\alpha_1'} \in \Lambda_{s+1}$ and any $Z_{\alpha_2\to\alpha_2'} \in \Lambda_{s+1}$. Set $\widetilde{\Lambda}_{s+1} = \{\alpha : Z_{\alpha\to\alpha'} \in \Lambda_{s+1}\}$.

State feedback control design technique→**Control Lyapunov function**[4]

- Obviously, $\widetilde{\Lambda}_i \cap \widetilde{\Lambda}_j = \emptyset,\ \forall\, i \neq j$. Therefore, one can find a positive integer $\tau \leq k^n$ such that

$$\left\{ Z_{\alpha \to \alpha'} : \alpha \in \Pi_\tau(r) \setminus \left(\bigcup_{i=0}^{\tau-1} \widetilde{\Lambda}_i \right), \alpha' \in \widetilde{\Lambda}_{\tau-1} \right\} \neq \emptyset$$

and

$$\left\{ Z_{\alpha \to \alpha'} : \alpha \in \Pi_{\tau+1}(r) \setminus \left(\bigcup_{i=0}^{\tau} \widetilde{\Lambda}_i \right), \alpha' \in \widetilde{\Lambda}_\tau \right\} = \emptyset.$$

In this case, one cannot choose a nonempty set $\Lambda_{\tau+1}$. Hence, we obtain a set of inequalities $\bigcup_{i=0}^{\tau} \Lambda_i$. $\bigcup_{i=0}^{\tau} \Lambda_i$ *is said to be an admissible set of control Lyapunov inequalities if* $\bigcup_{i=0}^{\tau} \widetilde{\Lambda}_i = \{1, 2, \ldots, k^n\}$.

- Obviously, **only the admissible set of control Lyapunov inequalities can determine state feedback controls**. Denote all of the admissible sets of control Lyapunov inequalities by $\boldsymbol{\Psi}_j$, $j = 1, \ldots, l$.

State feedback control design technique→**Control Lyapunov function**[4]

- For any $j = 1, 2, \ldots, l$, define

$$\Phi_j = \left\{ G = \delta_{k^m}[\mu_1\ \mu_2\ \cdots\ \mu_{k^n}] : \mu_\alpha \in U_{\alpha,\alpha'}, \alpha = 1, 2, \ldots, k^n \right\}.$$

Theorem 4:

Consider system (2). Assume that (4) of Theorem 3 holds. The set consisting of all the state feedback stabilizers is

$$\Phi = \bigcup_{j=1}^{l} \Phi_j.$$

State feedback control design technique→**Control Lyapunov function**[4]

Example: Consider system (2) with

$$L=\delta_4[1\ 3\ 3\ 3\ 2\ 2\ 1\ 3\ 4\ 3\ 1\ 4\ 3\ 3\ 3\ 2]$$

and $H=\delta_2[1\ 2\ 2\ 1]$. Our objective is to design all possible state feedback stabilizers under which system (2) is globally stabilizable to $x^*=\delta_4^1$.

- $$Z_{1\to 1}=\{a_1=a_1\}\,|_{\{\delta_4^1\}},\ Z_{1\to i}=\emptyset,\ \ i=2,3,4,$$
  $$Z_{2\to 3}=\{a_3>a_2\}\,|_{\{\delta_4^1,\delta_4^3,\delta_4^4\}},\ Z_{2\to i}=\emptyset,\ \ i=1,2,4,$$
  $$Z_{3\to 1}=\{a_1>a_3\}\,|_{\{\delta_4^2,\delta_4^3\}},\ Z_{3\to i}=\emptyset,\ \ i=2,3,4,$$
  $$Z_{4\to 2}=\{a_2>a_4\}\,|_{\{\delta_4^4\}},\ Z_{4\to 3}=\{a_3>a_4\}\,|_{\{\delta_4^1,\delta_4^2\}},$$
  $$Z_{4\to i}=\emptyset,\ \ i=1,4.$$
- $\Pi_1(1)=\{1,3\},\quad \Pi_\tau(1)=\{1,2,3,4\},\ \ \tau\geq 2.$
- By Theorem 3, system (2) can be **globally stabilizable** to $x^*=\boldsymbol{\delta_4^1}$ by state feedback control.

State feedback control design technique→**Control Lyapunov function**[4]

➢ Through a simple computation, one can obtain that $\Lambda_0 = \{Z_{1\to 1}\}$, $\widetilde{\Lambda}_0 = \{1\}$; $\Lambda_1 = \{Z_{3\to 1}\}$, $\widetilde{\Lambda}_1 = \{3\}$; $\Lambda_2 \subseteq \{Z_{2\to 3}, Z_{4\to 3}\}$. For the choice of Λ_2, we have the following three cases.

- When $\Lambda_2 = \{Z_{2\to 3}\}, \widetilde{\Lambda}_2 = \{2\}$, we can obtain $\Lambda_3 = \{Z_{4\to 2}\}, \widetilde{\Lambda}_3 = \{4\}$. Since $\bigcup_{i=0}^{3} \widetilde{\Lambda}_i = \{1, 2, 3, 4\}$,

$$\Psi_1 = \Big\{\{a_1 = a_1\}\,|_{\{\delta_4^1\}}, \{a_1 > a_3\}\,|_{\{\delta_4^2, \delta_4^3\}}, \{a_3 > a_2\}\,|_{\{\delta_4^1, \delta_4^3, \delta_4^4\}}, \{a_2 > a_4\}\,|_{\{\delta_4^4\}}\Big\}$$

$$\Phi_1 = \Big\{G = \delta_4[\mu_1 \;\; \mu_2 \;\; \mu_3 \;\; \mu_4] : \mu_1 \in \{1\}, \mu_2 \in \{1, 3, 4\}, \mu_3 \in \{2, 3\}, \mu_4 \in \{4\}\Big\}.$$

State feedback control design technique→**Control Lyapunov function**[4]

- When $\Lambda_2 = \{Z_{4\to 3}\}, \widetilde{\Lambda}_2 = \{4\}$, we have $\Lambda_3 = \emptyset$. Since $\bigcup_{i=0}^{2} \widetilde{\Lambda}_i = \{1,3,4\} \neq \{1,2,3,4\}$, $\bigcup_{i=0}^{2} \Lambda_i$ is an inadmissible set of control Lyapunov inequalities. Hence, we cannot find any state feedback stabilizer for this case.
- When $\Lambda_2 = \{Z_{2\to 3}, Z_{4\to 3}\}, \widetilde{\Lambda}_2 = \{2,4\}$, one can see that $\bigcup_{i=0}^{2} \widetilde{\Lambda}_i = \{1,2,3,4\}$, and thus $\bigcup_{i=0}^{2} \Lambda_i$ is an admissible set of control Lyapunov inequalities. Therefore,

$$\Psi_2 = \Big\{\{a_1 = a_1\}\,|_{\{\delta_4^1\}}, \{a_1 > a_3\}\,|_{\{\delta_4^2,\delta_4^3\}}, \{a_3 > a_2\}\,|_{\{\delta_4^1,\delta_4^3,\delta_4^4\}}, \{a_3 > a_4\}\,|_{\{\delta_4^1,\delta_4^2\}}\Big\}$$

and

State feedback control design technique→**Control Lyapunov function**[4]

$$\Phi_2 = \Big\{ G = \delta_4[\mu_1 \ \ \mu_2 \ \ \mu_3 \ \ \mu_4] : \mu_1 \in \{1\},$$
$$\mu_2 \in \{1,3,4\}, \mu_3 \in \{2,3\}, \mu_4 \in \{1,2\} \Big\}.$$

➢ To sum up, we totally obtain the following 18 state feedback stabilizers which belong to $\Phi = \Phi_1 \bigcup \Phi_2$:

$$\Phi = \Big\{ G = \delta_4[\mu_1 \ \ \mu_2 \ \ \mu_3 \ \ \mu_4] : \mu_1 \in \{1\},$$
$$\mu_2 \in \{1,3,4\}, \mu_3 \in \{2,3\}, \mu_4 \in \{1,2,4\} \Big\}.$$

Pinning Control of Networks

Pinning control of complex networks

$$\dot{x}_i(t) = f(x_i(t)) + c\sum_{j=1}^{N} a_{ij}\Gamma x_j(t) + u_i(t), \quad i = 1, 2, \ldots, N$$

with $u_i(t) = \begin{cases} -c \cdot a \cdot \Gamma x_i(t), & i = 1, 2, \ldots, l; \\ 0, & \text{otherwise}; \end{cases}$

Only **small fraction** of individual nodes are **directly** controlled

Pinning controlled BNs [12]

$$\begin{cases} x_1(t+1) = f_1(x_1(t), \ldots, x_n(t), u_1(t)) \\ \cdots \\ x_r(t+1) = f_r\left(x_1(t), \ldots, x_n(t), u_r(t)\right) \\ x_{r+1}(t+1) = f_{r+1}\left(x_1(t), \ldots, x_n(t)\right) \\ \cdots \\ x_n(t+1) = f_n\left(x_1(t), \ldots, x_n(t)\right) \end{cases}$$

What is pinning control?

[5] F. Li, Pinning control design for the stabilization of Boolean networks, *IEEE Transactions on Neural Networks & Learning Systems*, 2016, 27(7): 1585-1590.
[14] J. Lu* et al. On Pinning Controllability of Boolean Control Networks, IEEE Trans. Automatic Control, 61(6): 1658-1663, 2016.

Pinning Controllability of Complex Networks

Proposition 1 [15-18]**:** The real part of the maximum eigenvalue of coupling matrix changes from zero to be negative under control:

$$A = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1N} \\ a_{21} & a_{22} & \cdots & a_{2N} \\ \vdots & \vdots & \ddots & \vdots \\ a_{N1} & a_{N2} & \cdots & a_{NN} \end{bmatrix} \longrightarrow \tilde{A} = \begin{bmatrix} a_{11}\textbf{-a} & a_{12} & \cdots & a_{1N} \\ a_{21} & a_{22} & \cdots & a_{2N} \\ \vdots & \vdots & \ddots & \vdots \\ a_{N1} & a_{N2} & \cdots & a_{NN} \end{bmatrix}$$

How to explain pinning control from mathematical viewpoint ?

[15] O. Taussky, A recurring theorem on determinants, American Math. Mon. 10:672-676, 1949.
[16] C.W. Wu, Synchronization in networks of nonlinear dynamical systems coupled via a directed graph, Nonlinearity 18:1057-1064, 2005.
[17] X. Li, X. Wang, and G. Chen, Pinning a complex dynamical network to its equilibrium, IEEE Trans. Circuits Syst.I, 51(10):2074-2087, 2004.
[18] T. Chen, X. Liu, and W. Lu, Pinning complex networks by a single controller, IEEE Trans. Circuits Syst. I, Reg. Papers,54(6):1317-1326, 2007.

Pinning Controllability of Complex Networks

ARTICLE

doi:10.1038/nature10011

Controllability of complex networks

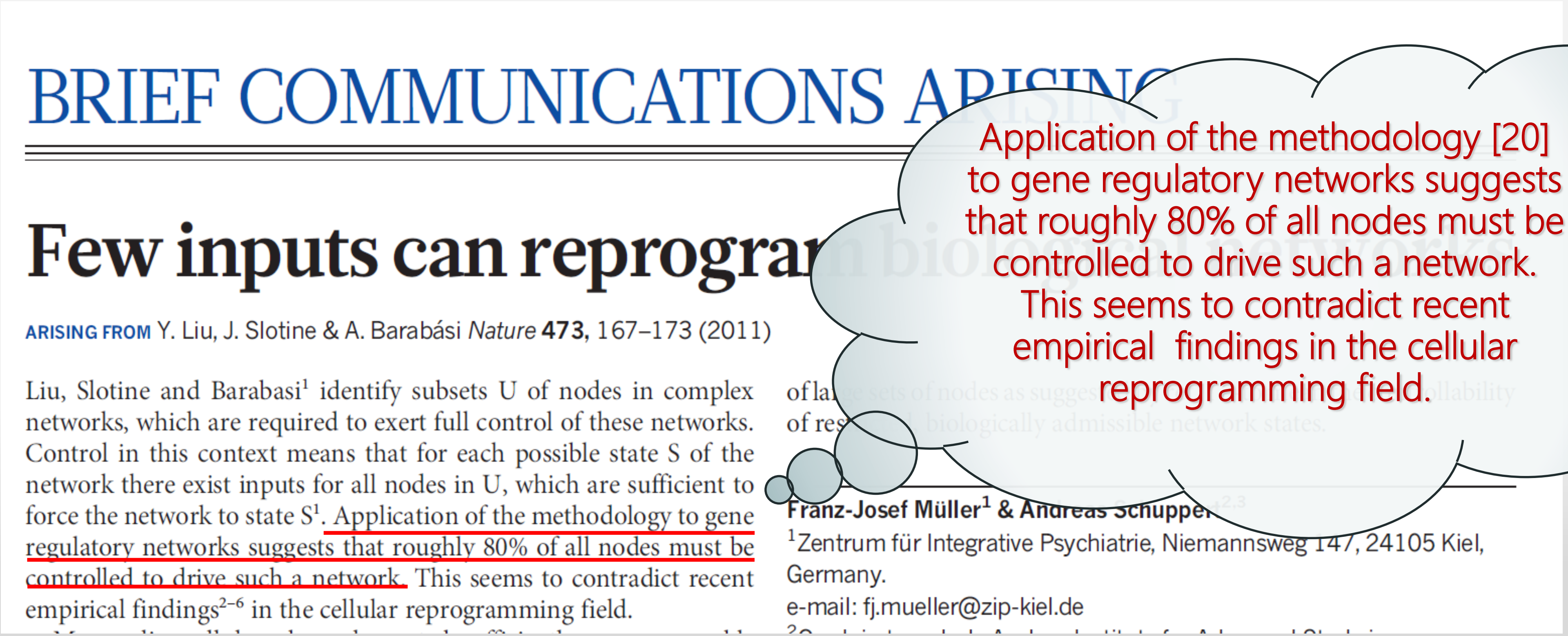
Yang-Yu Liu[1,2], Jean-Jacques Slotine[3,4] & Albert-László Barabási[1,2,5]

The ultimate proof of our understanding of natural or technological systems is reflected in our ability to control them. Although control theory offers mathematical tools for steering engineered and natural systems towards a desired state, a framework to control complex self-organized systems is lacking. Here we develop analytical tools to study the controllability of an arbitrary complex directed network, identifying the set of driver nodes with time-dependent control that can guide the system's entire dynamics. We apply these tools to several real networks, finding that the number of driver nodes is determined mainly by the network's degree distribution. We show that sparse inhomogeneous networks, which emerge in many real complex systems, are the most difficult to control, but that dense and homogeneous networks can be controlled using a few driver nodes. Counterintuitively, we find that in both model and real systems the driver nodes tend to avoid the high-degree nodes.

Why do we need pinning control and how to select pinning nodes?

[19] Y.Y. Liu, *et. al*. Controllability of complex networks. Nature, 473(7346):167-173, 2011.
[20] F. J. Muller, *et. al*. Few inputs can reprogram biological networks. Nature, 478(7369):E4, 2011.

Pinning Controllability of BNs



BRIEF COMMUNICATIONS ARISING

Few inputs can reprogra

ARISING FROM Y. Liu, J. Slotine & A. Barabási *Nature* **473**, 167–173 (2011)

Liu, Slotine and Barabasi[1] identify subsets U of nodes in complex networks, which are required to exert full control of these networks. Control in this context means that for each possible state S of the network there exist inputs for all nodes in U, which are sufficient to force the network to state S[1]. Application of the methodology to gene regulatory networks suggests that roughly 80% of all nodes must be controlled to drive such a network. This seems to contradict recent empirical findings[2–6] in the cellular reprogramming field.

Franz-Josef Müller[1] & Andreas Schupper

[1]Zentrum für Integrative Psychiatrie, Niemannsweg 147, 24105 Kiel, Germany.

e-mail: fj.mueller@zip-kiel.de

Application of the methodology [20] to gene regulatory networks suggests that roughly 80% of all nodes must be controlled to drive such a network. This seems to contradict recent empirical findings in the cellular reprogramming field.

[19] Y.Y. Liu, *et. al*. Controllability of complex networks. Nature, 473(7346):167-173, 2011.

[20] F. J. Muller, *et. al*. Few inputs can reprogram biological networks. Nature, 478(7369):E4, 2011.

Pinning Controllability of BNs

Pinning control is effective for biological networks

- Much empirical evidences show that few inputs can well capture the features or fully control a biological system [21, 22].
- ...we have devised a practical control method that can be implemented at a single node or link to force the system . . . . In particular, we have identified two important nodes, Wip1 and Mdm2, that are the most effective for this control. [23].
- . . . that the number of nodes (five or fewer genes out of about 30,000) needed to fully control a biological system . . . .[24].
- . . . the control of the synchronization patterns locally by a small fraction of the network nodes by adjusting the refractory time of only 2 out of 32 nodes [25].

[21] M. Ieda, *et. al.* Direct reprogramming of fibroblasts into functional cardiomyocytes by defined factors. New Cell,4248(3):670-679, 2007.

[22] F. J. Muller, *et. al.* A bioinformatic assay for pluripotency in human cells. Nature Methods, 8(4):315-317, 2011.

[23] G. Q. Lin, *et. al.* Modeling and controlling the two-phase dynamics of the p53 network: A Boolean network approach. New Journal of Physics, 16(12):125010, 2014.

[24] F. J. Muller, *et. al.* Few inputs can reprogram biological networks. Nature, 478(7369):E4, 2011.

[25] D. P. Rosin, *et. al.* Control of synchronization patterns in neural-like Boolean networks. Physical Review Letters,110(10):104102, 2013.

State feedback control design technique→**Pinning control**[5]

- The control based on reachable set or control Lyapunov function are applied to all the nodes or applied randomly to some nodes of the BN. The cost of control to apply control to all the nodes may be **greater than** just apply to a fraction of nodes.
- Furthermore, it may not achieve the control objective to apply the control randomly to some nodes, because it may control the wrong nodes.
- In this brief, we consider the pinning control design for the stabilization of BNs. **Our purpose** is to select a fraction of nodes by the algorithms proposed in this brief.

---

[5] F. Li, Pinning control design for the stabilization of Boolean networks, *IEEE Transactions on Neural Networks & Learning Systems*, 2016, 27(7): 1585-1590.

J.Q. Lu et al.. On pinning controllability of Boolean control networks. IEEE Transactions on Automatic Control, 61(6):1658-1663, 2016.

State feedback control design technique→**Pinning control**[5]

- BNs:

$$x(t+1) = M_1 x(t) M_2 x(t) \cdots M_n x(t) := L x(t) \quad (5)$$

- Assume that the BN with pinning control is given as

$$\begin{aligned} x_i(t+1) &= F_i(u_i(t), x_1(t), \ldots, x_n(t)), \quad i = 1, 2, \ldots, k \\ x_j(t+1) &= f_j(x_1(t), \ldots, x_n(t)), \quad j = k+1, \ldots, n \end{aligned} \quad (6)$$

where F_i, $i = 1, 2, \ldots, k$ are some logical functions and u_i, $i = 1, 2, \ldots, k$ are feedback controls. $i = 1, 2, \ldots, k$ are the pinning nodes that will be selected in the sequel. Furthermore, the control u_i should be designed

State feedback control design technique→**Pinning control**[5]

- *Algorithm 1:*

Step 1: Change the rth column of L to $\delta_{2^n}^r$.

Step 2: Denote by $E_k(r)$ the set consisting of all the initial states that can be steered to $\delta_{2^n}^r$ in k steps. That is $E_k(r) = \{x_0 \in \Delta_{2^n} : x(k; x_0) = \delta_{2^n}^r\}$. Let $\mathbf{E}(r) = \bigcup_{k=1}^{2^n} E_k(r)$ and calculate $\mathbf{E}(r)$.

Step 3: Find a $\delta_{2^n}^i \notin \mathbf{E}(r)$, $i \in \{1, 2, \ldots, 2^n\}$ and let $\mathrm{Col}_i(L)$ be the element of $\mathbf{E}(r)$.

- By doing this, ***L*** is changed to ***L'*** and the BN is **globally stable** to the fixed point.

Proposition **1:** Let $x^* = \delta_{2^n}^r$. Suppose that L is changed into L' according to Algorithm 1, then the BN (5) with the transition matrix L' is globally stable to x^*.

State feedback control design technique→**Pinning control**[5]

➤ *Algorithm 2:*

Step 1: Change the r_1th, r_2th,…, r_{k-1}th, r_kth columns of L to $\delta_{2^n}^{r_2}, \delta_{2^n}^{r_3}, \ldots, \delta_{2^n}^{r_k}, \delta_{2^n}^{r_1}$, respectively.

Step 2: Denote by $E_k(\mathscr{C}) = \{x_0 \in \Delta_{2^n} : x(k; x_0) = \delta_{2^n}^{r_1}\}$ and $\mathbf{E}(\mathscr{C}) = \bigcup_{k=1}^{2^n} E_k(\mathscr{C})$.

Step 3: Find $\delta_{2^n}^{i} \notin \mathbf{E}(\mathscr{C})$, $i \in \{1, 2, \ldots, 2^n\}$. Let the ith column of L be the element of $\mathbf{E}(\mathscr{C})$.

➤ By doing this, ***L*** **is changed to** ***L'*** and $\mathscr{C}$ becomes a **globally attractive limit cycle**.

Proposition 2 Suppose that L can be changed to L' according to Algorithm 2 then the BN (5) with the transition matrix L' is globally attractive to the limit cycle $\mathscr{C} = (\delta_{2^n}^{r_1}, \delta_{2^n}^{r_2}, \ldots, \delta_{2^n}^{r_k})$.

State feedback control design technique→**Pinning control**[5]

- Now, let us give the pinning control design for the BNs

- Assume that the transition matrix L of BN (5) is changed to L' according to Algorithm 1 Without loss of generality, we assume that the first,…, mth column of L alters and assume that 1st,…, mth column of $M_1,\ldots, M_k$ alters. We assume that $M_1,\ldots, M_k$ alter to $M'_1,\ldots, M'_k$.

- Suppose that $f_1(x_1,\ldots,x_n),\ldots,f_k(x_1,\ldots,x_n)$ be changed to $F_1(u_1,x_1,\ldots,x_n) = u_1(x_1,\ldots,x_n) \oplus_1 f_1(x_1,\ldots,x_n),\ldots,$ $F_k(u_k,x_1,\ldots,x_n) = u_k(x_1,\ldots,x_n) \oplus_k f_k(x_1,\ldots,x_n)$, respectively, where $\oplus_1,\ldots,\oplus_k$ are some logical functions, $u_1,\ldots,u_k$ are state feedback control.

State feedback control design technique→**Pinning control** [5]

- $$\begin{aligned} F_1(u_1, x_1, \ldots, x_n) &= u_1(x_1, \ldots, x_n) \oplus_1 f_1(x_1, \ldots, x_n) \\ &= M_{\oplus_1} \bar{M}_1 x_1, \ldots, x_n M_1 x_1, \ldots, x_n \\ &= M_{\oplus_1} \bar{M}_1 (I_{2^n} \otimes M_1) \Phi_n x_1, \ldots, x_n \end{aligned}$$

$$\vdots$$

- $$\begin{aligned} F_k(u_k, x_1, \ldots, x_n) &= u_k(x_1, \ldots, x_n) \oplus_k f_k(x_1, \ldots, x_n) \\ &= M_{\oplus_k} \bar{M}_k x_1, \ldots, x_n M_k x_1, \ldots, x_n \\ &= M_{\oplus_k} \bar{M}_k (I_{2^n} \otimes M_k) \Phi_n x_1, \ldots, x_n \end{aligned}$$

where $M_{\oplus_1}, \ldots, M_{\oplus_k} \in \mathcal{L}_{2\times 4}$ are the structure matrices of logical functions $\oplus_1, \ldots, \oplus_k$, respectively, $\bar{M}_1, \ldots, \bar{M}_k \in \mathcal{L}_{2\times 2^n}$ are the structure matrices of feedback control functions $u_1, \ldots, u_k$ respectively, $\Phi_n = \delta_{2^{2n}}[1, 2^n + 2, 2 \cdot 2^n + 3, \ldots, (2^n - 2)2^n + 2^n - 1, 2^{2n}]$.

State feedback control design technique→**Pinning control** [5]

- If we can solve $M_{\oplus 1}, \ldots, M_{\oplus k}, \bar{M}_1, \ldots, \bar{M}_k$ from the following equation:

$$\begin{cases} M_{\oplus 1} \bar{M}_1 (I_{2^n} \otimes M_1) \Phi_n = M'_1 \\ \vdots \\ M_{\oplus k} \bar{M}_k (I_{2^n} \otimes M_k) \Phi_n = M'_k \end{cases} \quad (7)$$

then , one can obtain the logical functions of $\oplus_1, \ldots, \oplus_k$, $u_1, \ldots, u_k$. Hence, the Boolean control network (6) is globally stable to the fixed point $\delta_{2^n}^r$ (the limit cycle $\mathscr{C}$).

State feedback control design technique→**Pinning control** [5]

- *Algorithm 3*

1) Change the columns of the transition matrix L of (5) using Algorithm *1*
2) Calculate the new structure matrices. Without loss of generality, we assume that the first,..., mth column of $M_1, \ldots, M_k$ alter to $M_1', \ldots, M_k'$.
3) Suppose that $f_1(x_1, \ldots, x_n), \ldots, f_k(x_1, \ldots, x_n)$ be changed to $F_1(u_1, x_1, \ldots, x_n) = u_1(x_1, \ldots, x_n) \oplus_1 f_1(x_1, \ldots, x_n)$,..., $F_k(u_k, x_1, \ldots, x_n) = u_k(x_1, \ldots, x_n) \oplus_k f_k(x_1, \ldots, x_n)$, respectively. Solve $M_{\oplus_1}, \ldots, M_{\oplus_k}$, $\bar{M}_1, \ldots, \bar{M}_k$ from (7). Then , one can obtain the logical functions $\oplus_1, \ldots, \oplus_k$, $u_1, \ldots, u_k$. Hence, the Boolean control network (6) is globally stable to the fixed point $\delta_{2^n}^r$ (the limit cycle $\mathscr{C}$).

State feedback control design technique→**Pinning control** [5]

***Example*:** Consider BN (5) with

$$L = \delta_{64}[52, 52, 52, 52, 52, 52, 52, 52, 52, 52, 52, 52, 52, 52, 52, 52, 52, 52, 52, 52, 21, 22, 21, 22, 52, 52, 52, 52, 21, 22, 21, 22, 52, 52, 52, 52, 52, 52, 52, 52, 41, 41, 43, 43, 41, 41, 43, 43, 52, 52, 52, 52, 53, 54, 53, 54, 57, 57, 59, 59, 61, 61, 61, 61].$$

- *Step 1 [Calculate **E**(52)]:* Change the 21th, 22th, 41th, 43th, 53th, 54th, 57th, 59th, 61st columns to δ_{64}^{17}, δ_{64}^{18}, δ_{64}^{9}, δ_{64}^{11}, δ_{64}^{21}, δ_{64}^{22}, δ_{64}^{25}, δ_{64}^{27}, δ_{64}^{57}, respectively.

  Hence, L is changed to L'

$$L' = \delta_{64}[52, 52, 52, 52, 52, 52, 52, 52, 52, 52, 52, 52, 52, 52, 52, 52, 52, 52, 52, 52, 17, 18, 21, 22, 52, 52, 52, 52, 21, 22, 21, 22, 52, 52, 52, 52, 52, 52, 52, 52, 9, 41, 11, 43, 41, 41, 43, 43, 52, 52, 52, 52, 21, 22, 53, 54, 25, 57, 27, 59, 57, 61, 61, 61].$$

State feedback control design technique→**Pinning control** [5]

- *Step 2:* The 21th column of L is δ_{64}^{21}, that is $\delta_2^1 \ltimes \delta_2^2 \ltimes \delta_2^1 \ltimes \delta_2^2 \ltimes \delta_2^1 \ltimes \delta_2^1$. The 21th column of L' is δ_{64}^{17}, that is $\delta_2^1 \ltimes \delta_2^2 \ltimes \delta_2^1 \ltimes \delta_2^1 \ltimes \delta_2^1 \ltimes \delta_2^1$. Hence, we can perturb function f_4. By doing this, it can be verified that, to achieve the problem of global stabilization to δ_{64}^{52}, we need to perturb f_1, f_4.
- *Step 3:* It can be proved that the structure matrix of f_1 is changed from

$$M_1 = \delta_2[2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, \\ 1, 1, 1, 1, 2, 2, 2, 2, 1, 1, 1, 1, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, \\ 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2]$$

$$M_1' = \delta_2[2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, \\ 1, 1, 1, 1, 2, 2, 2, 2, 1, 1, 1, 1, 2, 2, 2, 2, 2, 2, 2, 2, 1, 2, \\ 1, 2, 2, 2, 2, 2, 2, 2, 2, 2, 1, 1, 2, 2, 1, 2, 1, 2, 2, 2, 2, 2].$$

State feedback control design technique→**Pinning control** [5]

The structure matrix of f_4 is changed from

$$M_4 = \delta_2[1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 2$$
$$2, 2, 2, 1, 1, 1, 1, 2, 2, 2, 2, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1$$
$$1, 1, 1, 1, 1, 1, 1, 1, 1, 2, 2, 2, 2, 1, 1, 1, 1, 2, 2, 2, 2]$$

$$M_2' = \delta_2[1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1$$
$$1, 2, 2, 1, 1, 1, 1, 2, 2, 2, 2, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1$$
$$1, 1, 1, 1, 1, 1, 1, 1, 1, 2, 2, 2, 2, 1, 1, 1, 1, 1, 2, 2, 2].$$

- *Step 4:* Assume that f_1 is changed to

$$F_1 = u_1(x_1, \ldots, x_6) \oplus_1 f_1(x_1, \ldots, x_6)$$
$$= M_{\oplus_1} \bar{M}_1 (I_{64} \otimes M_1) \Phi_6 x_1 \ldots x_6 = M_1' x_1, \ldots, x_6$$

where $M_{\oplus_1}$, $\bar{M}_1$ are the structure matrices of logical functions $\oplus_1$, $u_1(x_1, \ldots, x_6)$, respectively.

State feedback control design technique→**Pinning control**[5]

Similarly, assume that f_4 is changed to

$$\begin{aligned} F_4 &= u_2(x_1, \ldots, x_6) \oplus_2 f_4(x_1, \ldots, x_6) \\ &= M_{\oplus_2} \bar{M}_2 (I_{64} \otimes M_4) \Phi_6 x_1 \ldots x_6 = M_2' x_1, \ldots, x_6 \end{aligned}$$

where $M_{\oplus_2}$, $\bar{M}_2$ are the structure matrices of logical functions $\oplus_2$, $u_2(x_1, \ldots, x_6)$, respectively.

- *Step 5:* Solving the equations

$$\begin{aligned} M_{\oplus_1} \bar{M}_1 (I_{64} \otimes M_1) \Phi_6 &= M_1' \\ M_{\oplus_2} \bar{M}_2 (I_{64} \otimes M_4) \Phi_6 &= M_2' \end{aligned}$$

yields $M_{\oplus_1} = \delta_2[1, 2, 2, 2]$, $M_{\oplus_2} = \delta_2[1, 2, 1, 1]$. Hence

$$\begin{aligned} F_1(u_1, x_1, \ldots, x_6) &= u_1 \wedge f_1(x_1, \ldots, x_6) \\ F_4(u_2, x_1, \ldots, x_6) &= u_4 \rightarrow f_4(x_1, \ldots, x_6). \end{aligned}$$

State feedback control design technique→**Pinning control**[5]

$$\bar{M}_1 = \delta_{64}[1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,\\ 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 2, 1, 2,\\ 1, 1, 1, 1, 1, 1, 1, 1, 1, 2, 2, 1, 1, 2, 1, 2, 1, 1, 1, 1, 1]$$

$$\bar{M}_2 = \delta_{64}[2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2,\\ 2, 1, 1, 2, 2, 2, 2, 1, 1, 1, 1, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2,\\ 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 1, 1, 1, 1, 2, 2, 2, 2, 2, 1, 1, 1].$$

$$u_1 = x_1 \vee [\neg x_1 \wedge [x_2 \wedge [x_3 \vee [\neg x_3 \wedge \neg(x_4 \wedge x_6)]]\\ \vee [\neg x_2 \wedge [x_3 \wedge [x_4 \vee (\neg x_4 \wedge \neg x_5)]\\ \vee [\neg x_3 \wedge \neg(x_4 \wedge x_6)]]]]]$$

$$u_2 = [x_1 \wedge [\neg x_2 \wedge [[x_3 \wedge \neg(x_4 \vee x_5)] \vee (\neg x_3 \wedge \neg x_4)]]]\\ \vee [\neg x_1 \wedge [\neg x_2 \wedge [(x_3 \wedge \neg x_4)\\ \vee [\neg x_3 \wedge (\neg x_4 \wedge \neg(x_5 \vee x_6))]]]].$$

[6] J. Zhong, et al, Global robust stability and stabilization of Boolean network with disturbances, *Automatica*, 84:142-148,2017.

State feedback control design technique→**Pinning control**[6]

Here, we consider the following BN with disturbances:

$$\begin{cases} x_1(t+1) = f_1(\xi_1(t), \dots, \xi_p(t), x_1(t), \dots, x_n(t)), \\ \quad \dots \\ x_n(t+1) = f_n(\xi_1(t), \dots, \xi_p(t), x_1(t), \dots, x_n(t)), \end{cases} \tag{1}$$

$$x(t+1) = L\xi(t)x(t),$$

$$\begin{cases} x_i := u_i(x_1, \dots, x_n) \oplus_i f_i(\xi_1, \dots, \xi_p, x_1, \dots, x_n), \\ \quad = \hat{f}_i(\xi_1, \dots, \xi_p, x_1, \dots, x_n), i = 1, \dots, k, \\ x_j = f_j(\xi_1, \dots, \xi_p, x_1, \dots, x_n), j = k+1, \dots, n, \end{cases} \tag{3}$$

$$M_{\oplus_i}\mathcal{K}_i(I_{2^n} \otimes F_i)W_{[2^p, 2^n]}(I_{2^p} \otimes \Phi_n) = \hat{F}_i, i = 1, \dots, k, \tag{4}$$

$M_{\oplus_1}, \dots, M_{\oplus_k} \in \mathscr{L}_{2\times 4}$ are structure matrices of logical functions $\oplus_1, \dots, \oplus_k$ to be determined and $\mathcal{K}_1, \dots, \mathcal{K}_k \in \mathscr{L}_{2\times 2^n}$ are the gain matrices of $u_1, \dots, u_k$ also to be determined.

[6] J. Zhong, et al, Global robust stability and stabilization of Boolean network with disturbances, *Automatica*, 84:142-148,2017.

State feedback control design technique→**Event-triggered control**[7]

- Event-triggered control (ETC) consists of two parts: (1) a **state feedback mechanism** to **determine the control inputs** and (2) a set of **states** to **decide when the control inputs should be considered**.
- Compared with traditional state feedback control, the designed ETC approach not only **shortens the transient period** of logical networks but also **decreases the number of controller executions**.
- The global stabilization problem of KVLCNs is introduced in the sequel via the **time-optimal** event-triggered controller and **switching-cost-optimal** event-triggered controller.

---

[7] S. Zhu, Y. Liu, Y. Lou and J. Cao, Stabilization of logical control networks: An event-triggered control approach, *Science China. Information Sciences*, 2020, 63: 112203.

State feedback control design technique→**Event-triggered control**[7]

➤ The KVLCN under ETC, presented as follows, consists of an inherent non-control KVLN (1a), an alternative KVLCN (1b), and a triggering event set $\Lambda \subseteq \mathscr{D}_k^n$ standing for certain individual states where the control inputs are triggered:

$$\begin{cases} x_1(t+1) = f_1(x_1(t), \ldots, x_n(t)), \\ x_2(t+1) = f_2(x_1(t), \ldots, x_n(t)), \\ \quad \vdots \\ x_n(t+1) = f_n(x_1(t), \ldots, x_n(t)), \end{cases} \tag{1a}$$

$$\begin{cases} x_1(t+1) = f'_1(x_1(t), \ldots, x_n(t), u_1(t), \ldots, u_m(t)), \\ x_2(t+1) = f'_2(x_1(t), \ldots, x_n(t), u_1(t), \ldots, u_m(t)), \\ \quad \vdots \\ x_n(t+1) = f'_n(x_1(t), \ldots, x_n(t), u_1(t), \ldots, u_m(t)), \end{cases} \tag{1b}$$

$$x(t+1) = Lx(t), \tag{3a}$$

$$x(t+1) = L'u(t)x(t), \tag{3b}$$

State feedback control design technique→**Event-triggered control**[7]

- The ETC mechanism is essentially an intermittent control strategy.

$$x(t+1) = \begin{cases} Lx(t), & x(t) \in \Delta_N \backslash \Gamma(\Lambda), \\ L'u(t)x(t), & x(t) \in \Gamma(\Lambda). \end{cases} \quad (4)$$

$$\Gamma(\Lambda) := \{\Gamma_n(\boldsymbol{x}) : \boldsymbol{x} \in \Lambda\}, x(t) = \Gamma_n(x_1(t), x_2(t), \ldots, x_n(t))$$

$$x(t+1) = [L, L'] \widetilde{u}(t)x(t) := \widetilde{L}\widetilde{u}(t)x(t), \ \widetilde{u}(t) \in \Delta_{M+1} \quad (5)$$

(1) If $x(t) \in \Delta_N \backslash \Gamma(\Lambda)$, then $\widetilde{u}(t) := \delta_{M+1}^1$.

(2) If $x(t) \in \Gamma(\Lambda)$, then one obtains that $\widetilde{u}(t) := [0, u(t)^{\mathrm{T}}]^{\mathrm{T}}$.

- The stabilization of system (5) is equivalent to the event-triggered stabilization of system (4).

State feedback control design technique→**Event-triggered control**[7]

- $u(t) = Gx(t) = \delta_M[\beta_1, \beta_2, \ldots, \beta_N]x(t),$ (6)

$$\tilde{u}(t) = \tilde{G}x(t) = \delta_{M+1}[\gamma_1, \gamma_2, \ldots, \gamma_N]\, x(t)$$

$$\gamma_j = \begin{cases} 1, & \delta_N^j \in \Delta_N \backslash \Gamma(\Lambda), \\ \beta_j + 1, & \delta_N^j \in \Gamma(\Lambda). \end{cases} \quad (7)$$

- The ***objective*** of this paper is to design the possible state feedback matrix $\tilde{G} \in \mathcal{L}_{(M+1)\times N}$ such that KVLCN (5) is globally stabilizable under the time-optimal stabilizer and the switching-cost-optimal stabilizer.

- The time-optimal stabilizer aims to **minimize the transient period** and the switching-cost-optimal stabilizer aims to **minimize the cardinal number of triggering event set** $|\Gamma(\Lambda)|$.

State feedback control design technique→**Event-triggered control**[7]

♣ **Design of time-optimal event-triggered stabilizer**

➤ Define

$$\mathscr{R}_v(r) = \Big\{ \delta_N^j \in \Delta_N \;:\; \text{A control sequence } \widetilde{\boldsymbol{u}}(0), \widetilde{\boldsymbol{u}}(1), \ldots, \widetilde{\boldsymbol{u}}(v-1) \in \Delta_{M+1} \text{ exists such that } x(v; \delta_N^j, \widetilde{\boldsymbol{u}}(0), \widetilde{\boldsymbol{u}}(1), \ldots, \widetilde{\boldsymbol{u}}(v-1)) = \delta_N^r \Big\}. \quad (8)$$

Theorem 1. For a given state $\delta_N^r \in \Delta_N$, system (4) can be globally δ_N^r-stabilized by an event-triggered controller if and only if both of the following conditions are satisfied:

(1) $\delta_N^r \in \mathscr{R}_1(r)$;

(2) An integer $l \in [1, N-1]$ exists such that $\mathscr{R}_l(r) = \Delta_N$.

➤ Without any confusion, the **minimal integer** satisfying condition (2) is denoted by ***l****.

State feedback control design technique→**Event-triggered control**[7]

♣ Design of time-optimal event-triggered stabilizer

- Split

$$\Delta_N = (\mathscr{R}_{l^*}(r) \setminus \mathscr{R}_{l^*-1}(r)) \cup \cdots \cup (\mathscr{R}_2(r) \setminus \mathscr{R}_1(r)) \cup (\mathscr{R}_1(r) \setminus \mathscr{R}_0(r)) \cup \mathscr{R}_0(r). \quad (9)$$

- Define $\alpha_i = \widetilde{L}\delta_{MN}^i$ for $i \in [1, MN]$.

- The 'state feedback matrix' $\widetilde{G} = \delta_{M+1}[\gamma_1, \gamma_2, \ldots, \gamma_N]$ can be given

(1) If $\alpha_r = r$, let $\gamma_r = 1$. Otherwise, namely, $\alpha_r \neq r$, let γ_r be a solution of $\alpha_{(\gamma_r-1)N+r} = r$.

(2) For $i \in [1, N]\backslash\{r\}$, if $\delta_N^{\alpha_i} \in \mathscr{R}_{l_i-1}(r)$, let $\gamma_i = 1$. Otherwise, let γ_i be a solution of $\delta_N^{\alpha_{(\gamma_i-1)N+i}} \in \mathscr{R}_{l_i-1}(r)$.

$$G = \delta_M[\beta_1, \beta_2, \ldots, \beta_N], \text{ where } \beta_i = \gamma_i - 1 \text{ if } \gamma_i \neq 1$$

and β_i can be arbitrarily selected in $[1, M]$ for $\gamma_i = 1$.

State feedback control design technique→**Event-triggered control**[7]

♣ Design of time-optimal event-triggered stabilizer

Example: Let $L = \delta_4[1,1,3,4]$ and $L' = \delta_4[1,3,4,1,1,3,4,1]$.

- Let $r = 1$.

$\mathscr{R}_1(1) = \{\delta_4^1, \delta_4^2, \delta_4^4\}$

$\mathscr{R}_2(1) = \Delta_4$.

➡ system can be globally stabilizable to δ_4^1 under ETC.

➡ $\gamma_1 = 1$, $\gamma_2 = 1$, $\gamma_3 = 2, 3$, and $\gamma_4 = 2, 3$.

➡ $\Gamma(\Lambda) = \{\delta_4^1, \delta_4^2\}$

➡ β_1, β_2, β_3, and β_4 can be arbitrarily selected from $\{1, 2\}$.

State feedback control design technique→**Event-triggered control**[7]

♣ Design of switching-cost-optimal event-triggered stabilizer

- In the following, based on the knowledge of **graph theory**, we present a universal and unified approach to minimize the triggering event set.
- First of all, a **labeled digraph** $\mathscr{G} = (V, A)$ is derived for equivalent graphical description of the dynamic of KVLCN (5).

KVLN (3a) ➡ $\mathscr{G}_0 = (V, A_0), A_0$ is a real line arc set ⟺ $[L]_{ji} = 1$.

$L' = [L'_1, L'_2, \ldots, L'_M]$

KVLCN (3b) ➡ $\mathscr{G} = \bigcup_{\mu=0}^{M} \mathscr{G}_\mu = \left(V, \bigcup_{\mu=0}^{M} A_\mu\right), A_\mu$ is a dashed line arc set

⇕

$[L'_\mu]_{ji} = 1, \mu \in [1, M]$

State feedback control design technique→**Event-triggered control**[7]

♣ Design of switching-cost-optimal event-triggered stabilizer

➤ The **pretreatment**:

(1) Delete all self loops.

(2) For all ordered pairs $(i, j) \in [1, N] \times [1, N]$ and $i \neq j$, we retain the arc with minimal weight joining i to j and delete the others. If two such arcs exist, we select the arbitrary one.

(3) Assign each dashed line arc joining i to j by a control set $u_{(i,j)} := \{\mu : [L_\mu]_{ji} = 1,\ \mu \in [1, M]\}$.

State feedback control design technique→**Event-triggered control**[7]

♣ Design of switching-cost-optimal event-triggered stabilizer

- As mentioned in [8], the stabilization problem of KVLCN can be **equivalently** described by the existence of spanning in-tree with the designated vertex ***r***, which is called the root of tree.
- An approach to find the switching-cost-optimal event-triggered stabilizer is exactly to **find a spanning in-tree at root *r* with the minimal number of dashed line arcs** in labeled digraph $\mathscr{G}$.
- To this end, **weights** ***N*** and ***1*** are respectively assigned to each dashed line arc and real line.

$$\mathscr{G} := (V, A, W), \text{ where } W \text{ is a set of weight } w(i,j) \text{ for all } (i,j) \in A.$$

---

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

State feedback control design technique→**Event-triggered control**[7]

♣ Design of switching-cost-optimal event-triggered stabilizer

- The spanning in-tree at root ***r*** with the **minimal sum of weight** is called the minimal spanning in-tree of labeled digraph $\mathscr{G}$.

Algorithm 1 Minimal spanning in-tree algorithm

Step 1: Initialize $i := 0$, $V_0 := V$, $E_0 := A$ and $W_0 := W$. Designate vertex r as the root.

Step 2: Calculate $J_1 = \{(v, \theta(v)) : v \in V_0 \setminus \{r\}\}$, where an order pair $(v, \theta(v))$ is the minimal weight arc among all $(v, j) \in E_0$.

Step 3: Check whether directed cycles exist in (V_i, J_{i+1}). If so, then proceed to Step 4. Otherwise, proceed to Step 7.

Step 4: Contract every cycle $\mathscr{C}$ into one new vertex to obtain a new diagraph $(V_{i+1}, E_{i+1}, W_{i+1})$; the weight set W_{i+1} is updated from W_i as follows; then $i := i + 1$ and go to Step 5.

- If (u, v) is an arc joining cycle $\mathscr{C}$, its weight is kept unchanged.
- If (u, v) is an arc away cycle $\mathscr{C}$, its weight is reassigned as $w(u, v) - w(u, \theta(u))$.
- The weights of the other arcs are kept unchanged.

Step 5: Perform pretreatment for the novel labeled digraph (V_i, E_i, W_i).

Step 6: Calculate $J_{i+1} = \{(v, \theta(v)) : v \in V_i \setminus \{r\}\}$, where an order pair $(v, \theta(v))$ is the minimal weight arc among all $(v, j) \in E_i$. Then, return to Step 3.

Step 7: Expand the contracted cycles formed during the preceding phase in reverse order of their contraction and remove one arc from each cycle to form a spanning in-tree.

State feedback control design technique→**Event-triggered control**[7]

♣ Design of switching-cost-optimal event-triggered stabilizer

- The returned minimal spanning in-tree in Algorithm ***1*** is denoted by $\mathscr{G}^0 = (V, A^0, W^0)$, where $A^0 \subseteq A$ and $W_0 \subseteq W$.
- $\lfloor D \rfloor$ consists of the starting vertex of each arc in D, $D \subseteq A$

Algorithm 2 Corresponding event-triggered controller design from minimal spanning in-tree

---

Step 1: Construct the triggering event set $\Gamma(\Lambda)$. If $[L]_{rr} = 1$, then $\Gamma(\Lambda) = \{\delta_N^i : i \in \lfloor A^0 \backslash A_0 \rfloor\}$. Otherwise, $\Gamma(\Lambda) = \{\delta_N^i : i \in \lfloor A^0 \backslash A_0 \rfloor \cup \{r\}\}$.

Step 2: Determine the state feedback matrix G. Let β_r be randomly selected in Δ_M if $r \notin \Gamma(\Lambda)$; else, $\beta_r = u_{(r,r)}$. For every $j \in [1, N] \backslash \{r\}$, if $j \in \Gamma(\Lambda)$, a unique integer $t_j \in [1, N]$ satisfies $(j, t_j) \in A^0$. Let β_j be an arbitrary integer in $u_{(j,t_j)}$. Otherwise, let β_j be an arbitrary integer in $[1, M]$. The feasible state feedback matrix can be designed as $G = \delta_M[\beta_1, \beta_2, \ldots, \beta_N]$.

State feedback control design technique→**Event-triggered control**[7]

♣ **Design of switching-cost-optimal event-triggered stabilizer**

Example:

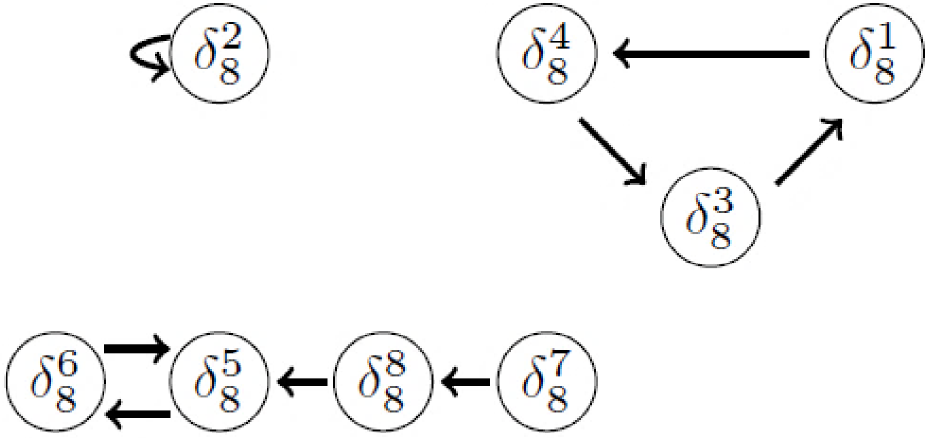


Figure 1 State transition graph of KVLN with respect to transition matrix $L = \delta_8[4, 2, 1, 3, 6, 5, 8, 5]$.

State feedback control design technique→**Event-triggered control**[7]

♣ **Design of switching-cost-optimal event-triggered stabilizer**

➢ The **pretreatment**:

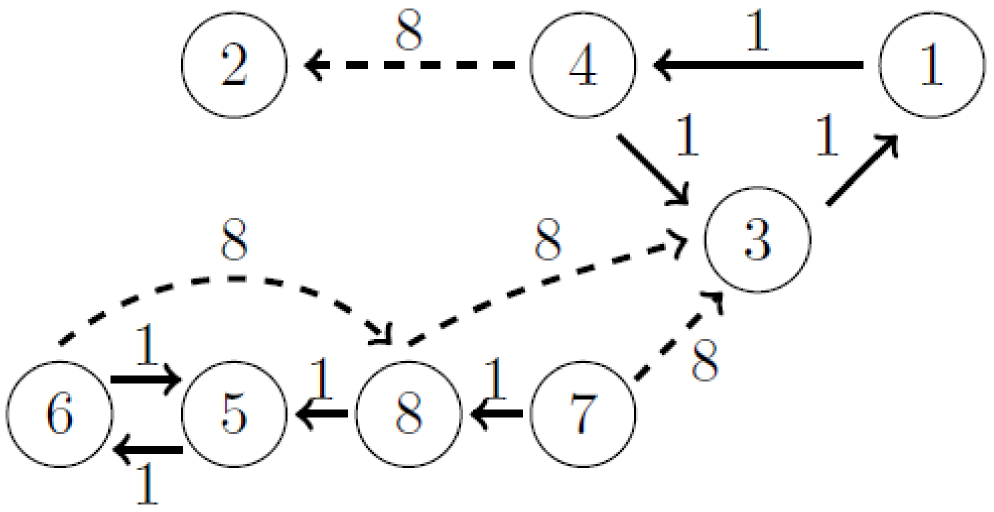


Figure 2 Labeled digraph after pretreatment (V_0, E_0, W_0).

State feedback control design technique→**Event-triggered control**[7]

♣ Design of switching-cost-optimal event-triggered stabilizer

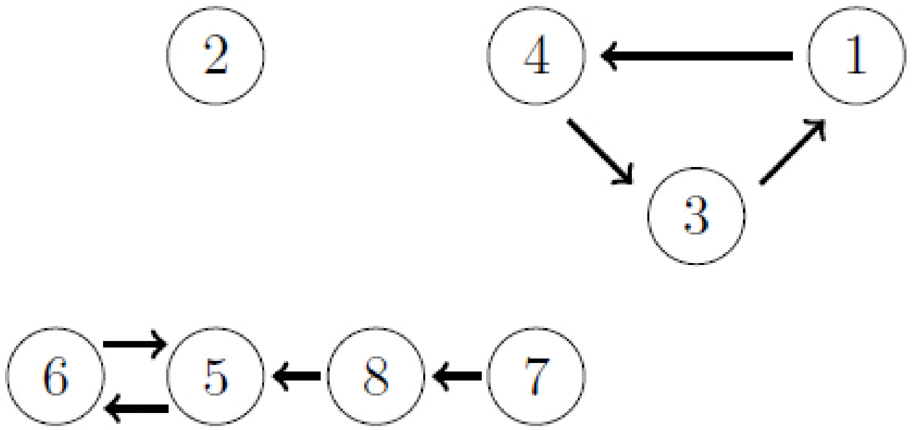


Figure 3 Calculate set $J_1 = \{(v, \theta(v)) \mid v \in [1, 8]\}$ by Step 2 in Algorithm 1. That is, $\theta(1) = 4$, $\theta(3) = 1$, $\theta(4) = 3$, $\theta(5) = 6$, $\theta(6) = 5$, $\theta(7) = 8$ and $\theta(8) = 5$.

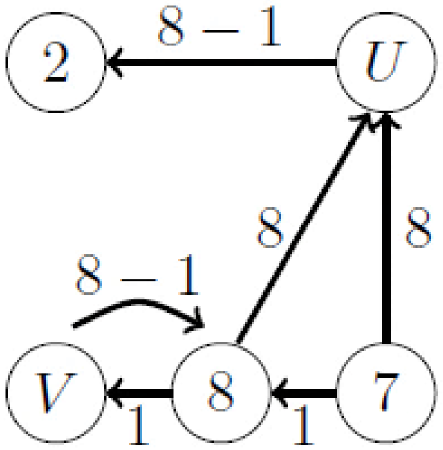


Figure 4 New constructed weighted directed graph (V_1, E_1, W_1). Based on Algorithm 1, $w(U, 2) = 8 - 1$ and $w(V, 8) = 8 - 1$. The weights of the other arcs remain unchanged.

State feedback control design technique→**Event-triggered control**[7]

♣ Design of switching-cost-optimal event-triggered stabilizer

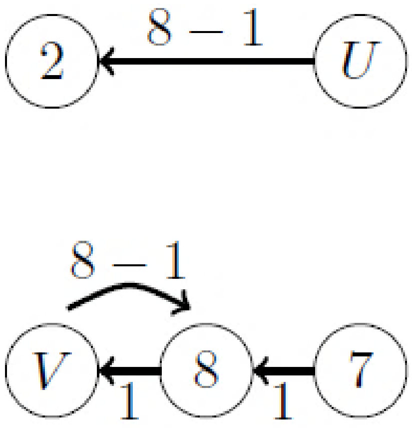


Figure 5 Find the set J_2 in Figure 4, where $J_2 = \{(V, 8), (8, V), (7, 8), (U, 2)\}$.

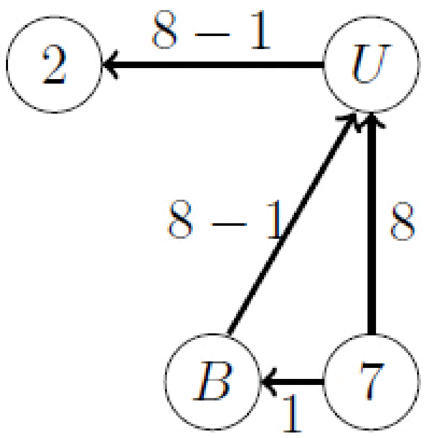
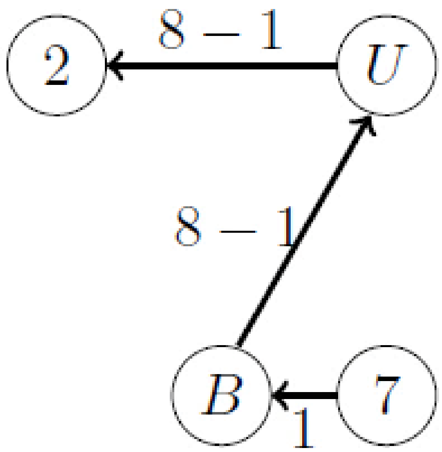


Figure 6 The vertices V and 8 are contracted into a novel vertex B. Let $w(U, B) = 8 - 1$ and the weights of the other arcs be unchanged.

State feedback control design technique→ **Event-triggered control**[7]

♣ Design of switching-cost-optimal event-triggered stabilizer



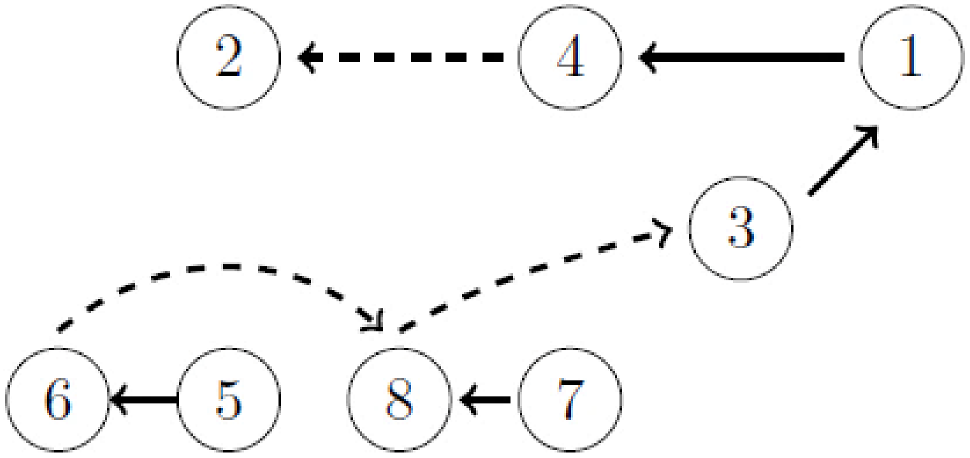


Figure 7 The set $J_3 = \{(2,U),(U,B),(B,7)\}$. **Figure 8** Minimal spanning in-tree $\mathscr{G}^0$ of Example 2.

➤ The corresponding **triggering event set** is designed as $\Gamma(\Lambda) = \{\delta_8^4, \delta_8^6, \delta_8^8\}$ and **the possible state feedback matrices** are $G = \delta_8[*,*,*,*,*,1,*,*]$, $*$ is 1 or 2. According to Figure 8, the number of **control executions** is equal to 3.

State feedback control → **Sampled-data control**[9]

- A Boolean dynamic system with n nodes,

$$\begin{cases} x_1(t+1) = f_1(x_1(t), \ldots, x_n(t), u_1(t), ..., u_m(t)), \\ \quad \cdots \\ x_n(t+1) = f_n(x_1(t), \ldots, x_n(t), u_1(t), ..., u_m(t)), \end{cases} \tag{2}$$

where f_i, $i = 1, 2, \ldots, n$ are logical functions, and u_j are control inputs.

- The feedback law to be determined for system (2) is in the form of

$$u(t) = Ex(t_l), \quad t_l \leq t < t_{l+1}, \tag{8}$$

where $t_l = l\tau \geq 0$ for $l = 0, 1, \ldots$ are sampling instants and $t_{l+1} - t_l = \tau$ denotes the constant sampling period.

[9] Liu Y, Cao J, Sun L, Lu, J. Sampled-data state feedback stabilization of Boolean control networks. *Neural Computation*, 2016, 28(4): 778-799.

State feedback control → **Sampled-data control**[9]

Theorem: System (2) is globally stabilized to $\delta_{2^n}^r$ by a **sampled-data state feedback control** (SDSFC) in the form of (8), if and only if there exists a $\boldsymbol{k} > 0$, such that

$$\begin{cases} \left(\left(LEW_{[2^n]}\right)^{\tau} \Phi_n^{\tau}\right)^k = \delta_{2^n}[\underbrace{r\ r\ \cdots\ r}_{2^n}], \\ \left(LEW_{[2^n]}\Phi_n\right)_{rr} = 1. \end{cases}$$

State feedback control → **Sampled-data control[9]**

➤ Main results (**Piecewise stabilization** of BCNs)

From $L = [{}^1L \; {}^2L \; \cdots \; {}^{2^m}L]$, we denote

$$\hat{L}^l = \bigvee_{j=1}^{2^m} ({}^jL)^l, \; l \geq 1. \tag{10}$$

We now consider the Piecewise constant control (PCC) for system (2) as follows,

$$u(t) = u(t_l) \in \Delta_{2^m}, \; t_l \leq t < t_{l+1}, \tag{11}$$

where $t_l = l\tau \geq 0$ for some $\tau > 0$ and $l \geq 0$.

State feedback control → **Sampled-data control**[9]

- Main results (**Piecewise stabilization** of BCNs)

Theorem: There exists a sequence of PCCs such that BCN (2) can be globally stabilized to $\delta_{2^n}^r$ there exists a minimum $1 \leq \boldsymbol{N} \leq 2^n$, such that

$$Row_r\left(\bigvee_{j=1}^{N}(\hat{L}^\tau)^j\right) > 0,\ \ and\ \ \left(\hat{L}^1\right)_{rr} = 1.$$

Theorem: A BCN (2) can be globally stabilized to $\delta_{2^n}^r$ by a sequence of PCCs ***iff*** it is stabilizable by means of a SDSFC.

State feedback control → **Sampled-data control[10]**

- A BCN under ASDC can be described as follows:

$$X(t+1) = f(X(t), U(t)), \quad (1)$$

$$U(t) = e(X(t_k)),\ t_k \le t < t_{k+1}, \quad (2)$$

t_k for $k = 0, 1, \ldots$ are sampling instants.

The sampling period is defined as follows.

Let $h_k \triangleq t_{k+1} - t_k$, where $h_k \in Z_h \triangleq \{i_1, \cdots, i_l\}$, $i_1 < i_2 < \cdots < i_l$ and i_j, $j = 1, \cdots, l$, are positive integers. Then h_k can take one value from these l values.

Using STP, a BCN under ASDC can be converted into a switched BN.

-
$$\begin{aligned} x(t_{k+1}) &= (LW_{[2^n,2^m]})^{h_k} x(t_k)\Phi_m^{h_k-1}u(t_k), \\ &\triangleq \tilde{L}^{h_k} x(t_k)\Phi_m^{h_k-1}u(t_k), \\ &= A_{\sigma(t_k)} x(t_k) B_{\sigma(t_k)} u(t_k), \\ &\triangleq F_{\sigma(t_k)} x(t_k), \end{aligned} \quad (3)$$

where $k = 0,1,\ldots$, $F_{\sigma(t_k)} = A_{\sigma(t_k)}(I_{2^n} \otimes B_{\sigma(t_k)} K)\Phi_n$ and $\sigma(t_k) \in Z_\sigma = \{1,2,\ldots,l\}$.

[10] Lu J, Sun L, Liu Y, et al. Stabilization of Boolean control networks under aperiodic sampled-data control. *SIAM Journal on Control and Optimization*, 2018, 56(6): 4385-4404.

State feedback control → **Sampled-data control**[10]

- Main results

It is worthwhile to note that switches may not occur at every sampling instant. The following example is given to explain this point.

Example: Assume that the sampling period h_k takes three values $i_1 = 2, i_2 = 4,\ i_3 = 6$; when system (3) has three subsystems and $\sigma(t_k) \in Z_\sigma = \{1,2,3\}$.

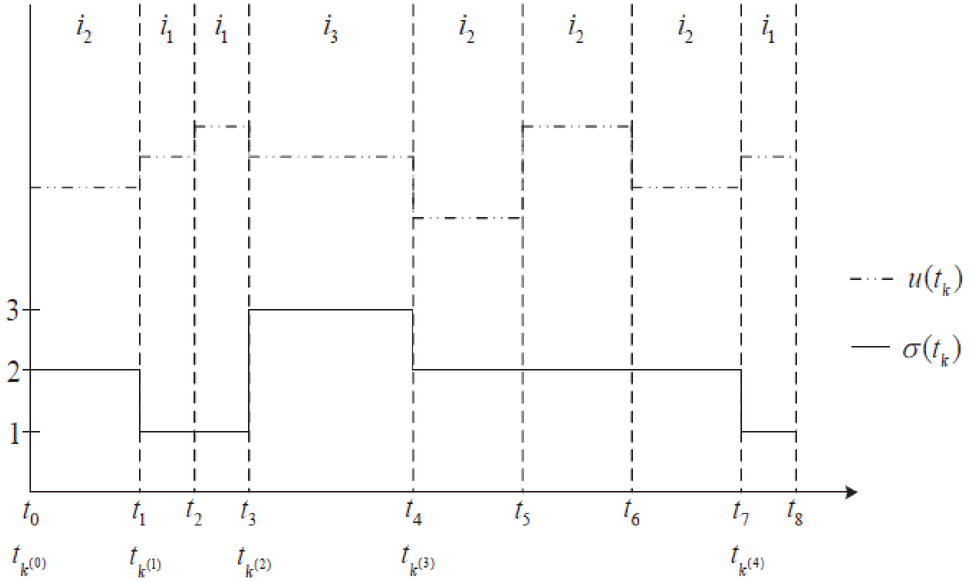


Figure 3

In figure 3, $t_k, k = 1, \dots, 8$ are the sampling instants. But the switches only occur at $t_{k(j)}, j = 1, \dots, 4$, where $t_{k(1)} = t_1, t_{k(2)} = t_3, t_{k(3)} = t_4$, and $t_{k(4)} = t_7$, which means that switches may not occur at every sampling instant.

State feedback control → **Sampled-data control**[10]

Main results

For any $t_k > t_0 = 0$, a switching sequence $t_0 = t_{k^{(0)}} < t_{k^{(1)}} < \cdots < t_{k^{(i)}} < t_k$ during interval $[t_0, t_k)$ is assumed. When $t \in [t_{k^{(j)}}, t_{k^{(j+1)}}), j = 1,2, \dots, i$, the $\sigma(t_{k^{(j)}})$th subsystem is activated; that is, switching sequence corresponding to the switching signal $\sigma(t_k)$ is given as follows:

$$\{(\sigma(t_{k^{(0)}}), t_{k^{(0)}}), \dots, (\sigma(t_{k^{(i)}}), t_{k^{(i)}}) | \sigma(t_{k^{(j)}}) \in Z_\sigma, j = 0,1, \dots, i, i \geq 1\},$$

where $t_{k^{(j)}} \in \{t_1, \dots, t_{k-1}\}, j = 1, \dots, i$.

For the switched BN, we consider two cases:

(i) switched BN with all stable subsystems;

(ii) switched BN containing both stable subsystems and unstable subsystems.

For these two cases, the techniques of switching-based Lyapunov function and the average dwell time method are used to derive sufficient conditions for global stability of BCNs under ASDC, respectively.

State feedback control → **Sampled-data control**[10]

Definition 1: $\{\beta_j | j \in \{1,2,\ldots,l\}\}$ is called a set of Lyapunov coefficients of system (3) if following conditions are satisfied

$$\beta_j^{\mathrm{T}} \delta_{2^n}^{2^n} = 0, \quad (4)$$

$$\beta_j^{\mathrm{T}} \delta_{2^n}^{r} > 0,\ r = 1,2,\cdots,2^n - 1, \quad (5)$$

$$\beta_j^{\mathrm{T}} (F_j - \lambda_j^{2i_j} I_{2^n}) \delta_{2^n}^{2^n} = 0, \quad (6)$$

$$\beta_j^{\mathrm{T}} (F_j - \lambda_j^{2i_j} I_{2^n}) \delta_{2^n}^{r} < 0,\ r = 1,2,\cdots,2^n - 1, \quad (7)$$

$$\beta_i \le \mu \beta_j,\ i,j \in Z_\sigma,\ \mu \ge 1, \quad (8)$$

Where $0 < \lambda_j < 1$, if the $j\text{−}th$ subsystem is stable; $\lambda_j \ge 1$, if the $j\text{−}th$ subsystem is unstable, and i_j is the sampling period.

State feedback control → **Sampled-data control**[10]

➢ **switched BN with all stable subsystems (i.e.,** all subsystems is globally stable at $X_e = (x_1^e, \dots, x_n^e)$, here, we assume $X_e = (0, \dots, 0)$**)**

Theorem 1: Given $\alpha_j \geq 0, \sum_{j=1}^{l} \alpha_j = 1$. If there exists a set of Lyapunov coefficients $\{\beta_j | j \in \{1,2,\dots,l\}\}$ as defined in Definition 1, such that $\tau_a > \frac{\ln \sqrt{\mu}}{\sum_{j=1}^{l} \alpha_j \ln \lambda_j^{-1}}$, then system (1) is globally stable under feedback (2) at X_e with a decay rate $\theta(\tau_a, \alpha_j) = \mu^{\frac{1}{2\tau_a}} \prod_{j=1}^{l} \lambda_j^{\alpha_j}$, where α_j are the activation frequencies of the sampling periods.

State feedback control → **Sampled-data control**[10]

➤ **switched BN containing both stable subsystems and unstable subsystems**(Here, suppose that the index set of stable subsystem is φ_s and the index set of unstable subsystem is φ_u, where $|\varphi_s| = r$ and $|\varphi_u| = l - r$. Denote $f_s = \sum_{j\in\varphi_s} \alpha_j$ and $f_u = \sum_{j\in\varphi_u} \alpha_j$)

Theorem 2: Given $\alpha_j \geq 0, \sum_{j=1}^{l} \alpha_j = 1$. If there exists a set of Lyapunov coefficients $\{\beta_j | j \in \{1,2,\dots,l\}\}$ as defined in Definition 1, such that $\prod_{j=1}^{l} \lambda_j^{\alpha_j} < 1, \tau_a > \frac{\ln\sqrt{\mu}}{\ln \lambda_s^{-1} + f_u \ln(\lambda_s \lambda_u^{-1})}, f_u < \frac{\ln \lambda_s^{-1}}{\ln(\lambda_u \lambda_s^{-1})}$, then system (1) is globally stable under feedback (2) at X_e with a decay rate $\hat{\theta}(\tau_a, f_u) = \mu^{\frac{1}{2\tau_a}} \lambda_s (\lambda_u \lambda_s^{-1})$.

State feedback control → **Sampled-data control**[10]

- Main results

We derive some conditions to guarantee not only the **global stability** of system (3) but also **the performance with an adequate level**.

Definition 2: $\{\beta_j | j \in \{1,2,\ldots,l\}\}$ is called **a set of Lyapunov coefficients for the cost function** $\boldsymbol{J = \sum_{k=0}^{\infty} \xi^T u(t_k) x(t_k)}$ of system (3) if following conditions are satisfied (4), (5), (6), (7) and

$$(\beta_j^{\mathrm{T}}(F_j - \lambda_j^{2i_j} I_{2^n}) + \xi^{\mathrm{T}} K \Phi_n)\delta_{2^n}^r < 0, \tag{9}$$

where $\boldsymbol{0 < \lambda_j < 1}$, if the j−th subsystem is **stable**; $\boldsymbol{\lambda_j \geq 1}$, if the j−th subsystem is **unstable**, and $\boldsymbol{i_j}$ **is the sampling period**.

State feedback control → **Sampled-data control**[10]

Main results

Theorem 3: If there exists a set of Lyapunov coefficients $\{\beta_j | j \in \{1,2,\ldots,l\}\}$ for the cost function $J = \sum_{k=0}^{\infty} \xi^T u(t_k) x(t_k)$ as definition 2, such that $\tau_a > \frac{\ln \sqrt{\mu}}{\sum_{j=1}^{l} \alpha_j \ln \lambda_j^{-1}}$ holds, then system (1) is globally stable under feedback (2) with decay rate $\theta(\tau_a, \alpha_j)$ and cost function satisfies $J \leq \frac{1-\lambda_0^2}{\left(1-\theta^2(\tau_a,\alpha_j)\right)} V_{\sigma(t_0)}(t_0)$. (所有子集稳定)

Theorem 4: : If there exists a set of Lyapunov coefficients $\{\beta_j | j \in \{1,2,\ldots,l\}\}$ for the cost function $J = \sum_{k=0}^{\infty} \xi^T u(t_k) x(t_k)$ as definition 2, such that $\prod_{j=1}^{l} \lambda_j^{\alpha_j} < 1, \tau_a > \frac{\ln \sqrt{\mu}}{\ln \lambda_s^{-1} + f_u \ln(\lambda_s \lambda_u^{-1})}$ hold, then system (1) is globally stable under feedback (2) with decay rate $\hat{\theta}(\tau_a, \alpha_j)$ and cost function satisfies $J \leq \frac{1-\lambda_0^2}{\left(1-\hat{\theta}^2(\tau_a,\alpha_j)\right)} V_{\sigma(t_0)}(t_0)$. (部分子集稳定)

State feedback control → **Sampled-data control**[10]

Main results

考虑系统：$x(t_k) = \bar{L}^{j_1}\bar{L}^{j_2} \cdots \bar{L}^{j_k} x(0)u(0) \cdots u(t_k - 1)$，其中$\bar{L} = LW_{[2^n,2^m]}$，$t_k - t_{k-1} = j_1, \dots, t_1 - t_0 = j_k$。将$\bar{L}^{j_c}$划分为$2^n$等分。

构造集合:

$$R_{t_k}\left(\delta_{2^n}^{2^n}\right) = \{\delta_{2^n}^{i}: \text{there exists } q_i \in \left\{(p_i - 1)\frac{1-2^{mj_1}}{1-2^m} + 1 \middle| p_i = 1,2,\dots,2^m\right\},$$

$$\text{such that } Col_{q_i}\left(\bar{L}_i^{j_1}\right) = \delta_{2^n}^{2^n}\}\backslash\{\delta_{2^n}^{2^n}\},$$

$$\vdots$$

$$R_{t_1}\left(\delta_{2^n}^{2^n}\right) = \{\delta_{2^n}^{i}: \text{there exists } q_i \in \left\{(p_i - 1)\frac{1-2^{mj_k}}{1-2^m} + 1 \middle| p_i = 1,2,\dots,2^m\right\},$$

$$\text{such that } Col_{q_i}\left(\bar{L}_i^{j_k}\right) \in R_{t_2}(\delta_{2^n}^{2^n})\}\backslash[\cup_{2\le l\le k} R_{t_l}(\delta_{2^n}^{2^n}) \cup \{\delta_{2^n}^{2^n}\}].$$

State feedback control → **Sampled-data control**[10]

Main results

Algorithm 1

Step 1. Solve $\delta_{2^n}^{2^n} = LK\delta_{2^n}^{2^n}\delta_{2^n}^{2^n}$ to get p_{2^n}. If there is no solution, then K does not exist.

Step 2. For any initial state $x(0) = \delta_{2^n}^{i}, i = 1,2,\dots,2^n - 1$, if $\delta_{2^n}^{i} \in R_{t_c}\left(\delta_{2^n}^{2^n}\right), c \in \{1,2,\dots,k\}$, then there exists $q_i \in \left\{(p_i - 1)\frac{1-2^{m j_{k+1-c}}}{1-2^m} + 1 \middle| p_i = 1,2,\dots,2^m\right\}$ such that $Col_{q_i}(\bar{L}_i^{j_{k+1-c}}) \in R_{t_{c+1}}\left(\delta_{2^n}^{2^n}\right)$; get p_i. If $\delta_{2^n}^{i} \notin \cup_{1\le l\le k} R_{t_l}(\delta_{2^n}^{2^n})$, then K does not exist.

Step 3. The feedback matrix $K = \delta_{2^m}[p_1, p_2, \dots, p_{2^n}]$ can be obtained.

State feedback control → **Sampled-data control**[11]

Main results

$$x(t+1)=Lu(t)x(t) \qquad (4)$$

Definition 1. Input $u(t) \in \Delta_{2^m}^{\infty}$, $t \in \mathbb{N}$, is said to be $\Lambda-$nonuniform-sampled, if there is a sequence of integers t_i, which are called sampling points, satisfying $t_0 = 0$, $t_{i+1} - t_i \in \Lambda$ and $u(t) = u(t_i)$, $t \in \{t_i, t_i + 1, \ldots, t_{i+1} - 1\}$, $i \in \mathbb{N}$, where Λ is a subset of positive integers.

Definition 2. Boolean control network (4) is stabilizable to $\delta_{2^n}^{\alpha}$ under $\Lambda-$nonuniform-sampled inputs, if for any given initial state x_0, there exist a $\Lambda-$nonuniform-sampled input $u(t)$ and an integer T, such that $\delta_{2^n}^{\alpha} = x(x_0, u(t))$, $t \geq T$, where $\Lambda = \{\tau_a, \tau_b\}$ and $x(x_0, u(t))$ represents the state that (4) reaches at moment t under initial state x_0 and input $u(t)$.

---

[11] Yu Y, Feng J, Wang B, et al. Sampled-data controllability and stabilizability of Boolean control networks: Nonuniform sampling. *Journal of the Franklin Institute*, 2018, 355(12): 5324-5335.

State feedback control → **Sampled-data control**[11]

Main results

Under $\Lambda-$nonuniform-sampled inputs, to investigate the state transition of Boolean control network (4), we construct the following network, which can demonstrate states of (4) at sampling points t_k, $k \in \mathbb{N}$,

$$x(t_{k+1}) = L_S v(t_k) u(t_k) x(t_k), \tag{6}$$

where $v(\cdot) \in \Delta_2$, structure matrix $L_S = [(L\delta_{2^m}^1)^{\tau_a}, \ldots, (L\delta_{2^m}^{2^m})^{\tau_a}, (L\delta_{2^m}^1)^{\tau_b}, \ldots, (L\delta_{2^m}^{2^m})^{\tau_b}] \in \mathcal{L}_{2^n \times 2^{m+n+1}}$, t_{k+1} and t_k satisfy

$$t_{k+1} - t_k = \begin{cases} \tau_a, & v(t_k) = \delta_2^1, \\ \tau_b, & v(t_k) = \delta_2^2. \end{cases} \tag{7}$$

$$x(t+1) = L_S v(t) u(t) x(t), \tag{8}$$

Proposition 1. *If state $\delta_{2^n}^{\alpha}$ is a fixed point of Boolean control network Eq. (4), then Eq. (4) is stabilizable to $\delta_{2^n}^{\alpha}$ under $\Lambda-$nonuniform-sampled inputs if and only if Eq. (8) is stabilizable to $\delta_{2^n}^{\alpha}$.*

Output feedback control design technique[12]

- *The objective of this paper* is to design an output feedback stabilizer in the form of

$$\begin{cases} u_1(t) = h_1(y_1(t), \ldots, y_p(t)), \\ \vdots \\ u_m(t) = h_m(y_1(t), \ldots, y_p(t)) \end{cases} \qquad (4)$$

such that under the control (4), the BCN is stabilized to a given state $X_e = (x_1^e, x_2^e, \ldots, x_n^e)$, where $h_i : \mathcal{D}^p \mapsto \mathcal{D}, i = 1, \ldots, m$ are logical functions.

- $$\begin{cases} x(t+1) = Lu(t)x(t) \\ y(t) = Hx(t), \end{cases} \qquad (5)$$ and $$u(t) = Ky(t). \qquad (6)$$

[12] H. Li and Y. Wang, Output feedback stabilization control design for Boolean control networks, *Automatica*, 2013, 49(12): 3641-3645.

Output feedback control design technique[12]

$$x(t) = LKy(t-1)x(t-1) = LKHx(t-1)x(t-1)$$
$$= LKH\Phi_n x(t-1) = \cdots = (LKH\Phi_n)^t x(0),$$

Theorem 1. *Consider the BCN* (5). *The system is globally stabilizable to* $x_e = \delta_{2^n}^{\alpha}$ *by an output feedback control, if and only if there exist a logical matrix* $K \in \mathcal{L}_{2^m \times 2^p}$ *and an integer* $1 \le \tau \le 2^n$ *such that*

$$(LKH\Phi_n)^{\tau} = \delta_{2^n}[\underbrace{\alpha \ \cdots \ \alpha}_{2^n}]. \tag{7}$$

Output feedback control design technique[12]

- How to **design** output feedback stabilizers?

  (i) Design state feedback stabilizers $\boldsymbol{u}(\boldsymbol{t})=\boldsymbol{G}\boldsymbol{x}(\boldsymbol{t})$. Li et al. (2013)

  (ii) Find the logical matrix $\boldsymbol{K}$ such that $\boldsymbol{G}=\boldsymbol{K}\boldsymbol{H}$.

- Define

$$\Lambda = \{G = \delta_{2^m}[p_1 \ \cdots \ p_{2^n}] : p_i \in P_i, i = 1, \ldots, 2^n\}$$

$$\Theta = \{K = \delta_{2^m}[v_1 \ \cdots \ v_{2^p}] : KH \in \Lambda\}.$$

$$O(k) = \{\delta_{2^n}^i : Col_i(H) = \delta_{2^p}^k\}.$$

$$I(k) = \begin{cases} \bigcap\limits_{\delta_{2^n}^i \in O(k)} P_i, & O(k) \neq \emptyset, \\ \{1, 2, \ldots, 2^m\}, & O(k) = \emptyset. \end{cases} \tag{8}$$

Output feedback control design technique[12]

Theorem 2. *System* (5) *is globally stabilizable to* $x_e = \delta_{2^n}^{\alpha}$ *by an output feedback control* $u(t) = Ky(t)$, $K \in \Theta$, *if and only if*

$$I(k) \neq \emptyset, \quad \forall\, k = 1, 2, \ldots, 2^p. \tag{9}$$

Theorem 3. *Suppose that* (9) *holds. Then, the output feedback gain matrices of system* **(5)** *can be designed in the form of*

$$K = \delta_{2^m}[v_1\ v_2\ \cdots\ v_{2^p}], \quad v_k \in I(k), \tag{10}$$

where $I(k)$ *is given in* (**8**).

Set stabilization

- In some cases, interest lies in whether a system or a collection of interconnected systems converges to or can be **stabilized to a subset of the state space**, instead of to a single point.

- $x(t+1)=Lu(t)x(t)$ (1)

Definition (*Set Stabilizability*). Let $\mathcal{M}$ be a subset of Δ_{2^n}. BCN (1) is said to be $\mathcal{M}$-stabilizable if, for any initial state $x_0 \in \Delta_{2^n}$, there is a control sequence $\mathbf{u}$ and a $\tilde{T}(x_0, \mathbf{u}) \in \mathbb{Z}_{\geq 0}$ such that

$$x(t; x_0, \mathbf{u}) \in \mathcal{M}, \quad \forall t \geq \tilde{T}(x_0, \mathbf{u}).$$

Set stabilization → **Existing methods**

State feedback control design technique

- **Control invariant subset**
- **Pinning control**
- **Event-triggered control**
- **Sampled-data control**

Output feedback control design technique

State feedback control→ **Control invariant subset**[13]

Definition 1. A subset $\mathcal{C} \subseteq \Delta_{2^n}$ is called a control invariant subset of BCN (1) if, for any $x_0 \in \mathcal{C}$, there exists a control sequence $\mathbf{u} = \{u(t)\}_{t \in \mathbb{Z}_{\geq 0}}$ such that $x(t; x_0, \mathbf{u}) \in \mathcal{C}, \forall t \in \mathbb{Z}_{\geq 0}$.

- The union of any two control invariant subsets is still a control invariant subset. The union of all of the control invariant subsets contained in a given subset $\mathcal{M}$ is also a control invariant subset. It is the largest control invariant subset contained in $\mathcal{M}$, and is denoted by $I_C(\mathcal{M})$.

- **How to calculate the largest control invariant subset**?

---

[13] Y. Guo, P. Wang, W. Gui and C. Yang, Set stability and set stabilization of Boolean control networks based on invariant subsets, *Automatica*, 2015, 61: 106-112.

Set feedback control → **Control invariant subset**[13]

- Define: for any nonzero $x \in \mathscr{B}_{m\times 1}$, $\mathscr{S}(x)=\{\mathbf{z}\in\Delta_{\boldsymbol{m}} \mid \mathbf{z}\wedge \boldsymbol{x}=\mathbf{z}\}$

***k*-step controllability matrix** $\mathbf{C}_k = (L \ltimes_{\mathscr{B}} \mathbf{1}_{2^m})^{(k)}$

controllability matrix $\mathbf{C} = (\mathscr{B}) \sum_{k=1}^{2^n} \mathbf{C}_k$

$$\mathscr{R}^{(-k)}(y) := \{x \mid y \in \mathscr{R}^{(k)}(x)\},\ \mathscr{R}^{-1}(y) := \{x \mid y \in \mathscr{R}(x)\}$$

$$\mathscr{R}^{(k)}(\delta_{2^n}^j) = \mathscr{S}(\mathrm{Col}_j(\mathbf{C}_k)),\ \mathscr{R}(\delta_{2^n}^j) = \mathscr{S}(\mathrm{Col}_j(\mathbf{C})),$$
$$\mathscr{R}^{(-k)}(\delta_{2^n}^i) = \mathscr{S}^T(\mathrm{Row}_i(\mathbf{C}_k)),\ \mathscr{R}^{-1}(\delta_{2^n}^i) = \mathscr{S}^T(\mathrm{Row}_i(\mathbf{C})).$$

- *Assume that* $q = |\mathcal{M}|$. *Define* M_0 *as* $\mathrm{Col}_j(M_0) = \begin{cases} \delta_{2^n}^j, & \delta_{2^n}^j \in \mathcal{M} \\ \delta_{2^n}^0, & \delta_{2^n}^j \notin \mathcal{M} \end{cases}$

State feedback control → **Control invariant subset**[13]

Proposition 2. *Define a sequence of Boolean matrices as*

$$M_{i,C} := M_{i-1,C} \ltimes_{\mathscr{B}} \mathbf{C}_1 \ltimes_{\mathscr{B}} M_0$$
$$= (M_0 \ltimes_{\mathscr{B}} \mathbf{C}_1)^{(i)} \ltimes_{\mathscr{B}} M_0, \quad i = 1, 2, \ldots, q,$$

where $M_{0,C} := M_0$.

Then, it holds that $I_C(\mathcal{M}) = \mathscr{S}^T[\mathrm{Row}_\Sigma(M_{q,C})]$.

Lemma 3. *Let* $\mathcal{M} \subseteq \Delta_{2^n}$. *BCN* **(1)** *is* $\mathcal{M}$*-stabilizable if and only if it is* $I_C(\mathcal{M})$*-stabilizable. In addition, for any* $\mathcal{M}$*-stabilizable BCN, there holds* $\tilde{T}_{\mathcal{M}}(x_0) = \tilde{T}_{I_C(\mathcal{M})}(x_0), \forall x_0$. $\tilde{T}_{\mathcal{M}}(x_0) := \min_{\mathbf{u} \in \mathcal{U}_{x_0}} \tilde{T}_{\mathcal{M}}(x_0, \mathbf{u})$

State feedback control → **Control invariant subset**[13]

Proposition 4: *Let* $\mathcal{M} \subseteq \Delta_{2^n}$. *Define* $N_0 \in \mathscr{B}_{2^n \times 2^n}$ *as*

$$N_0 = I_{2^n} \wedge [M_{q,C}^T \ltimes_{\mathscr{B}} M_{q,C}]$$

Then,

(1) *BCN* **(1)** *is* $\mathcal{M}$*-stabilizable if and only if* $\mathrm{Row}_{\Sigma}(N_0\mathbf{C}) = \mathbf{1}_{2^n}^T$;

(2) *If BCN* **(1)** *is* $\mathcal{M}$*-stabilizable, then* $\tilde{T}_{\mathcal{M}}(\delta_{2^n}^j) = \min_{k\in\mathbb{Z}_{\geq 0}} \{k \mid \mathrm{Col}_j(N_0\mathbf{C}_k) \neq 0\}$ *where* $\mathbf{C}_0 := I_{2^n}$.

- How to design the **time-optimal** state feedback controller $u(t) = Fx(t)$?

State feedback control → **Control invariant subset**[13]

- **Partition** the state space Δ_{2^n} into $\tilde{T}_{\mathcal{M}}+1$ subsets $\mathcal{N}_i$, $0 \le i \le \tilde{T}_{\mathcal{M}}$

$$
\begin{aligned}
\mathcal{N}_0 &= I_C(\mathcal{M}) \quad (I_C(\mathcal{M}) \neq \emptyset) \\
\mathcal{N}_1 &= \mathscr{R}^{(-1)}(\mathcal{N}_0) \setminus \mathcal{N}_0 \\
&\vdots \\
\mathcal{N}_{\tilde{T}_{\mathcal{M}}} &= \mathscr{R}^{(-1)}(\mathcal{N}_{\tilde{T}_{\mathcal{M}}-1}) \setminus (\mathcal{N}_{\tilde{T}_{\mathcal{M}}-1} \cup \cdots \cup \mathcal{N}_1 \cup \mathcal{N}_0)
\end{aligned}
$$

- **Define**: for $0 \le i \le \tilde{T}_{\mathcal{M}}$, $\tilde{L} = LW[2^n, 2^m]$.

$$
\operatorname{Col}_j(N_i) = \begin{cases} \delta_{2^n}^j, & \delta_{2^n}^j \in \mathcal{N}_i \\ \delta_{2^n}^0, & \delta_{2^n}^j \notin \mathcal{N}_i. \end{cases} \qquad f(x) = \begin{cases} \left[\operatorname{Row}_\Sigma(N_0\tilde{L}x)\right]^T, & x \in \mathcal{N}_0 \\ \left[\operatorname{Row}_\Sigma(N_{i-1}\tilde{L}x)\right]^T, & x \in \mathcal{N}_i, \end{cases}
$$

and $\mathbf{F} = [f(\delta_{2^n}^1), f(\delta_{2^n}^2), \ldots, f(\delta_{2^n}^{2^n})] \in \mathscr{B}_{2^m \times 2^n}.$

State feedback control → **Control invariant subset**[13]

> ***Proposition 5:*** The controller is a time-optimal M-stabilizer ***if and only if*** F is a logical sub-matrix of $\mathbf{F}$, *i.e.*, $F \wedge \mathbf{F} = F$.

- The Boolean matrix ***F*** characterizes ***all*** of the time-optimal feedback gains and, by ***the construction of F***, ***all of the time-optimal control sequences*** stated in the following result.

> ***Proposition 6:*** A control sequence $\{u(t)\}_{t\in\mathbb{Z}_{\geq 0}}$ is a time-optimal M-stabilizing control sequence ***if and only if*** $u(t) \in \mathscr{S}[\mathbf{F}x(t)]$.

State feedback control → **Pinning control**[14]

- $$x(t+1) = Lx(t), \quad L = *_{i=1}^{n} M_i \in \mathcal{L}_{2^n \times 2^n} \tag{2}$$

- The index set of pinning controlled nodes is assumed to be $\Theta = \{\eta_1, \eta_2, \ldots, \eta_v\}$, and system (2) under pinning control is given as

$$x_j(t+1) = F_j(u_j(t), x_1(t), \ldots, x_n(t)), \quad j \in \Theta, \tag{3}$$
$$x_i(t+1) = f_i(x_1(t), \ldots, x_n(t)), \quad i \in \{1, 2, \ldots, n\} \backslash \Theta,$$

- Let $\mathcal{M} \subseteq \Delta_{2^n}$ be an arbitrary given set. Our objective is to realize set $\mathcal{M}$-stable for system (2) under pinning control strategy.

[14] R. Liu, J. Lu, J. Lou and et al., Set stabilization of Boolean networks under pinning control strategy, *Neurocomputing*, 2017, 260: 142-148.

State feedback control → **Pinning control**[14]

- *Step* 1): convert the set $\mathcal{M}$ to be an invariant subset
- Assume that $\mathcal{M} = \{\delta_{2^n}^1, \delta_{2^n}^2, \ldots, \delta_{2^n}^m\}$.

$$\Omega = \{\nu | L\delta_{2^n}^{\nu} \notin \mathcal{M}, \nu = 1, 2, \ldots, m\} = \{\nu_1, \nu_2, \ldots, \nu_r\}$$

➡ $L\delta_{2^n}^{\nu_j} > \delta_{2^n}^m, \ j = 1, 2, \ldots, r.$

➡ change the ν_jth column of L into $\delta_{2^n}^q, \ q \in \{1, 2, \ldots, m\}$

- principle:

$$col_{\nu_j}(M_i) = \begin{cases} \text{unchanged}, & \text{if } col_{\nu_j}(M_i) = \delta_2^1, \\ \delta_2^1, & \text{if } col_{\nu_j}(M_i) = \delta_2^2, \end{cases}$$

$$i = 1, 2, \ldots, N_{\nu_j}.$$

State feedback control → **Pinning control**[14]

- *Step 2)*: $E_k(\mathcal{M}) = \{\bigcup_{r=1}^{m} E_k(\delta_{2^n}^r), \delta_{2^n}^r \in \mathcal{M}\}$, $\mathbb{E}(\mathcal{M}) = \{\bigcup_{k=1}^{2^n} E_k(\mathcal{M})\}$.
- Let $\Lambda = \{\lambda : \delta_{2^n}^{\lambda} \in \Delta_{2^n} \backslash \mathbb{E}(\mathcal{M})\}$.

  Let Ω_{sub} be the set consisting of all fixed points and fundamental circles in the set $\Delta_{2^n} \backslash E(M)$. Since the states $x(0) \in \{\delta_{2^n}^{\lambda} : \lambda \in \Lambda\}$ must reach a stable set $\Omega_{sub} \subseteq \{\delta_{2^n}^{\lambda} : \lambda \in \Lambda\}$, we can construct the set Ω^* ($\Omega^* \subseteq \Omega_{sub}$) as follows: Ω^* includes all of the fixed points in Ω_{sub} and at least one point in every fundamental circle. Hence, $\mathbb{E}(\{\delta_{2^n}^{\mu} : \mu \in \Omega^*\}) = \{\delta_{2^n}^{\lambda} : \lambda \in \Lambda\}$.

  Let $\Omega^* = \{\mu_1, \mu_2, \ldots, \mu_p\}$.
- principle:

$$col_{\mu_j}(M_i) = \begin{cases} \text{unchanged}, & \text{if } col_{\mu_j}(M_i) = \delta_2^1 \\ \delta_2^1, & \text{if } col_{\mu_j}(M_i) = \delta_2^2 \end{cases} \quad i = 1, 2 \ldots, N_{\mu_j}.$$

State feedback control → **Pinning control**[14]

Proposition 1. *After Steps 1) and 2), system* **(2)** *with structure matrix L' becomes $\mathcal{M}$-stable.*

- Now, we will give the design procedure of pinning control.
- M_j alter to M'_j, $j \in \Theta$, $\Theta = \{\eta_1, \eta_2, \ldots, \eta_v\}$

$$f_j(x_1, \ldots, x_n) \longrightarrow F_j(u_j, x_1, \ldots, x_n) = u_j(x_1, \ldots, x_n) \oplus_j f_j(x_1, \ldots, x_n)$$
$$= M_{\oplus_j} \overline{M}_j x_1 \cdots x_n M_j x_1 \cdots x_n$$
$$= M_{\oplus_j} (\overline{M}_j * M_j) x_1 \cdots x_n,$$

Solve $M_{\oplus_j}$ and $\overline{M}_j$ from $M_{\oplus_j}(\overline{M}_j * M_j) = M'_j$

State feedback control → **Pinning control**[14]

Remark 1: It should be noted that $M_{\oplus_j} \in M_{\oplus}$ is just a sufficient condition for the solvability of $M_{\oplus_j}(\overline{M}_j * M_j) = M'_j$.

Remark 2: Compared with (**F. Li 2016**), if we just need some of feasible state feedback controllers, our method would be **better**, since we do not need to discuss all the situations of the variables . By using $\boldsymbol{M_{\oplus}}$, we can **reduce the computational complexity** and obtain some of feasible state feedback control matrices at the same time.

State feedback control → **Event-triggered control**[15]

- $\begin{cases} x(t+1) = Lu(t)x(t)\xi(t), \\ y(t) = Hx(t), \end{cases}$ (1) → $x(t+1) = \hat{L}\xi(t)u(t)x(t), \hat{L} = LW_{[k^q, k^{m+n}]}$

$u(t) = \Psi(t, x(0))x(t),$

$\hat{L} = [\hat{L}_1 \ \hat{L}_2 \ \cdots \ \hat{L}_{k^q}],$

$\hat{L}_i = [\hat{L}_i^1 \ \hat{L}_i^2 \ \cdots \ \hat{L}_i^{k^m}].$

- Given a nonempty set $A \subseteq \Delta_{k^n}$, for $t \geq 1$, $t \in \mathbb{Z}_+$, $\Upsilon_0(A) := A$, define

$$\Upsilon_t(A) = \{\delta_{k^n}^{\beta} : \text{ there exists an integer } 1 \leq j_\beta \leq k^m$$

$$\text{such that} \sum_{\delta_{k^n}^{\beta'} \in \Upsilon_{t-1}(A)} \sum_{i=1}^{k^q} (\hat{L}_i^{j_\beta})_{\beta', \beta} = k^q\}, t \geq 2$$

---

[15] Y. Li, H. Li and W. Sun, Event-triggered control for robust set stabilization of logical control networks, *Automatica*, 2018, 95: 556-560.

State feedback control → **Event-triggered control**[15]

Theorem 1. *Given a nonempty set $A \subseteq \Delta_{2^n}$ and an initial state $x(0) = \delta_{k^n}^{\alpha}$. Assume that $A \subseteq \Upsilon_1(A)$. System (1) is robustly stabilizable to A under the control , if and only if there exists a positive integer T such that $x(0) \in \Upsilon_T(A)$.*

Remark 1: From example in this study, the time-variant state feedback control obtained is in change every moment before some time ***t***, which needs many computation costs. Motivated by this, we will propose the event-triggered control to reduce the costs in the following.

State feedback control → **Event-triggered control**[15]

- Given a nonempty set $A \subseteq \Delta_{k^n}$ and an initial state $x(0) = \delta_{k^n}^{\alpha}$.

$$x(t+1) = Blk_{\alpha}\left(\ltimes_{i=t}^{0} (L\Psi(i, x(0))M_{r,k^n}) \right) \ltimes_{j=0}^{t} \xi(j)$$

- From the arbitrariness of $\ltimes_{j=0}^{t}\xi(j)$, $x(t+1)$ forms a set, denoted by

$$\Omega(t+1) = Col\left(Blk_{\alpha}\left(\ltimes_{i=t}^{0} (L\Psi(i, x(0))M_{r,k^n}) \right) \right).$$

- Then, the event-triggered condition is given as

$$d_H(\Omega(t+1), A) > 0, \quad \textbf{(2)}$$

where $d_H(\Omega(t+1), A)$ denotes the Hausdorff distance[1]

---

[1] The Hausdorff distance between two nonempty set M and N is $d_H(M, N) = \max_{a\in M}\{\min_{b\in N}\{d(a, b)\}\}$, where $d(a, b)$ is the Euclidean distance between a and b.

State feedback control → **Event-triggered control**[15]

- Given $\Psi(0, x(0)) \in \mathcal{L}_{k^m \times k^n}$ under which $x(0) \in \Upsilon_1(A)$, we keep $\Psi(t, x(0)) = \Psi(0, x(0))$ until the event-triggered condition occurs.
- Denote $t_1 = \min\{t > 0 : d_H(\Omega(t+1), A) > 0\}$, that is, t_1 is the first triggering time, which implies that $\Psi(t_1, x(0)) = \Psi(0, x(0))$ should be updated. Then, we keep $\Psi(t, x(0)) = \Psi(t_1, x(0))$ until the event-triggered condition occurs again.
- Let $t_2 = \min\{t > t_1 : d_H(\Omega(t+1), A) > 0\}$ be the second triggering time at which $\Psi(t_2, x(0)) = \Psi(t_1, x(0))$ should be updated.
- Keep this procedure going, one can obtain the sequence of triggering times: $t_1 < t_2 < \cdots < t_s < \cdots,$ which corresponds to a sequence of control updates: $\Psi(t_1, x(0)), \ldots, \Psi(t_s, x(0)), \ldots$.

State feedback control → **Event-triggered control**[15]

- the state feedback event-triggered controller can be designed as follows: $u(t) = \Psi(t_\rho, x(0))x(t),\ t \in [t_\rho, t_{\rho+1}) \cap \mathbb{N}$, where $\rho = 0, 1, \ldots, s, \ldots$, and $t_0 := 0$.

Theorem 2. *Given a nonempty set $A \subseteq \Delta_{k^n}$ and an initial state $x(0) = \delta_{k^n}^{\alpha}$. System (1) is robustly stabilizable to A under the event-triggered condition* (2) *, if $A \subseteq \Upsilon_1(A)$ and $x(0) \in \Upsilon_1(A)$.*

State feedback control → **Sampled-data control**[16]

➤ ***Objective***: Design the following **sampled-data state feedback** controller :

$$\begin{cases} u_1(t) = e_1(x_1(t_l), \ldots, x_m(t_l)) \\ u_2(t) = e_2(x_1(t_l), \ldots, x_m(t_l)) \\ \quad\vdots \\ u_m(t) = e_m(x_1(t_l), \ldots, x_m(t_l)) \end{cases} \quad t_l \le t < t_{l+1}$$

such that BCN (1) can be stabilized to the given set ***S***, where constant SP $\tau := t_{l+1} - t_l \in \mathbb{Z}_+$, $t_l = l\tau \ge 0$, $l = 0, 1, \ldots$ are sampling instants.

➤ Algebraic form:

$$u(t) = Ex(t_l), \ \ t_l \le t < t_{l+1} \qquad (6)$$

[16] S. Zhu, Y. Liu, J. Lou, J. Lu and F. E. Alsaadi, Sampled-data state feedback control for the set stabilization of Boolean control networks, IEEE Transactions on Systems, Man, and Cybernetics: Systems, 2020, 50(4): 1580-1589.

State feedback control → **Sampled-data control**[16]

- $$x(t+1) = Lu(t)x(t) = LEx(t_l)x(t)$$
  $$= \widetilde{L}x(t_l)x(t), \quad t_l \le t < t_{l+1} \qquad (7)$$
  where $t_{l+1} - t_l = \tau$ and $\widetilde{L} = L \ltimes E \in \mathcal{L}_{2^n \times 2^{2n}}$.
- when $0 \le t \le \tau$,
  $$x(t) = LEx(0)x(t-1) = (LEx(0))^t x(t-1).$$
- when $\tau < t \le 2\tau$, $x(\tau)$ acts as another initial state,
  $$x(t) = LEx(\tau)x(t-1) = (LEx(\tau))^{t-\tau} x(\tau).$$
- Split L to 2^m matrix as $[L_1 \ L_2 \ \cdots \ L_{2^m}]$. When $x(t_l) = \delta_{2^n}^i$,

$t_l = l\tau$, for $t_l \le t < t_{l+1}$
$$x(t) = x(t_l) = \delta_{2^n}^i$$
$$u(t) = Ex(t_l) = \delta_{2^m}[p_1 \ p_2 \ \cdots \ p_{2^n}]\delta_{2^n}^i.$$

State feedback control → **Sampled-data control**[16]

Design SDSFC ← **Calculate** the **SPCIS** ← **Find** the **Sampled Point Set**

Definition **1** *(Sampled Point Set):* Given a set $S \subseteq \Delta_{2^n}$, the set $S^* \subseteq S$ is called the sampled point set of BCN (1) with SP τ, if for any $x(0) \in S^*$, there exists a control sequence $\mathbf{u}$ such that $x(1) \in S, x(2) \in S, \ldots, x(\tau) \in S$.

Definition **2** *(SPCIS):* The set $\widetilde{S} \subseteq S$ is said to be an SPCIS of S for BCN (1) under SDSFC, if for any $x(t_l) \in \widetilde{S}$, there exists a sampled-data state feedback controller (6) such that $x(t_l+1) \in S, x(t_l+2) \in S, \ldots, x(t_l+\tau) = x(t_{l+1}) \in \widetilde{S}$. A set $\widetilde{S}^*$ is called the largest SPCIS of BCN (1) under SDSFC, if it contains the largest number of elements among all SPCISs of $\widetilde{S}$.

State feedback control → **Sampled-data control**[16]

Theorem 1 The set S has an SPCIS of BCN (7) under SDSFC (6) if and only if there exists a set $\widetilde{S} \subseteq S^*$ such that for any $\delta_{2^n}^{i_a} \in \widetilde{S}$, we can find a control input $\delta_{2^m}^{\alpha} \in \Delta_{2^m}$ guaranteeing that the following formula holds:

$$
\begin{gathered}
\mathrm{Col}_{i_a}(L_\alpha) \in S \\
L_\alpha \mathrm{Col}_{i_a}(L_\alpha) \in S \\
\vdots \\
(L_\alpha)^{\tau-1}\mathrm{Col}_{i_a}(L_\alpha) \in \widetilde{S}.
\end{gathered}
$$

State feedback control → **Sampled-data control**[16]

➤ **Find the largest SPCIS** $\tilde{S}^*$ in set S^*:

Step 1: Find $\delta_{2^n}^{i_a} \in S^*$, for any input $\delta_{2^n}^{\alpha} \in \Delta_{2^m}$, there exists an integer $j \in [1, \tau - 2]$ such that $(L_\alpha)^j \mathrm{Col}_{i_a}(L_\alpha) \notin S$ or $(L_\alpha)^{\tau-1} \mathrm{Col}_{i_a}(L_\alpha) \notin S^*$. Let S_1 be the set of all $\delta_{2^n}^{i_a}$, if $S_1 = \emptyset$, for any initial state $x(0) \in S$ and $t > 0$, $x(t) \in S$, so $\widetilde{S^*} = S^*$; otherwise go to step 2.

Step 2: Let $S_1^* = S^* \backslash S_1$, find state $\delta_{2^n}^{i_b} \in S_1^*$, for any input $\delta_{2^n}^{\alpha} \in \Delta_{2^m}$, there exists an integer $j \in [1, \tau - 2]$ such that $(L_\alpha)^j \mathrm{Col}_{i_b}(L_\alpha) \notin S$ or $(L_\alpha)^{\tau-1} \mathrm{Col}_{i_b}(L_\alpha) \notin S_1^*$. Let S_2 be the set of all such $\delta_{2^n}^{i_b}$, if $S_2 = \emptyset$, $\widetilde{S^*} = S_1^*$; otherwise go to step 3.

Step 3: Continue a similar process until we find the set S_k^* satisfying for any $\delta_{2^n}^{i_c} \in S_k^*$, there exists $\delta_{2^m}^{\alpha} \in \Delta_{2^m}$ such that $(L_\alpha)^j \mathrm{Col}_{i_c}(L_\alpha) \in S$, $j = 1, 2, \ldots, \tau - 2$, and $(L_\alpha)^{\tau-1} \mathrm{Col}_{i_c}(L_\alpha) \in S_k^*$. Then $\widetilde{S^*} = S_k^*$ is the largest SPCIS.

State feedback control → **Sampled-data control**[16]

- Denote by $\Omega_l(\widetilde{S}^*)$ the set of states that can be steered to $\widetilde{S}^*$ in $l\tau$ steps under some control sequence, that is,

$$\begin{cases} \Omega_1(\widetilde{S}^*) = \{x_0 \in \Delta_{2^n} : (LEx_0)^\tau x_0 \in \widetilde{S}^*\} \\ \Omega_{l+1}(\widetilde{S}^*) = \{x_0 \in \Delta_{2^n} : (LEx_0)^\tau x_0 \in \Omega_l(\widetilde{S}^*)\}. \end{cases}$$

Theorem ***2*** System (1) can be globally stabilized to the set S by SDSFC (6), if and only if the following two conditions are satisfied simultaneously.

1) $\widetilde{S}^* \neq \emptyset$.

2) there exists an integer T such that $\Omega_T(\widetilde{S}^*) = \Delta_{2^n}$.

State feedback control → **Sampled-data control**[16]

- $\Delta_{2^n} = \Omega_1(\widetilde{S}^*) \cup (\Omega_2(\widetilde{S}^*) \backslash \Omega_1(\widetilde{S}^*)) \cup \cdots \cup (\Omega_T(\widetilde{S}^*) \backslash \Omega_{T-1}(\widetilde{S}^*)).$
- for every $1 \leq i \leq 2^n$, $\exists 1 \leq l_i \leq T$ such that

$$\delta_{2^n}^i \in \Omega_{l_i}(\widetilde{S}^*) \backslash \Omega_{l_i-1}(\widetilde{S}^*) \text{ with } \Omega_0(\widetilde{S}^*) = \emptyset.$$

- If $l_i = 1$ and $\delta_{2^n}^i \in \widetilde{S}^*$, let p_i be the solution such that

$$\begin{aligned}
&\mathrm{Col}_i(L_{p_i}) \in S \\
&(L_{p_i})\mathrm{Col}_i(L_{p_i}) \in S \\
&\qquad \vdots \\
&(L_{p_i})^{\tau-2}\mathrm{Col}_i(L_{p_i}) \in S \\
&(L_{p_i})^{\tau-1}\mathrm{Col}_i(L_{p_i}) \in \widetilde{S}^*.
\end{aligned} \tag{8}$$

State feedback control → **Sampled-data control**[16]

- If $l_i = 1$ and $\delta_{2^n}^i \notin \widetilde{S}^*$, let p_i be the solution such that

$$(L_{p_i})^{\tau-1}\mathrm{Col}_i(L_{p_i}) \in \widetilde{S}^*. \quad (9)$$

- If $2 \le l_i \le T$, let p_i be the solution such that

$$(L_{p_i})^{\tau-1}\mathrm{Col}_i(L_{p_i}) \in \Omega_{l_i}(\widetilde{S}^*)\backslash\Omega_{l_i-1}(\widetilde{S}^*). \quad (10)$$

Theorem **3** If there exists SDSFC (6) such that system can be stabilized to the set S, p_i is the solution of (8)-(10). Then the feedback law (6) with the state feedback matrix E is given as

$$E = \delta_{2^m}[p_1\ p_2\ \cdots\ p_{2^n}],$$

which globally stabilizes BCN (1) to S.

State feedback control → **Sampled-data control**[17]

Theorem 1. *Assume that* $|\mathcal{M}| = q$ *and* $q \leq \tau$, *where* τ *is sampling period. Define* M_0 *as*

$$Col_j(M_0) = \begin{cases} \delta_{2^n}^j, & \delta_{2^n}^j \in \mathcal{M} \\ \delta_{2^n}^0, & \delta_{2^n}^j \notin \mathcal{M}, \end{cases}$$

and a set of Boolean matrices ${}^lM_i, 1 \leq l \leq 2^m, 1 \leq i \leq q$, *as*

$${}^lM_i := M_0{}^lL{}^lM_{i-1} = (M_0{}^lL)^i(M_0).$$

Then $I_c(\mathcal{M}) = \bigcup_{l=1}^{2^m} \zeta^T[Row_\Sigma({}^lM_q)]$ *holds.*

[17] L. Sun, J. Lu, J. Lou and L. Li, Set stabilization of Boolean networks via sampled-data control, *Asian Journal of Control*, 2019, 21(6): 2685-2690.

State feedback control → **Sampled-data control**[17]

Theorem 2. *Assume that* $|\mathcal{M}| = q$ *and* $q > \tau$, *where* τ *is sampling period and* $k\tau < q \le (k+1)\tau$, $k \in \mathbb{Z}_{>0}$. *Define* M_{y0} *as*

$$Col_j(M_{y0}) = \begin{cases} \delta^j_{2^{(k+1)n}}, & \delta^j_{2^{(k+1)n}} \in \mathcal{M}_y \\ \delta^0_{2^{(k+1)n}}, & \delta^j_{2^{(k+1)n}} \notin \mathcal{M}_y, \end{cases}$$

and a collection of Boolean matrices ${}^l\tilde{M}_i, 1 \le l \le 2^{(k+1)m}, 1 \le i \le \tau$, *as*

$${}^l\tilde{M}_i := M_{y0}{}^l\tilde{L}{}^l\tilde{M}_{i-1} = (M_{y0}{}^l\tilde{L})^i M_{y0}.$$

Then, it holds that $I_c(\mathcal{M}) \subseteq \{x(0) | \ltimes_{i=0}^{k} x(i\tau) \in K\}$ *where* $K = \bigcup_{l=1}^{2^{(k+1)m}} \zeta^T[Row_{\sum}({}^l\tilde{M}_\tau)]$.

State feedback control → **Sampled-data control**[17]

Let $\Gamma_0 := I_c(\mathcal{M})$. Choose any nonempty set $\Gamma_\tau \subseteq E_\tau(I_c(\mathcal{M}))\backslash\Gamma_0$. Then choose any nonempty set $\Gamma_{2\tau} \subseteq R_\tau(\Gamma_\tau)\bigcap[E_{2\tau}(I_c(\mathcal{M}))\backslash(\Gamma_0\bigcup\Gamma_\tau)]$. For any $l \in \mathbb{Z}_{\geq 0}$, choose any nonempty set $\Gamma_{l\tau} \subseteq R_\tau(\Gamma_{(l-1)\tau})\bigcap[E_{l\tau}(I_c(\mathcal{M}))\backslash(\bigcup_{i=0}^{l-1}\Gamma_{i\tau})]$.

Theorem 3. *For any given subset $\mathcal{M}$, system* (1) *can be $\mathcal{M}$-stabilizable under SDSFC* (6) *if and only if*

(I). The set $I_c(\mathcal{M})$ is nonempty.

(II). There exists an N, such that $\bigcup_{i=0}^{N}\Gamma_{i\tau} = \Delta_{2^n}$.

State feedback control → **Sampled-data control**[17]

Algorithm 1.

Step 1. *Choose one of the complete families of reachable sets. Without loss of generality, let the complete family of reachable sets be* $\{\Gamma_0, \Gamma_\tau, \ldots, \Gamma_{N\tau}\}$. *If there is no complete family of reachable sets, then* E *does not exist.*

Step 2. *For any initial state* $x_0 = \delta_{2^n}^{\mu}$, $\mu = 1, 2, \ldots, 2^n$, *since* $\bigcup_{i=0}^{N} \Gamma_{i\tau} = \Delta_{2^n}$ *and* $\Gamma_{i\tau} \subseteq R_\tau(\Gamma_{(i-1)\tau})$, $i = 1, \ldots, N$, *one can find a unique integer* $0 \leq k_\mu \leq N$ *such that* $\delta_{2^n}^{\mu} \in \Gamma_{k_\mu \tau}$. *If* $\delta_{2^n}^{\mu} \in I_c(\mathcal{M})$, *finding all possible integers* $1 \leq p_\mu \leq 2^m$, *such that* $L\delta_{2^m}^{p_\mu}\delta_{2^n}^{\mu} \in I_c(\mathcal{M})$, $(L\delta_{2^m}^{p_\mu})^2\delta_{2^n}^{\mu} \in I_c(\mathcal{M})$, $\ldots$, $(L\delta_{2^m}^{p_\mu})^\tau\delta_{2^n}^{\mu} \in I_c(\mathcal{M})$. *If* $\delta_{2^n}^{\mu} \notin I_c(\mathcal{M})$, *finding all possible integers* $1 \leq p_\mu \leq 2^m$, *such that* $(L\delta_{2^m}^{p_\mu})^\tau\delta_{2^n}^{\mu} \in \Gamma_{(k_\mu - 1)\tau}$.

Step 3. *The feedback matrix* $E = \delta_{2^m}[p_1, \ldots, p_{2^n}]$ *can be obtained.*

Output feedback control design technique[18]

- $x(t+1) = Fx(t)u(t)$ (3a)

  $y(t) = Hx(t)$ (3b)

Definition 1: For a given set $\mathcal{M} \subseteq \Delta_N$, system (3) is said to be $\mathcal{M}$-stabilizable by means of time-invariant output feedback if there exists a matrix $K_y \in \mathcal{L}_{M \times P}$ such that under the output feedback control law $u(t) = K_y y(t)$, $t \in \mathbb{N}$, for any initial state $x(0) \in \Delta_N$, one can find an integer $\tau \in \mathbb{N}$ such that

$$x(t; x(0), u) \in \mathcal{M} \quad \forall t \geq \tau.$$

[18] R. Liu, J. Lu, W. Zheng and J. Kurth, Output feedback control for set of Boolean control networks, *IEEE Transactions on Neural Networks and Learning Systems*, DOI: 10.1109/TNNLS.2019.2928028.

Output feedback control design technique[18]

Definition **2:** A set $S \subseteq \Delta_N$ is called an OFCI subset of BCN (3) if, for any state $x_0 \in S$, there exists a control sequence $u = \{u(t)\}_{t\in\mathbb{N}}$ such that $x(t; x_0, u) \in S$ and u satisfies the output feedback law $u(t) = K_y y(t)$, $t \in \mathbb{N}$.

➤ **Design** the **output feedback stabilizers** **(Spanning In-Tree Method, *Algorithm4*)** ← **Calculate** the **OFCI subsets** **(*Algorithms 2 and 3*)** ← **Calculate** the **SFCI sets** **(*Algorithm 1*)**

Output feedback control design technique[18]

♣ Calculate the SFCI sets

➤ $F = [F_1 \cdots F_N]$

Algorithm 1 This Algorithm Is Used to Obtain the Largest SFCI Subset of $\mathcal{M}$

For any $\delta_N^{\Lambda} \in \mathcal{M}$, $S_0 = \mathcal{M}$, $Q = \emptyset$, and iterate:

$$Q = \{\delta_N^{\Lambda} | col(F_{\Lambda}) \bigcap S_i = \emptyset, \text{ for every } \delta_N^{\Lambda} \in S_i\},$$
$$S_{i+1} = S_i \backslash Q.$$

Terminate when $S_{i+1} = S_i$.

Proposition ***1:*** The set S_{ω} obtained in Algorithm 1 is the largest SFCI subset of $\mathcal{M}$.

Output feedback control design technique[18]

♣ Calculate the OFCI sets

- Without loss of generality, we assume that $S_\omega = \{\delta_N^1, \ldots, \delta_N^\tau\}$, where $\tau = |S_\omega| \le T$. Let

$$\mathcal{P}(i) = \{\delta_M^j | \mathrm{col}_j(F_i) \in S_\omega, \quad j \in [1, M]\}.$$

- Define:

$$\Upsilon = \{K' | K' = [\delta_M^{\Lambda_1} \ \ldots \ \delta_M^{\Lambda_\tau} \ \mathbf{0}_{M\times(N-\tau)}] \in \mathcal{B}_{M\times N}, \ \delta_M^{\Lambda_i} \in \mathcal{P}(i), \ i \in [1, \tau]\}$$

$\Gamma = |\Upsilon| = \Pi_{i=1}^{\tau} |\mathcal{P}(i)| \le M^T$. For S_ω, construct

$$\begin{cases} x(t+1) & = [F_1 \ \cdots \ F_\tau \ \mathbf{0}_{N\times(NM-\tau M)}]x(t)u(t) \\ u(t) & = K'x(t) \end{cases} \tag{7}$$

Output feedback control design technique[18]

♣ Calculate the OFCI sets

- $$\begin{aligned} x(t+1) &= [F_1 \ \cdots \ F_\tau \ \mathbf{0}_{N\times(NM-\tau M)}]x(t)u(t) \\ &= [F_1 \ \cdots \ F_\tau \ \mathbf{0}_{N\times(NM-\tau M)}]x(t)K'x(t) \\ &= [\mathrm{col}_{\Lambda_1}(F_1) \ \cdots \ \mathrm{col}_{\Lambda_\tau}(F_\tau) \ \mathbf{0}_{N\times(N-\tau)}]x(t) \\ &= Lx(t) \end{aligned} \qquad (8)$$

where $L = [\mathrm{col}_{\Lambda_1}(F_1) \ \cdots \ \mathrm{col}_{\Lambda_\tau}(F_\tau) \ \mathbf{0}_{N\times(N-\tau)}] \in \mathcal{B}_{N\times N}$

- Let

$$L' = \begin{bmatrix} L_{1,1} & \dots & L_{1,\tau} \\ \vdots & \vdots & \vdots \\ L_{\tau,1} & \dots & L_{\tau,\tau} \end{bmatrix}$$

Output feedback control design technique[18]

♣ Calculate the OFCI sets

Algorithm 3 Algorithm Is Presented to Calculate the Necessary Output Feedback Compatible Subsets of S_ω ($\mathcal{M}$) and the Associated K

Input: S_ω, Υ
Output: $(\mathcal{M}', K)$

for all K' in Υ **do**
  construct system (8) corresponding to K'
  calculate $\mathcal{C}$ of system (8) through Algorithm 2
  suppose $\mathcal{C} = \{\mathcal{C}_1, \ldots, \mathcal{C}_\eta\}$, where $\eta = |\mathcal{C}|$
  for all possible combinations of $\mathcal{C}_1, \ldots, \mathcal{C}_\eta$ **do**
    denote by $\mathcal{M}'$ the set consisting of all the states in the combination of $\mathcal{C}_1, \ldots, \mathcal{C}_\eta$
    if $\mathcal{M}'$ is output feedback compatible under K' **then**
      construct matrix K as

$$\mathrm{col}_i(K) = \begin{cases} col_i(K'), & \forall \delta_N^i \in \mathcal{M}' \\ *, & \text{otherwise} \end{cases}$$

      output $(\mathcal{M}', K)$
    end if
  end for
end for

Algorithm 2 This Algorithm Is Used to Calculate All Elementary Cycles for System (8)

Input: L', S_ω
Output: $\mathbb{C}$

initialize length variable $l = 1$, matrix $\mathbf{L}_1 = L'$, set $\mathbb{C}' = \emptyset$, set $S' = S_\omega = \{\delta_N^1, \ldots, \delta_N^\tau\}$
while $\mathbf{L}_l \neq \mathbf{0}_{\tau\times\tau}$ **do**
  $\mathbb{C}' = \{\delta_N^j \in S' | (\mathbf{L}_l)_{j,j} \neq 0, j \in [1, \tau]\}$
  if $\mathbb{C}' \neq \emptyset$ **then**
    $S' = S' \backslash \mathbb{C}'$
    for all $\delta_N^j \in \mathbb{C}'$ **do**
      set $row_j(\mathbf{L}_l) = \mathbf{0}_{1\times\tau}$
    end for
    while $\mathbb{C}' \neq \emptyset$ **do**
      find a $\delta_N^j \in \mathbb{C}'$, and let $\delta_N^{\lambda_1} = \delta_N^j$
      initialize set $\mathbb{C} = \{\delta_N^{\lambda_1}\}$
      if $l \geq 2$ **then**
        for $k = 2 : l$ **do**
          $\delta_N^{\lambda_k} = L\delta_N^{\lambda_{k-1}}$
          $\mathbb{C} = \mathbb{C} \bigcup \{\delta_N^{\lambda_k}\}$
        end for
      end if
      output $\mathbb{C}$
      $\mathbb{C}' = \mathbb{C}' \backslash \mathbb{C}$
    end while
  end if
  $\mathbf{L}_{l+1} = \mathbf{L}_l L'$
  $l = l + 1$
end while

Output feedback control design technique[18]

♣ Design the output feedback stabilizers

➢ Let $\mathbf{W} = \left[\vee_{i=1}^{M}\text{col}_i(F_1) \;\cdots\; \vee_{i=1}^{M}\text{col}_i(F_N)\right]$ be the adjacency matrix of G and G is the associated digraph of system (3).

➢ Let $\mathbf{W}' = \varphi(\mathbf{W}, S)$, $\bar{S} = [1, N] \backslash S = \mathcal{M}'$.

$$\varphi(\mathbf{W}, S) = \begin{bmatrix} \mathbf{W}_{\upsilon_1,\upsilon_1} & \cdots & \mathbf{W}_{\upsilon_1,\upsilon_\tau} & 0 \\ \vdots & \vdots & \vdots & \vdots \\ \mathbf{W}_{\upsilon_\tau,\upsilon_1} & \cdots & \mathbf{W}_{\upsilon_\tau,\upsilon_\tau} & 0 \\ [\text{Row}_{\bar{S}}(\mathbf{W})]_{\upsilon_1} & \cdots & [\text{Row}_{\bar{S}}(\mathbf{W})]_{\upsilon_\tau} & 0 \end{bmatrix} \Rightarrow \text{a digraph } G'$$

➢ For a matrix $D \in \mathcal{L}_{N\times N}$, an integer $i \in [1, N]$, we define

$$\mathcal{P}'(i, D) = \left\{\delta_M^j \,|\, \text{col}_i(D) = \text{col}_j(F_i), \quad \forall j \in [1, M]\right\}.$$

Output feedback control design technique[18]

♣ Design the output feedback stabilizers

Algorithm 6 This Algorithm Calculates All Output Feedback Matrices K_y

Input: $(\mathcal{M}', K)$

Output: K_y

for all $(\mathcal{M}', K)$ **do**
  suppose $[1, N]\backslash\mathcal{M}' = \{\upsilon_1, \ldots, \upsilon_\tau\}$, $\tau = |[1, N]\backslash\mathcal{M}'|$
  let $\mathbf{W}' = \varphi(\mathbf{W}, [1, N]\backslash\mathcal{M}')$
  associate digraph G' with matrix $\mathbf{W}'$
  set root $\mathbf{r} = \tau + 1$
  use Algorithm 5 to obtain all the spanning in-trees $T'_{\mathbf{r}}$ in digraph G'
  for each $T'_{\mathbf{r}}$ in G' **do**
    denote the adjacency matrix of $T'_{\mathbf{r}}$ by W'_T
    calculate all the matrices $W_T \leq \mathbf{W}$ such that $col_j(W_T) = F_j \ltimes col_j(K)$, $\forall j \in \mathcal{M}'$, $W_T \in \mathcal{L}_{N\times N}$ and $\varphi(W_T, [1, N]\backslash\mathcal{M}') = W'_T$
    for each W_T **do**
      initialize a series of sets $S_l = \Delta_M$, $l = 1, \ldots, P$
      for all $l' \in [1, N]$ **do**
        $\delta_P^l = H\delta_N^{l'}$
        if $l' \in \mathcal{M}'$ **then**
          $S_l = S_l \bigcap \{col_{l'}(K)\}$
        end if
        if $l' \in [1, N]\backslash\mathcal{M}'$ **then**
          $S_l = S_l \bigcap \mathcal{P}'(l', W_T)$
        end if
      end for
      if $S_l \neq \emptyset$, $l = 1, \ldots, P$ **then**
        set $K_y = [\kappa_1, \ldots, \kappa_P]$, where $\kappa_l \in S_l$, $l \in [1, P]$
      end if
    end for
  end for
end for

Set Stabilization ⟶ Applications

Output regulation[19]

Output tracking[20]

Synchronization[21-24]

[19] H. Li, L. Xie and Y. Wang, **Output regulation** of Boolean control networks, *IEEE Transactions on Automatic Control*, 2017, 62(6): 2993-2998.

[20] H. Li, Y. Wang and L. Xie, **Output tracking control** of Boolean control networks via state feedback: Constant reference signal case, *Automatica*, 2015, 59: 54-59.

[21] Y. Liu, L. Sun, J. Lu and J. Liang, **Feedback controller** design for the **synchronization** of Boolean control networks, *IEEE Transactions on Neural Networks & Learning Systems*, 2017, 27(9): 1991-1996.

[22] F. Li, **Pinning control** design for the **synchronization** of two coupled Boolean networks, *IEEE Transactions on Circuits & Systems II: Express Briefs*, 2016, 63(3): 309-313.

[23] Y. Liu, L. Tong, J. Lou, J. Lu and J. Cao, **Sampled-data control** for the **synchronization** of Boolean control networks, IEEE Transactions on Cybernetics, 2019, 49(2): 726-732.

[24] J. Yang, J. Lu, L. Li, Y. Liu, Z. Wang and F. E. Alsaadi, **Event-triggered** control for the **synchronization** of Boolean control networks, *Nonlinear Dynamics*, 2019, 96: 1335-1344.

………………………………

Applications ⟶ Output regulation[19]

$$
\begin{cases}
x_1(t+1) = f_1(X(t), U(t)), \\
x_2(t+1) = f_2(X(t), U(t)), \\
\quad \vdots \\
x_n(t+1) = f_n(X(t), U(t)); \\
y_j(t) = h_j(X(t)), \quad j = 1, \cdots, p,
\end{cases} \quad (5)
$$

$$
\begin{cases}
\widehat{x}_1(t+1) = \widehat{f}_1(\widehat{X}(t)), \\
\widehat{x}_2(t+1) = \widehat{f}_2(\widehat{X}(t)), \\
\quad \vdots \\
\widehat{x}_{n_1}(t+1) = \widehat{f}_{n_1}(\widehat{X}(t)); \\
\widehat{y}_j(t) = \widehat{h}_j(\widehat{X}(t)), \quad j = 1, \cdots, p,
\end{cases} \quad (6)
$$

- ***The output regulation problem*** is to design the following state feedback control:

$$
\begin{cases}
u_1(t) = g_1(X(t), \widehat{X}(t)), \\
\quad \vdots \\
u_m(t) = g_m(X(t), \widehat{X}(t)),
\end{cases} \quad (7)
$$

under which there exists an integer $\tau > 0$ such that

$$Y(t; X_0, U) = \widehat{Y}(t; \widehat{X}_0)$$

holds for $\forall\ t \geq \tau$, $\forall\ X_0 \in \mathcal{D}^n$ and $\forall\ \widehat{X}_0 \in \mathcal{D}^{n_1}$.

[19] H. Li, L. Xie and Y. Wang, **Output regulation** of Boolean control networks, *IEEE Transactions on Automatic Control*, 2017, 62(6): 2993-2998.

Applications ⟶ Output regulation[19]

$$\begin{cases} x(t+1) = Lu(t)x(t), \\ y(t) = Hx(t), \end{cases} \quad (8)$$

$$\begin{cases} \widehat{x}(t+1) = \widehat{L}\widehat{x}(t), \\ \widehat{y}(t) = \widehat{H}\widehat{x}(t), \end{cases} \quad (9)$$

$$u(t) = Gx(t)\widehat{x}(t) \quad (10)$$

$$x(t)\widehat{x}(t) = R^t x(0)\widehat{x}(0) \quad (11)$$

$$R = LGW_{[2^n, 2^{n+n_1}]} M_{r,2^n} (I_{2^{n+n_1}} \otimes \widehat{L})(I_{2^n} \otimes M_{r,2^{n_1}})$$

$$y(t) - \widehat{y}(t) = Qx(t-1)\widehat{x}(t-1) = QR^{t-1}x(0)\widehat{x}(0)$$

$$Q = HLGW_{[2^n, 2^{n+n_1}]} M_{r,2^n} - \widehat{H}\widehat{L}E_d^n$$

Applications ⟶ Output regulation[19]

Theorem 1: The output regulation problem is solvable, if and only if there exist a logical matrix $G \in \mathcal{L}_{2^m \times 2^{n+n_1}}$ and an integer $1 \leq \tau \leq 2^{n+n_1}$ such that

$$QR^{\tau-1} = \mathbf{0}$$

Applications ⟶ Output regulation[19]

♣ *Control design*

$$\begin{cases} z(t+1) = \widetilde{L}u(t)z(t), \ \widetilde{L} = L(I_{2^{m+n}} \otimes \widehat{L}) \\ w(t) = \widetilde{H}z(t), \ \widetilde{H} = H(I_{2^n} \otimes \widehat{H}) \end{cases}$$

where $z(t) = x(t) \ltimes \widehat{x}(t)$, $w(t) = y(t) \ltimes \widehat{y}(t)$

- Define

$$\mathcal{O}_j = \left\{ \delta_{2^{n+n_1}}^i : Col_i(\widetilde{H}) = \delta_{2^p}^j \ltimes \delta_{2^p}^j \right\}. \quad \mathcal{O} = \bigcup_{j=1}^{2^p} \mathcal{O}_j.$$

⇒ The **output regulation control design** problem becomes how to stabilize the augmented system to a nonempty set $S \subseteq \mathcal{O}$.

Applications ⟶ Output regulation[19]

♣ *Control design*

- Step 1: Calculate $\mathcal{R}_k(S)$ and $\mathcal{R}_k^\circ(S), k = 1, \cdots, \tau$
- Step 2: For each $i = 1, 2, \cdots, 2^{n+n_1}$, calculate the unique integer $1 \leq k_i \leq \tau$ such that $\delta_{2^{n+n_1}}^i \in \mathcal{R}_{k_i}^\circ(S)$. Find an integer $1 \leq v_i \leq 2^m$ such that

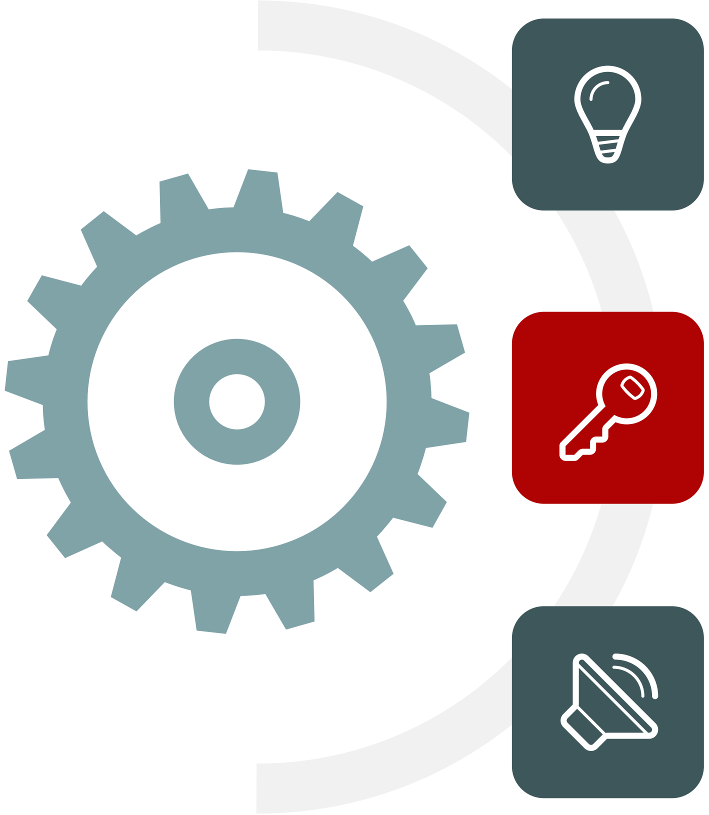
$$\begin{cases} \widetilde{L} \ltimes \delta_{2^m}^{v_i} \ltimes \delta_{2^{n+n_1}}^i \in S, \ k_i = 1; \\ \widetilde{L} \ltimes \delta_{2^m}^{v_i} \ltimes \delta_{2^{n+n_1}}^i \in \mathcal{R}_{k_i-1}(S), \ 2 \leq k_i \leq \tau. \end{cases}$$

- Step 3: The state feedback gain matrix can be designed as

$$G = \delta_{2^m}[v_1 \ v_2 \ \cdots \ v_{2^{n+n_1}}].$$

Part 3:
Controllability

Controllability



- Definition of controllability
- Current approaches
- Controllability under different controls

Definition

- Controllability[Akutsu et al. 2007, Cheng & Qi 2008]

Given $x_0, x_e \in D^n$. The Boolean control network (10) is said to be controllable from x_0 to x_e (by free Boolean sequence) at the s steps, if we can find control $u(t) \in D^m$, $t = 0, 1, \ldots, s-1$, such that the state $\ltimes_{i=1}^n A_i(0) = x_0$ and $\ltimes_{i=1}^n A_i(s) = x_e$, $i = 1, \ldots, n$.

- Fixed-time controllability[Laschov&Margaliot 2012]

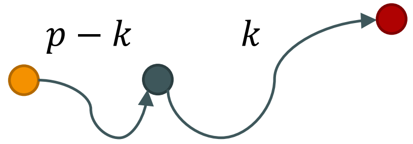
The BCN is k *fixed-time controllable* if for any $a, b \in (\{e^1_{2^n}, \ldots, e^{2^n}_{2^n}\}$ there exists a control $u \in \mathbb{U}^k$ that steers the BCN from $x(0) = a$ to $x(k) = b$,

Input networks
Free inputs
Sampled-data inputs

$$\begin{cases} A_1(t+1) = f_1(A_1(t), \ldots, A_n(t), u_1(t), \ldots, u_m(t)) \\ \vdots \\ A_n(t+1) = f_n(A_1(t), \ldots, A_n(t), u_1(t), \ldots, u_m(t)), \end{cases} \quad (10)$$

$$x(t+1) = Lu(t)x(t) := L_u(t)x(t), x \in D^n$$

Corollary 3. *If a BCN is k fixed-time controllable, then it is p fixed-time controllable for any $p \geq k$.*



Cheng D, Qi H. Controllability and observability of Boolean control networks[J]. Automatica, 2009, 45(7): 1659-1667.
Laschov D, Margaliot M. Controllability of Boolean control networks via the Perron–Frobenius theory[J]. Automatica, 2012, 48(6): 1218-1223.

Current approaches

input-state transfer matrix[Cheng & Qi 2008]

Input networks

(1) The controls are logical variables satisfying certain logical rule, called the input network, as

$$\begin{cases} u_1(t+1) = g_1(u_1(t), \dots, u_m(t)), \\ \vdots \\ u_m(t+1) = g_m(u_1(t), \dots, u_m(t)). \end{cases} \tag{12}$$

Algebraic form

$$u(t+1) = Gu(t), \quad u \in D^m \tag{13}$$

Theorem 9. *Consider system (10) with control (12), equivalently, (13), where G is fixed. x_d is s step reachable from x_0, iff*

$$x_d \in \mathrm{Col}\left\{\Theta^G(s,0)W_{[2^n,2^m]}x_0\right\}, \tag{24}$$

where and hereafter Col *is the column set.*

Definition 8. For a fixed G the input-state transfer matrix $\Theta^G(t,0)$ is defined as follows: for any $u_0 \in D^m$ and any $x(0) = x_0 \in D^n$, we have $x(t) = \Theta^G(t,0)u_0x_0$, $t > 0$.

It is obvious that $\Theta^G(t,0)$ depends on G. In the following we will find the input-state transfer matrix. Since

$x_1 = Lu_0x_0,$

we have $\Theta^G(1,0) = L$. Next, we calculate $x_2 = x(2)$, which is

$x_2 = Lu_1x_1 = LGu_0Lu_0x_0 = LG(I_{2^m} \otimes L)\Phi_m u_0 x_0,$

where Φ_m is defined as

$$\Phi_m = \ltimes_{i=1}^{m} I_{2^{i-1}} \otimes \left[\left(I_2 \otimes W_{[2,2^{m-i}]}\right) M_r\right];$$

$M_r = \delta_4[1,4]$ is defined in Cheng (2007), and $\otimes$ is the Kronecker product. Then we have $\Theta^G(2,0) = LG(I_{2^m} \otimes L)\Phi_m$. Using mathematical induction, it is easy to prove that

$$\begin{aligned}\Theta^G(t,0) = {} & LG^{t-1}(I_{2^m} \otimes LG^{t-2})(I_{2^{2m}} \otimes LG^{t-3}) \cdots \\ & (I_{2^{(t-1)m}} \otimes L)(I_{2^{(t-2)m}} \otimes \Phi_m) \\ & (I_{2^{(t-3)m}} \otimes \Phi_m) \cdots (I_{2^m} \otimes \Phi_m)\Phi_m. \end{aligned} \tag{23}$$

Cheng D, Qi H. Controllability and observability of Boolean control networks[J]. Automatica, 2009, 45(7): 1659-1667.
Cheng D, Semi-tensor product of matrices and its applications-A survey[J]. In: Proc. ICCM 2007, 3 : 641–668.

Current approaches

□ 1: input-state transfer matrix[Cheng & Qi 2008]

Free sequences

Define $\tilde{L} = LW_{[2^n,2^m]}$, then

$$x(t+1) = \tilde{L}x(t)u(t). \tag{32}$$

Using it repetitively yields

$$x(s) = \tilde{L}^s x(0)u(0)u(1)\cdots u(s-1). \tag{33}$$

Definition 23. System (10) is said to be globally reachable from x_0 (by controls of free length Boolean sequence) if $R(x_0) = D^n$. System (10) is called globally controllable (by controls of free length Boolean sequence) if $R(x_0) = D^n, \forall x_0 \in D^n$.

Theorem 18. *x_e is reachable from x_0, at the sth time step by controls of Boolean sequences of length s, iff*

$$x_s \in \mathrm{Col}\{\tilde{L}^s x_0\}. \tag{34}$$

Corollary 20. *x_d is reachable from x_0, iff*

$$x_d \in \mathrm{Col}\left\{\bigcup_{i=1}^{\infty} \tilde{L}^i x_0\right\}. \tag{37}$$

Proposition 21. (1) *The reachable set, $R(x_0)$, is a subset of* $\mathrm{Col}\{\tilde{L}\}$;
(2) *Assume that k^* is the smallest $k > 0$, such that*

$$\mathrm{Col}\{\tilde{L}^{k+1}x_0\} \subset \mathrm{Col}\left\{\tilde{L}^s x_0 \,\middle|\, s = 1, 2, \ldots, k\right\},$$

then the reachable set

$$R(x_0) = \mathrm{Col}\left\{\bigcup_{i=1}^{k^*} \tilde{L}^i x_0\right\}. \tag{38}$$

Cheng D, Qi H. Controllability and observability of Boolean control networks[J]. Automatica, 2009, 45(7): 1659-1667.

Current approaches

☐ 2: input-state incidence matrix [zhao et al., scl, 2011]

$$\begin{cases} x(t+1) = Lu(t)x(t) \\ y(t) = Hx(t), \end{cases} \quad (4)$$

follows that the input-state incidence matrix of system (4) is

$$\mathcal{J} := \mathcal{J}|_{(4)} = \left.\begin{bmatrix} L \\ L \\ \vdots \\ L \end{bmatrix}\right\} 2^m \in \mathcal{B}_{2^{m+n} \times 2^{m+n}}, \quad (10)$$

where the first block corresponds to $u(t+1) = \delta_{2^m}^1$, the second block corresponds to $u(t+1) = \delta_{2^m}^2$, and so on.

Theorem 3.1. *Consider system (4). Assume that the (i, j)-th element of the s-th power of its input-state incidence matrix, $\mathcal{J}_{ij}^s = c$. Then there are c paths from point P_j reach P_i at s-th step with proper controls.*

Proof. We prove it by mathematical induction. When $s = 1$ the conclusion follows from the definition of input-state incidence matrix.

Now assume $\mathcal{J}_{ij}^s$ is the number of the paths from P_j to P_i at the s-th step. Since a path from P_j to P_i at the $(s+1)$-th step can always be considered as a path from P_j to P_k at the s-th step and then from P_k to P_i at one step. It can be calculated as

$$c = \sum_{k=1}^{2^{m+n}} \mathcal{J}_{ik} \mathcal{J}_{kj}^s,$$

which is exactly $\mathcal{J}_{ij}^{s+1}$. □

Corollary 2.4. *Consider system (4). Its input-state incidence matrix is*

$$\mathcal{J} = \mathbf{1}_{2^m} \ltimes \mathcal{J}_0, \quad \text{where } \mathcal{J}_0 = L. \quad (11)$$

Moreover, the basic block of $\mathcal{J}^s$ is

$$\mathcal{J}_0^s = L \ltimes (\mathbf{1}_{2^m} \ltimes L)^{s-1}. \quad (12)$$

Proposition 2.5.

$$\mathcal{J}_0^{s+1} = M^s L,$$

where

$$M = \sum_{i=1}^{2^m} \mathrm{Blk}_i(L).$$

Zhao Y, et al., Input-state incidence matrix of Boolean control networks and its applications[J]. Systems & Control Letters, 2010, 59(12): 767-774.

Current approaches

☐ 2: input-state incidence matrix [zhao et al., scl, 2011]

Theorem 3.3. *Consider system (4) with its input-state incidence matrix $\mathcal{J}$.*

1. *$x(s) = \delta_{2^n}^{\alpha}$ is reachable from $x(0) = \delta_{2^n}^{j}$ at s-th step, iff*

$$\sum_{i=1}^{2^m} (\mathrm{Blk}_i(\mathcal{J}_0^s))_{\alpha j} = (M^s)_{\alpha j} > 0. \quad (15)$$

2. *$x = \delta_{2^n}^{\alpha}$ is reachable from $x(0) = \delta_{2^n}^{j}$, iff*

$$\sum_{s=1}^{2^{m+n}} \sum_{i=1}^{2^m} (\mathrm{Blk}_i(\mathcal{J}_0^s))_{\alpha j} = \sum_{s=1}^{2^{m+n}} (M^s)_{\alpha j} > 0. \quad (16)$$

3. *The system is controllable at $x(0) = \delta_{2^n}^{j}$, iff*

$$\sum_{s=1}^{2^{m+n}} \sum_{i=1}^{2^m} \mathrm{Col}_j[\mathrm{Blk}_i(\mathcal{J}_0^s)] = \sum_{s=1}^{2^{m+n}} \mathrm{Col}_j(M^s) > 0. \quad (17)$$

4. *The system is controllable, iff*

$$\sum_{s=1}^{2^{m+n}} \sum_{i=1}^{2^m} \mathrm{Blk}_i(\mathcal{J}_0^s) = \sum_{s=1}^{2^{m+n}} M^s > 0. \quad (18)$$

Note that let $A \in \mathcal{M}_{m\times n}$. The inequality $A > 0$ means all the entries of A are positive, i.e., $a_{i,j} > 0, \ \forall i, j$.

Assume $x_0 = \delta_{2^n}^{j}$ and $x_d = \delta_{2^n}^{i}$. We give the following algorithm.

Algorithm 4.1. Assume the (i, j)-th element of the controllability matrix, $c_{i,j} > 0$.

- Step 1: Find the smallest s, such that in the block decomposed form

$$\mathcal{J}_0^s = \left[\mathrm{Blk}_1(\mathcal{J}_0^s) \quad \mathrm{Blk}_2(\mathcal{J}_0^s) \quad \cdots \quad \mathrm{Blk}_{2^m}(\mathcal{J}_0^s)\right], \quad (22)$$

  (where $\mathrm{Blk}_i(\mathcal{J}_0^s) \in \mathcal{M}_{2^n\times 2^n}$) there exists a block, say, $\mathrm{Blk}_\alpha\left(\mathcal{J}_0^{(s)}\right)$, which has its (i, j)-element

$$[\mathrm{Blk}_\alpha(\mathcal{J}_0^{(s)})]_{ij} > 0. \quad (23)$$

  Set $u(0) = \delta_{2^m}^{\alpha}$ and $x(s) = \delta_{2^n}^{i}$. If $s = 1$, stop. Else, go to the next step.
- Step 2: Find k, β, such that

$$[\mathrm{Blk}_\beta(\mathcal{J}_0)]_{ik} > 0; \qquad [\mathrm{Blk}_\alpha(\mathcal{J}_0^{s-1})]_{kj} > 0.$$

  Set $u(s-1) = \delta_{2^m}^{\beta}$ and $x(s-1) = \delta_{2^n}^{k}$.
- Step 3: If $s - 1 = 1$, stop. Else, set $s = s - 1$, and $i = k$ (that is, replace s by $s - 1$ and replace i by k), and go back to Step 2.

Proposition 4.2. *As long as $x_d \in R(x_0)$ the control sequence $\{u(0), u(1), \ldots, u(s-1)\}$ generated by Algorithm 4.1 can drive the trajectory from x_0 to x_d. Moreover, the corresponding trajectory is $\{x(0) = x_0, x(1), \ldots, x(s) = x_d\}$, which is also produced from the algorithm.*

Zhao Y, et al., Input-state incidence matrix of Boolean control networks and its applications[J]. Systems & Control Letters, 2010, 59(12): 767-774.

Current approaches

☐ 3: Perron–Frobenius theory [Laschov and Margaliot, Auto, 2012]

$$x(k+1) = L \ltimes u(k) \ltimes x(k). \quad (3)$$

For $a, b \in \{e_{2^n}^1, \dots, e_{2^n}^{2^n}\}$ and a set of undesirable states C, let $l(k; a, b, C)$ denote the number of different control sequences that steer the BCN (3) from $x(0) = a$ to $x(k) = b$, while avoiding C (i.e. $x(i) \notin C$ for $i = 0, 1, \dots, k$).

Let $|C|$ denote the cardinality of C. Let 1_r denote the column vector of length r with all entries equal to 1, and let $Q = L \ltimes 1_{2^m}$.

Theorem 2. *Suppose that the states in C are $e_{2^n}^{i_1}, \dots, e_{2^n}^{i_z}$ where $z = |C|$. Let Q_C be the matrix obtained from Q by substituting zeros in the rows and columns with indexes $i_1, \dots, i_z$. Then*

$$l(k; a, b, C) = b^T (Q_C)^k a. \quad (7)$$

Proof. By induction on k. Consider the case $k = 1$. Let $s = l(1; a, b, C)$. If $a \in C$ or $b \in C$, then clearly $s = 0$. Since in Q_C either the row corresponding to b or the column corresponding to a is zero, $b^T Q_C a = 0$. So in this case, $l(1; a, b, C) = b^T Q_C a$. Now suppose that $a \notin C$ and $b \notin C$. Let $w^1, \dots, w^s$ be the different control sequences steering (3) from $x(0) = a$ to $x(1) = b$, i.e.

$$b = L \ltimes w^i(0) \ltimes a, \quad i \in \{1, \dots, s\}. \quad (8)$$

Since each control value is a column of I_{2^m}, there exist $t = 2^m - s$ different control sequences $v^j \in \mathbb{U}^1$ such that

$$b \neq L \ltimes v^j(0) \ltimes a, \quad j \in \{1, \dots, t\}. \quad (9)$$

Note that the term on the right-hand side of this inequality must be a column of I_{2^n}. Therefore, multiplying (8) and (9) from the left by b^T yields

$$1 = b^T L \ltimes w^i(0) \ltimes a, \quad i \in \{1, \dots, s\},$$
$$0 = b^T L \ltimes v^j(0) \ltimes a, \quad j \in \{1, \dots, t\}.$$

Since each of the control values is a different column of I_{2^m}, summing up this set of $s + t = 2^m$ equations yields

$$s = b^T \ltimes L \ltimes 1_{2^m} \ltimes a = b^T Q a.$$

We conclude that when $a \notin C$ and $b \notin C$, $l(1; a, b, C) = b^T Q a$. Since in this case $b^T Q a = b^T Q_C a$, $l(1; a, b, C) = b^T Q_C a$. This proves (7) for $k = 1$. For the induction step, consider

$$\begin{aligned} b^T (Q_C)^{k+1} a &= (e_{2^n}^j)^T (Q_C)^{k+1} e_{2^n}^i \\ &= ((Q_C)^k Q_C)_{ji} \\ &= \sum_{p=1}^{2^n} ((Q_C)^k)_{jp} (Q_C)_{pi} \\ &= \sum_{p=1}^{2^n} (e_{2^n}^j)^T ((Q_C)^k)\, e_{2^n}^p (e_{2^n}^p)^T Q_C\, e_{2^n}^i . \end{aligned}$$

Laschov D, Margaliot M. Controllability of Boolean control networks via the Perron–Frobenius theory[J]. Automatica, 2012, 48(6): 1218-1223.

Current approaches

3: Perron–Frobenius theory [Laschov and Margaliot, Auto, 2012]

Definition 3. A matrix $M \in \mathbb{R}^{n\times n}$, with $n \geq 2$, is said to be *reducible* if there exists a permutation matrix $P \in \{0, 1\}^{n\times n}$, and an integer r with $1 \leq r \leq n-1$ such that

$$P^T MP = \begin{bmatrix} B & C \\ 0 & D \end{bmatrix}, \tag{11}$$

where $B \in \mathbb{R}^{r\times r}$, $D \in \mathbb{R}^{(n-r)\times(n-r)}$, $C \in \mathbb{R}^{r\times(n-r)}$ and $0 \in \mathbb{R}^{(n-r)\times r}$ is a zero matrix. A matrix is said to be *irreducible* if it is not reducible.

Theorem 3 (Berman & Plemmons 1987, Ch. 2). *Suppose that $A \in \mathbb{R}^{n\times n}$ is nonnegative. Then A is irreducible if and only if for any $i, j \in \{1, \ldots, n\}$ there exists an integer $k \geq 1$ such that $(A^k)_{ij} > 0$.*

controllability

Theorem 4. *The BCN (3) is controllable if and only if $\tilde{Q}_C$ is irreducible.*

Theorem 6 (Horn & Johnson 1985, Ch. 8). *A nonnegative matrix $A \in \mathbb{R}^{n\times n}$ is called* primitive *if there exists an integer $j \geq 1$ such that $A^j > 0$. In this case, the smallest such j is called the* index of primitivity *of A, denoted $\gamma(A)$. If A is primitive, then $\gamma(A) \leq n^2 - 2n + 2$.*

Fixed-time controllability

Theorem 5. *The BCN (3) is k fixed-time controllable if and only if $(\tilde{Q}_C)^k > 0$.*

Corollary 2. *If the matrix $\tilde{Q}_C$ is primitive, then*

$$\gamma(\tilde{Q}_C) \leq q^2 - 2q + 2 \tag{12}$$

and the BCN (3) is $\gamma(\tilde{Q}_C)$ fixed-time controllable. If $\tilde{Q}_C$ is not primitive, then the BCN is not k fixed-time controllable for any k.

Laschov D, Margaliot M. Controllability of Boolean control networks via the Perron–Frobenius theory[J]. Automatica, 2012, 48(6): 1218-1223.
Berman A, Plemmons R. J. Nonnegative matrices in the mathematical sciences, SIAM, 1987.
Horn R. A, Johnson C. R. Matrix analysis, Cambridge University Press,1985.

Current approaches

- complexity reduction of controllability matrix

$O(2^{2n})$ — Zhu Q, Liu Y, Lu, JQ, et al. Further results on the controllability of Boolean control networks. IEEE TAC, 2018. ← Tarjan Algorithm& Deep first search

$O(2^{3n})$ — Liang J, Chen H, et al. An improved criterion for controllability of Boolean control networks. IEEE TAC, 2017. ← Warshall algorithm

$O(2^{4n})$ — Zhao Y, et al. Input-state incidence matrix of Boolean control networks and its applications. SCL, 2010.

Controllability under different controls

Mixed control strategy[Cheng et al. IEEE CSL, 2018]

The system considered in this section is

$$\begin{cases} x_1(t+1) = f_1(x_1(t), \ldots, x_n(t); u_1(t), \ldots, u_r(t), \\ \qquad\qquad v_1(t), \ldots, v_q(t)), \\ x_2(t+1) = f_2(x_1(t), \ldots, x_n(t); u_1(t), \ldots, u_r(t), \\ \qquad\qquad v_1(t), \ldots, v_q(t)), \\ \quad\vdots \\ x_n(t+1) = f_n(x_1(t), \ldots, x_n(t); u_1(t), \ldots, u_r(t), \\ \qquad\qquad v_1(t), \ldots, v_q(t)), \end{cases} \tag{17}$$

where $v_j(t)$, $j = 1, \ldots, q$, are free inputs and $u_i(t)$, $i = 1, \ldots, r$, are networked inputs, generated by the following input network

$$\begin{cases} u_1(t+1) = g_1(u_1(t), \ldots, u_r(t)) \\ u_2(t+1) = g_2(u_1(t), \ldots, u_r(t)) \\ \quad\vdots \\ u_r(t+1) = g_r(u_1(t), \ldots, u_r(t)), \end{cases} \tag{18}$$

where $f_i : \mathcal{D}^{n+r+q} \to \mathcal{D}$, $i = 1, \ldots, n$ and $g_j : \mathcal{D}^r \to \mathcal{D}$, $j = 1, \ldots, r$ are Boolean functions.

$$\begin{cases} u(t+1) = Gu(t) \\ x(t+1) = Lv(t)u(t)x(t). \end{cases} \tag{21}$$

Consider $w(t) = u(t)x(t)$ as new state variables, then (21) becomes

$$w(t+1) = \Phi v(t)w(t), \tag{22}$$

where $\Phi = [G(\mathbf{1}_{2^q}^{\mathrm{T}} \otimes I_{2^r} \otimes \mathbf{1}_{2^n}^{\mathrm{T}})] * L$.

Theorem 6: System (17) with mixed controls is controllable if and only if the combined system (22) is set controllable with

$$P^d = P^0 = \{s_i \mid i = 1, \ldots, 2^n\}, \tag{23}$$

where $s_i = \{w \mid \mathbf{1}_{2^r}^{\mathrm{T}} w = \delta_{2^n}^i\}$.

Cheng D, Li C, Zhang X, et al. Controllability of Boolean networks via mixed controls[J]. IEEE Control Systems Letters, 2018, 2(2): 254-259.

Controllability under different controls

- Pinning control [Lu J.Q. et al. IEEE TAC, 2016]

$$\begin{cases} x_1(t+1) = f_1(x_1(t), \ldots, x_n(t), u_1(t)) \\ \cdots \\ x_r(t+1) = f_r(x_1(t), \ldots, x_n(t), u_r(t)) \\ x_{r+1}(t+1) = f_{r+1}(x_1(t), \ldots, x_n(t)) \\ \cdots \\ x_n(t+1) = f_n(x_1(t), \ldots, x_n(t)) \end{cases} \quad (2)$$

Control inputs are only injected on a fraction of nodes!

$$\begin{cases} x^1(t+1) = L_1 x(t) \ltimes_{i=1}^r u_i(t) \\ x^2(t+1) = L_2 x(t) \end{cases}$$

$$\begin{cases} L_1 = (\otimes_{i=1}^r F_i) \ltimes_{i=1}^{r-1} [(I_{2^n} \otimes W_{[2^n, 2^i]}) \Phi_n]. \\ L_2 = F_{r+1} * F_{r+2} * \cdots * F_n \in \mathcal{L}_{2^{n-r} \times 2^n}. \end{cases}$$

$$\begin{aligned} x(t+1) &= L_1 x(t) u(t) L_2 x(t) \\ &\triangleq L_1 x(t) u(t) \tilde{L_2} x(t) u(t) \\ &\triangleq M x(t) u(t) \end{aligned}$$

where $\tilde{L_2} = L_2(I_{2^n} \otimes \mathbf{1}_{2^r}^T)$ and $M = L_1 * \tilde{L_2}$.

Let $\tilde{M} = M W_{[2^r, 2^n]}$ and $\bar{M} = \tilde{M} \ltimes \mathbf{1}_{2^r}^T$.

Theorem 2: For system (2), consider two given states $a = \delta_{2^n}^i$, $b = \delta_{2^n}^j$ and a given time step k. Let $N(k; a, b)$ denote the number of different control sequences that steer the BCNs (2) from $x(0) = a$ to $x(k) = b$ with k steps. Then

$$N(k; a, b) = b^T (\bar{M})^k a. \quad (10)$$

Theorem 4: Consider system (2).

- $x(k) = \delta_{2^n}^j$ is reachable from $x(0) = \delta_{2^n}^i$ at the k-th step if and only if $(\bar{M}^k)_{ji} > 0$.
- $\delta_{2^n}^j$ is reachable from $x(0) = \delta_{2^n}^i$ if and only if there exists an integer k such that $\sum_{s=1}^k (\bar{M}^s)_{ji} > 0$.

Theorem 5: The BCNs with pinning controllers (2) is controllable if and only if $\bar{M}$ is irreducible.

Lu J, Zhong J, Huang C, et al. On pinning controllability of Boolean control networks. IEEE TAC, 2016, 61(6): 1658-1663.

Controllability under different controls

- Sampled-data control [Zhu Q.X. et al. IEEE TCNS, 2019]

$$\begin{cases} x_i(t+1) = f_i(u_1(t), \ldots, u_m(t), x_1(t), \ldots, x_n(t)), \\ \qquad i = 1, 2, \ldots, n \\ y_j(t) = h_j(x_1(t), \ldots, x_n(t)), j = 1, 2, \ldots, p \end{cases} \quad (1)$$

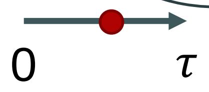
$$u_k(t) = u_k(t_l), t_l \le t < t_{l+1}, k = 1, 2, \ldots, m$$

$\boldsymbol{t_l = l\tau, l \in N}$ are sampling instants, and $\boldsymbol{t_{l+1} - t_l = \tau}$ is constant sampling period

We denote the sampled-data control sequence with the constant sampling period τ just by

$$\pi^t = \{u(0), u(1), \ldots, u(\tau), \ldots, u(2\tau), \ldots, u(t-1)\}, t > 0.$$

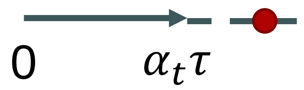
In addition, the set of all sampled-data control sequences $\pi^t = \{u(0), u(1), \ldots, u(\tau), \ldots, u(2\tau), \ldots, u(t-1)\}$ is denoted by $\prod^t$, $t > 0$.



$$x(t_0 + t) = {}^tLu(t_0)x(t_0)$$

where ${}^tL := [(L_1)^t, (L_2)^t, \ldots, (L_M)^t]$. Let

$$ {}^t\mathcal{M} := \sum_{i=1}^{M} {}_{\mathscr{B}}(L_i)^t \quad (6)$$



$${}^t\mathcal{M} = {}^{\beta_t}\mathcal{M} \times_{\mathscr{B}} ({}^\tau\mathcal{M})^{(\alpha_t)}$$

$$\beta_t = t \mod \tau, \ \alpha_t = \frac{t - \beta_t}{\tau}.$$

$$\lfloor {}^s\mathcal{M} \rfloor = \begin{cases} \sum_{k=1}^{s} {}_{\mathscr{B}} {}^k\mathcal{M}, & s > 0, \\ I, & s = 0 \end{cases}$$

$$\lfloor ({}^\tau\mathcal{M})^{(s)} \rfloor = \begin{cases} \sum_{k=1}^{s} {}_{\mathscr{B}} ({}^\tau\mathcal{M})^{(k)}, & s > 0, \\ I, & s = 0. \end{cases}$$

Theorem 1: Consider SDBCN (1), $x(0) = x_0$.

1) $x_e = \delta_N^i$ is reachable from $x_0 = \delta_N^j$ at the sth time step if and only if $({}^s\mathcal{M})_{ij} = 1$.
2) $x_e = \delta_N^i$ is reachable from $x_0 = \delta_N^j$ if and only if $\lfloor {}^{N\tau}\mathcal{M} \rfloor_{ij} = 1$.
3) SDBCN (1) is said to be controllable at $x_0 = \delta_N^i$, if and only if $Col_i\left(\lfloor {}^{N\tau}\mathcal{M} \rfloor\right) > 0$.
4) SDBCN (1) is said to be controllable if and only if $\lfloor {}^{N\tau}\mathcal{M} \rfloor > 0$.

Zhu Q, Liu Y, Lu J, et al. Controllability and observability of Boolean control networks via sampled-data control. IEEE CNS, 2018, 6(4): 1291-1301.

Part 4:
Several special kinds of BNs

Several special kinds of BNs

Switched Boolean network

Conjunctive Boolean Networks

Boolean networks with time delays

Large-scale Boolean Networks

Switched Boolean networks (SBNs)

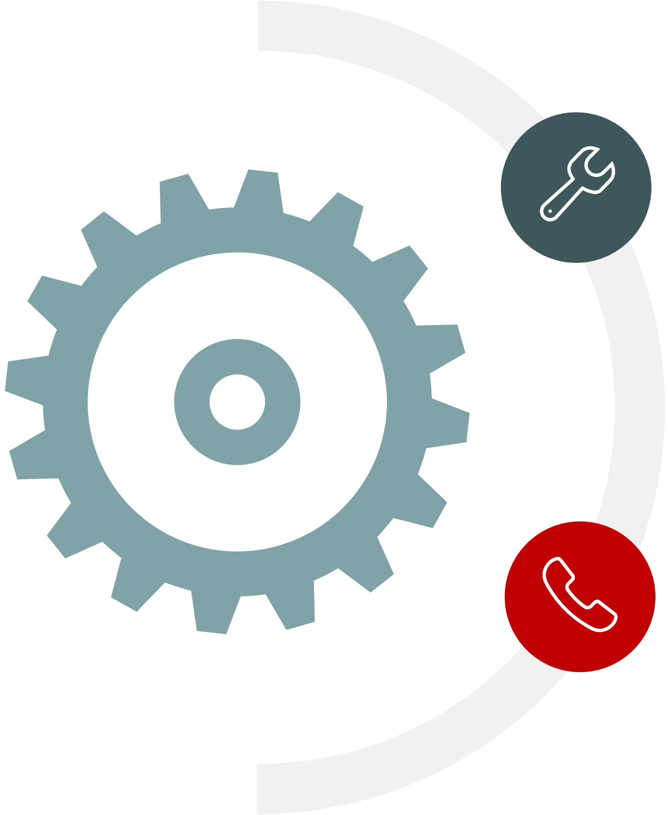
While typical Boolean networks are described by purely discrete dynamics, the dynamics of gene regulatory networks in practice are often governed by different switching models.

- A practical example is the cell's growth and division in a eukaryotic cell, which are usually described as a sequence of four processes triggered by a set of conditions or events. In this case, the cell differentiation can be viewed as a switched system.

- Another typical example is the genetic switch in the bacteriophage λ, which contains two distinct models, that is, lysis and lysogeny.

Thus, it is necessary for us to investigate switched Boolean control networks (BCNs).

Switched Boolean network



Controllability

Stability and stabilization

On controllability of switched Boolean networks

Li H and Wang Y [1] first proposed the necessity of studying switched Boolean networks and studied the controllability of switched Boolean control networks.

Li H and Wang Y [3] first studied the controllability of switched Boolean control networks with state and input constraints.

2012年

2014年

2015年

2019年

Chen H and Sun J [2] first proposed state-dependent switched Boolean control network and studied the output controllability of state-dependent switched Boolean control networks.

Zhang Q , et al., [4] first studied set controllability for switched Boolean control networks.

[1] Li H and Wang Y. On reachability and controllability of switched Boolean control networks. Automatica, 2012, 48: 2917-2922.

[2] Chen H and Sun J. Output controllability and optimal output control of state-dependent switched Boolean control networks. Automatica, 2014, 50(7): 1929-1934.

[3] Li H and Wang Y. Controllability analysis and control design for switched Boolean networks with state and input constraints. SIAM Journal on Control and Optimization, 2015, 53(5): 2955-2979.

[4] Zhang Q, Feng J, Pan J, et al. Set controllability for switched Boolean control networks. Neurocomputing, 2019, 359: 476-482.

Representative results [1]

Model (**free-form switching signals**)

Consider a switched BCN with n nodes, m control inputs and w sub-networks as

$$\begin{cases} x_1(t+1) = f_1^{\sigma(t)}(x_1(t), \ldots, x_n(t), u_1(t), \ldots, u_m(t)), \\ x_2(t+1) = f_2^{\sigma(t)}(x_1(t), \ldots, x_n(t), u_1(t), \ldots, u_m(t)), \\ \vdots \\ x_n(t+1) = f_n^{\sigma(t)}(x_1(t), \ldots, x_n(t), u_1(t), \ldots, u_m(t)), \end{cases} \tag{3}$$

where $\sigma : N \to W = \{1, 2, \ldots, w\}$ is the switching signal. Given a finite-time switching signal $\sigma : \{0, 1, \ldots, l\} \to W$ with l a given positive integer, set $\sigma(k) = i_k, k = 0, 1, \ldots, l$. Then, we obtain the following switching sequence: $\pi := \{(0, i_0), (1, i_1), \ldots, (l, i_l)\}$.

[1] Li H and Wang Y. On reachability and controllability of switched Boolean control networks. Automatica, 2012, 48: 2917-2922.

Representative results [1]

Main results
(switching-input-state incidence matrix)

Using the vector form of logical variables x_i and u_i, and setting $x = \ltimes_{i=1}^{n} x_i$ and $u = \ltimes_{i=1}^{m} u_i$, the SBN can be expressed as

$$x(t+1) = L_{\sigma(t)} u(t) x(t), \qquad (6)$$

Proposition 1. *Consider the switched BCN (3) with its algebraic form (6). The switching-input-state incidence matrix of the system (3) can be given as*

$$\mathcal{S} = \left.\begin{bmatrix} \widehat{L} \\ \widehat{L} \\ \vdots \\ \widehat{L} \end{bmatrix}\right\} w2^m \in \mathcal{B}_{w2^{m+n} \times w2^{m+n}}, \qquad (9)$$

where $\widehat{L} = [L_1 \ldots L_w] \in \mathcal{L}_{2^n \times w2^{m+n}}$.

[1] Li H and Wang Y. On reachability and controllability of switched Boolean control networks. Automatica, 2012, 48: 2917-2922.

Representative results [1]

Main results
(switching-input-state incidence matrix)

Proposition 2. $\mathcal{S}$ given in (9) is a row-periodic matrix with period 2^n, and $\mathcal{S} = \mathbf{1}_{w2^m}\widehat{L}$. In addition, $\mathcal{S}^l$, $l \in \mathbb{Z}_+$ is also a row-periodic matrix with period 2^n, and its basic block is

$$\mathcal{S}_0^l = \widehat{M}^{l-1}\widehat{L} \in \mathbb{R}^{2^n \times w2^{m+n}}, \tag{11}$$

where $\widehat{M} = \sum_{i=1}^{w2^m} Blk_i(\widehat{L})$, and $\widehat{L} = [L_1 \ \cdots \ L_w]$.

Theorem 2. *Consider the switched BCN (3) with its switching-input-state incidence matrix $\mathcal{S}$ given in (9). Then,*

(1) $x(l) = \delta_{2^n}^{\alpha}$ *is reachable from* $x(0) = \delta_{2^n}^{j}$ *at the l-th step, if and only if*

$$\sum_{i=1}^{w2^m}\left(Blk_i(\mathcal{S}_0^l)\right)_{\alpha j} = \left(\widehat{M}^l\right)_{\alpha j} > 0, \tag{13}$$

where $\mathcal{S}_0^l = \widehat{M}^{l-1}\widehat{L}$, $\widehat{M} = \sum_{i=1}^{w2^m} Blk_i(\widehat{L})$ *and* $\widehat{L} = [L_1 \ \cdots \ L_w]$;

(2) $x = \delta_{2^n}^{\alpha}$ *is reachable from* $x(0) = \delta_{2^n}^{j}$, *if and only if*

$$\sum_{l=1}^{w2^{m+n}}\left(\widehat{M}^l\right)_{\alpha j} > 0; \tag{14}$$

(3) *the system is controllable at* $x(0) = \delta_{2^n}^{j}$, *if and only if*

$$\sum_{l=1}^{w2^{m+n}} Col_j\left(\widehat{M}^l\right) > 0; \tag{15}$$

(4) *the system is controllable, if and only if*

$$\sum_{l=1}^{w2^{m+n}} \widehat{M}^l > 0. \tag{16}$$

[1] Li H and Wang Y. On reachability and controllability of switched Boolean control networks. Automatica, 2012, 48: 2917-2922.

Representative results [2]

Model (**state-feedback switching signals**)

Consider a state-dependent switched Boolean control network

$$\begin{cases} x_1(t+1) = f_1^{\sigma(t)}(x_1(t), \ldots, x_n(t), u_1(t), \ldots, u_m(t)), \\ x_2(t+1) = f_2^{\sigma(t)}(x_1(t), \ldots, x_n(t), u_1(t), \ldots, u_m(t)), \\ \vdots \\ x_n(t+1) = f_n^{\sigma(t)}(x_1(t), \ldots, x_n(t), u_1(t), \ldots, u_m(t)), \end{cases}$$
$$y_j(t) = h_j(x_1(t), \ldots, x_n(t)), \quad j = 1, 2, \ldots, p \ (p \leq n), \qquad (2)$$

$$\begin{cases} x(t+1) = \hat{L}x(t)u(t), \\ y(t) = Hx(t), \end{cases} \qquad (5)$$

$$\begin{cases} u_1(t+1) = g_1(u_1(t), u_2(t), \ldots, u_m(t)), \\ u_2(t+1) = g_2(u_1(t), u_2(t), \ldots, u_m(t)), \\ \vdots \\ u_m(t+1) = g_m(u_1(t), u_2(t), \ldots, u_m(t)), \end{cases} \qquad (6)$$

where $x_i(t) \in D, i = 1, 2, \ldots, n$ are Boolean variables, $u_i(t) \in D, i = 1, 2, \ldots, m$, are control inputs. $\sigma : N \to W = \{1, 2, \ldots, w\}$ is the state-dependent switching signal. Using the properties of the semi-tensor product, we have

$$\sigma(t) = Qx(t),$$

where $Q \in \mathcal{L}_{w \times 2^n}$.

[2] Chen H and Sun J. Output controllability and optimal output control of state-dependent switched Boolean control networks. Automatica, 2014, 50(7): 1929-1934.

Representative results [2]

Main results

Consider the state-dependent switched Boolean networks with controls inputs $u(t+1) = Gu(t)$

Definition
The system is said to be **output-controllable** at the initial state $x(0)$, if for any destination output state $y_f \in \Delta_{2^p}$, one can always find T and a control sequence $u(0), u(1), \dots, u(T-1)$, under which $y(T) = y_f$. The system is said to be output-controllable, if the system is output-controllable at any $x(0) \in \Delta_{2^n}$.

Theorem 3.1. *Considering the state-dependent switched Boolean networks (5) with controls (6), we have the following results:*

(i) *$y(s) = \delta_{2p}^i$ is s-output-reachable from the initial state $x(0) = \delta_{2n}^j$, if and only if $(V_s)_{i,j} > 0$.*
(ii) *The system (5) is s-output-controllable at the initial state $x(0) = \delta_{2n}^j$, if and only if $Col_j(V_s) > 0$.*
(iii) *The system (5) is s-output-controllable, if and only if $V_s > 0$.*
(iv) *The system (5) is output-controllable at the initial state $x(0) = \delta_{2n}^j$, if and only if there exists an integer N, such that $Col_j(\sum_{p=1}^N V_p) > 0$.*
(v) *The system (5) is output-controllable, if and only if there exists an integer N, such that $(\sum_{p=1}^N V_p) > 0$.*

[2] Chen H and Sun J. Output controllability and optimal output control of state-dependent switched Boolean control networks. Automatica, 2014, 50(7): 1929-1934.

Representative results [2]

Main results

Consider the state-dependent switched Boolean networks (5) with a free control sequence (Set $u(t) = u_1(t)u_2(t) \cdots um(t)$. Then $u(t) \in Dm$, $t = 1, 2, \ldots$, is a designed control sequence)

Theorem 3.2. *Considering the state-dependent switched Boolean networks* (5) *with a free control sequence, we have the following results:*

(i) $y(s) = \delta_{2p}^{i}$ *is s-output-reachable from the initial state* $x(0) = \delta_{2^n}^{j}$, *if and only if* $(\bar{V}_s)_{i,j} > 0$.

(ii) *The system is s-output-controllable at the initial state* $x(0) = \delta_{2^n}^{j}$, *if and only if* $Col_j(\bar{V}_s) > 0$.

(iii) *The system is s-output-controllable, if and only if* $\bar{V}_s > 0$.

(iv) *The system is output-controllable at the initial state* $x(0) = \delta_{2^n}^{j}$, *if and only if there exists an integer N, such that* $Col_j(\sum_{p=1}^{N} \bar{V}_p) > 0$.

(v) *The system is output-controllable, if and only if there exists an integer N, such that* $(\sum_{p=1}^{N} V_p) > 0$.

[2] Chen H and Sun J. Output controllability and optimal output control of state-dependent switched Boolean control networks. Automatica, 2014, 50(7): 1929-1934.

Representative results [3]

Model

An SBN with state and input constraints can be described as

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

Let $|C_x| = \alpha$ and $|Cu| = \beta$; then **the state's constraint set and the input's constraint set** can be expressed as $C_x = \{\delta_{2^n}^{i_k}: k = 1, \ldots, \alpha;\ i_1 < \cdots < i_\alpha\}$ and $C_u = \{\delta_{2^m}^{j_k}: k = 1, \ldots, \beta;\ j_1 < \cdots < j_\beta\}$, respectively.

[3] Li H and Wang Y. Controllability analysis and control design for switched Boolean networks with state and input constraints. SIAM Journal on Control and Optimization, 2015, 53(5): 2955-2979.

Representative results [3]

Main results (**Constrained incidence matrix**)

Define the following block selection matrices:

$$J_i^{(p,q)} := \underbrace{\left[0_{q\times q} \ \cdots \ 0_{q\times q} \ \underbrace{I_q}_{i\text{–th}} \ 0_{q\times q} \ \cdots \ 0_{q\times q} \right]}_{p} \in \mathbb{R}^{q\times pq}, \quad i = 1,2,\ldots,p,$$

Using the block selection matrices, we thus obtain a new matrix for each i as follows:

$$\widehat{L}_i = \left[\widehat{Blk}_1(L_i) \ \cdots \ \widehat{Blk}_{2^m}(L_i) \right] \left[\left(J_{j_1}^{(2^m,\alpha)}\right)^T \ \cdots \ \left(J_{j_\beta}^{(2^m,\alpha)}\right)^T \right] \in \mathcal{B}_{\alpha\times\alpha\beta}, \quad i = 1,2,\ldots,w,$$

where

$$\widehat{Blk}_s(L_i) = \begin{bmatrix} J_{i_1}^{(2^n,1)} \\ \vdots \\ J_{i_\alpha}^{(2^n,1)} \end{bmatrix} Blk_s(L_i) \left[\left(J_{i_1}^{(2^n,1)}\right)^T \ \cdots \ \left(J_{i_\alpha}^{(2^n,1)}\right)^T \right] \in \mathcal{B}_{\alpha\times\alpha}.$$

The constrained incidence matrix

$$\mathcal{I} = \left.\begin{bmatrix} \widehat{L} \\ \widehat{L} \\ \vdots \\ \widehat{L} \end{bmatrix}\right\} w\beta \in \mathcal{B}_{w\alpha\beta\times w\alpha\beta},$$

[3] Li H and Wang Y. Controllability analysis and control design for switched Boolean networks with state and input constraints. SIAM Journal on Control and Optimization, 2015, 53(5): 2955-2979.

Representative results [4]

Main results (**Set controllability**)

$$\begin{cases} x(t+1) = L\sigma(t)u(t)x(t), \\ y(t) = Hx(t), \end{cases} \quad (5)$$

Definition 3 (Set controllability). SBCN (5) is

(1) set controllable from $s_j^0 \in X^0$ to $s_i^d \in X^d$, if there exist $x_0 \in s_j^0$ and $x_d \in s_i^d$ such that system (5) is controllable from x_0 to x_d;
(2) set controllable at s_j^0, if it is set controllable from s_j^0 to any $s_i^d \in X^d$;
(3) set controllable, if it is set controllable at any $s_j^0 \in X^0$.

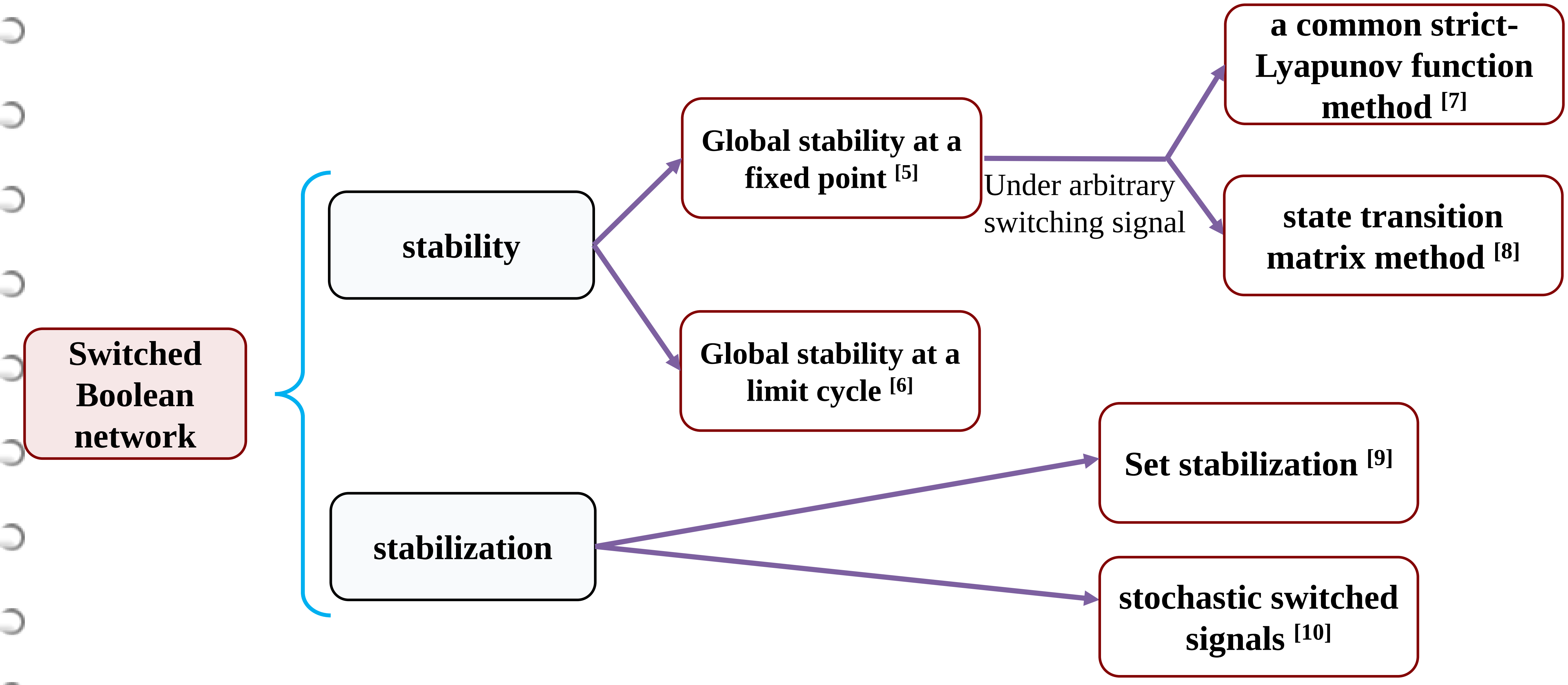
Theorem 1. *Consider SBCN (5) with initial sets* $X^0 = \{s_1^0, s_2^0, \ldots, s_p^0\}$ *and destination sets* $X^d = \{s_1^d, s_2^d, \ldots, s_q^d\}$*, where* $s_j^0, s_i^d \in \mathcal{P}(\Delta_{2^n})\backslash\{\emptyset\}$, $i = 1, \ldots, q$, $j = 1, \ldots, p$. *Then the set controllability matrix can be expressed as*

$$\mathcal{C}_s = J_d^T \times_{\mathcal{B}} M^{[2^n]} \times_{\mathcal{B}} J_0 \in \mathcal{B}_{q\times p}, \quad (9)$$

where $J_0 := [V(s_1^0)\ V(s_2^0)\ \cdots\ V(s_p^0)]$ *and* $J_d := [V(s_1^d)\ V(s_2^d)\ \cdots\ V(s_q^d)]$.

[4] Zhang Q, Feng J, Pan J, et al. Set controllability for switched Boolean control networks. Neurocomputing, 2019, 359: 476-482.

Stability and stabilization of switched Boolean networks



Representative results [5]

Model

Consider a switched BCN with n nodes, m control inputs and w sub-networks as

$$\begin{cases} x_1(t+1) = f_1^{\sigma(t)}(x_1(t), x_2(t), \ldots, x_n(t)), \\ x_2(t+1) = f_2^{\sigma(t)}(x_1(t), x_2(t), \ldots, x_n(t)), \\ \vdots \\ x_n(t+1) = f_n^{\sigma(t)}(x_1(t), x_2(t), \ldots, x_n(t)), \end{cases} \quad (3.1)$$

where $\sigma\colon N \to W = \{1, 2, \ldots, w\}$ is the switching signal.

$$x(t+1) = L_{\sigma(t)} x(t) \quad (3.4)$$

$$\sigma(t) = Gx(t) \quad (3.5)$$

$$x(t+1) = \widehat{L}\sigma(t)x(t), \quad (3.6)$$

where $\widehat{L} = [L_1 \cdots L_w] \in \mathcal{L}_{2^n \times w2^n}$,

Definition 3.1. The system (3.1) is said to be consistently stabilizable to $X_e = (x_1^e, \ldots, x_n^e) \in \mathcal{D}^n$, if there exists a switching signal $\sigma : \mathbb{N} \to W$, such that the trajectory initialized at any $X_0 = (x_1(0), \ldots, x_n(0)) \in \mathcal{D}^n$ converges to X_e under σ.

Assumption 3.3. $X_e = (x_1^e, \ldots, x_n^e)$ is a fixed point of the k-th sub-network of the system (3.1).

[5] Li H and Wang Y. Consistent stabilizability of switched Boolean networks. Neural Networks, 2013, 46: 183-189.

Representative results [5]

Model

It is easy to see that, under a free-form switching sequence $\{(0, \sigma(0)), (1, \sigma(1)), \ldots, (\tau - 1, \sigma(\tau - 1))\}$, along the trajectory starting from any initial state $x(0) \in \Delta_{2^n}$, we have

$$\begin{aligned} x(\tau) &= \widehat{L}\sigma(\tau-1)\cdots\widehat{L}\sigma(1)\widehat{L}\sigma(0)x(0) \\ &= \widehat{L}(I_w \otimes \widehat{L})\cdots(I_{w^{\tau-1}} \otimes \widehat{L}) \ltimes_{i=\tau-1}^{0} \sigma(i)x(0) \\ &:= \widehat{\overline{L}} \ltimes_{i=\tau-1}^{0} \sigma(i)x(0), \end{aligned} \tag{3.7}$$

where $\ltimes_{i=\tau-1}^{0} \sigma(i) = \sigma(\tau-1) \ltimes \sigma(\tau-2) \ltimes \cdots \ltimes \sigma(0)$, and

$$\widehat{\overline{L}} = \widehat{L}(I_w \otimes \widehat{L})\cdots(I_{w^{\tau-1}} \otimes \widehat{L}) \in \mathcal{L}_{2^n \times w^\tau 2^n}. \tag{3.8}$$

Split $\widehat{\overline{L}}$ into w^τ equal blocks as

$$\widehat{\overline{L}} = \left[Blk_1\left(\widehat{\overline{L}}\right)\cdots Blk_{w^\tau}\left(\widehat{\overline{L}}\right)\right], \tag{3.9}$$

where $Blk_i\left(\widehat{\overline{L}}\right) \in \mathcal{L}_{2^n \times 2^n}$. Then, for $\ltimes_{i=\tau-1}^{0} \sigma(i) = \delta_{w^\tau}^{\alpha}$, one can obtain

$$x(\tau) = Blk_\alpha\left(\widehat{\overline{L}}\right)x(0). \tag{3.10}$$

[5] Li H and Wang Y. Consistent stabilizability of switched Boolean networks. Neural Networks, 2013, 46: 183-189.

Representative results [5]

Main results (**Consistent stabilizability of SBNs**)

Consider the consistent stabilizability of the system (3.1) by a **free-form switching signal**.

Theorem 3.4. *Consider the system (3.1), and assume that Assumption 3.3 holds. Then, the system is consistently stabilizable to $x_e = \delta_{2^n}^{\mu}$ by a free-form switching signal, if and only if there exist two positive integers $1 \le \tau \le w(2^n)^{2^n}$ and $1 \le \alpha \le w^{\tau}$, such that*

$$Blk_{\alpha}\left(\widehat{L}\right) = \delta_{2^n}[\underbrace{\mu \cdots \mu}_{2^n}] \tag{3.11}$$

holds, where $\widehat{L}$ is given in (3.9). Moreover, if (3.11) holds, then the free-form consistent stabilizing switching signal is given as

$$\sigma(t) = \begin{cases} \sigma^*(t), & 0 \le t \le \tau - 1, \\ \delta_w^k, & t \ge \tau, \end{cases} \tag{3.12}$$

where σ^* is determined by $\ltimes_{i=\tau-1}^{0} \sigma^*(i) = \delta_{w^{\tau}}^{\alpha}$.

Consider the consistent stabilizability of the system (3.1) by a **state-feedback switching signal**.

Theorem 3.5. *Consider the system (3.1), and assume that Assumption 3.3 holds. Then, the system is consistently stabilizable to $x_e = \delta_{2^n}^{\mu}$ by a state-feedback switching signal $\sigma(t) = Gx(t)$, if and only if there exists a positive integer $1 \le \tau \le 2^n$ such that*

$$(\widehat{LG\Phi_n})^{\tau} = \delta_{2^n}[\underbrace{\mu \cdots \mu}_{2^n}], \tag{3.20}$$

where $\widehat{L}$ and G are given in (3.6) and (3.5), respectively.

[5] Li H and Wang Y. Consistent stabilizability of switched Boolean networks. Neural Networks, 2013, 46: 183-189.

Representative results [6]

Main results (Global stability at a limit cycle)

Definition 3.1. A switched Boolean network (1) is globally stable at the cycle $\mathcal{C}=(\delta_{2^n}^{r_1},\delta_{2^n}^{r_2},\ldots,\delta_{2^n}^{r_k})$, if for arbitrary switching signals, for every $x(0):=x_0\in\Delta_{2^n}$, there exists $T\in Z_+$ such that $x(t)=\delta_{2^n}^{r_j}$ for every $t\geq T$, where $j\in[1,k]$ and $j\equiv(t-T+1)\bmod k$.

$$\begin{cases} x_1(t+1)=f_1^{\sigma(t)}(x_1(t),\ldots,x_n(t)),\\ x_2(t+1)=f_2^{\sigma(t)}(x_1(t),\ldots,x_n(t)),\\ \vdots\\ x_n(t+1)=f_n^{\sigma(t)}(x_1(t),\ldots,x_n(t)), \end{cases} \quad (1)$$

Theorem 3.1. *Consider the switched Boolean network (1) with its algebraic form (3). Then the system (1) is globally stable at the cycle $\mathcal{C}=(\delta_{2^n}^{r_1},\delta_{2^n}^{r_2},\ldots,\delta_{2^n}^{r_k})$ under any switching signal, if and only if there exists a positive integer $s\leq 2^n$ such that*

$$M_{r_2,r_1}=M_{r_3,r_2}=\cdots=M_{r_k,r_{k-1}}=M_{r_1,r_k}=\omega, \quad (4)$$

$$M_{r_i,r_j}^{s}=0,\quad i=k+1,\ldots,2^n,\ j=k+1,\ldots,2^n. \quad (5)$$

$$x(t+1)=L_{\sigma(t)}x(t), \quad (3)$$

[6] Li F. Global stability at a limit cycle of switched Boolean networks under arbitrary switching signals. Neurocomputing, 2014, 133: 63-66.

Representative results [7]

Model

Consider the following SBN:

$$(4.1)\qquad \begin{cases} x_1(t+1) = f_1^{\sigma(t)}(x_1(t), \ldots, x_n(t)), \\ x_2(t+1) = f_2^{\sigma(t)}(x_1(t), \ldots, x_n(t)), \\ \quad\vdots \\ x_n(t+1) = f_n^{\sigma(t)}(x_1(t), \ldots, x_n(t)), \end{cases} \Rightarrow \quad x(t+1) = L_{\sigma(t)}x(t) \quad (4.3)$$

where $\sigma : \mathbb{N} \mapsto W = \{1, 2, \ldots, w\}$ is the switching signal, $x_i \in \mathcal{D}$, $i = 1, 2, \ldots, n$, are logical variables, and $f_i^j : \mathcal{D}^n \mapsto \mathcal{D}$, $i = 1, 2, \ldots, n$, $j = 1, 2, \ldots, w$, are logical functions.

Given a switching signal $\sigma : \mathbb{N} \mapsto W$, assume that $\{t_1, t_2, \ldots, t_s, \ldots\}$ is the switching time sequence. We refer to the sequence $\{\sigma(t_0) = i_0, \sigma(t_1) = i_1, \ldots, \sigma(t_s) = i_s, \ldots\}$ as its switching index sequence, where $t_0 = 0$. Let $h_i = t_{i+1} - t_i$, $i = 0, 1, \ldots, s, \ldots$; then we obtain the following switching sequence:

$$(4.2)\qquad \pi = \{(i_0, h_0), (i_1, h_1), \ldots, (i_s, h_s), \ldots\}.$$

[7] Li H, Wang Y. Lyapunov-based stability and construction of Lyapunov functions for Boolean networks. SIAM Journal on Control and Optimization, 2017, 55(6): 3437-3457.

Representative results [7]

Main results (a common strict-Lyapunov function method)

Theorem 4.2. Consider the SBN (4.1). The system is asymptotically stable at x_e under arbitrary switching signal if and only if all the w subnetworks share a **common strict-Lyapunov function** $V(x)$ in the form of (3.3) satisfying $V(x_e) = 0$.

Remark 4.3. The method to find a common strict-Lyapunov function for the SBN (4.1) contains the following steps:

(3.3) $$V(x_1, x_2, \dots, x_n) = c_0 + c_1 x_1 + c_2 x_2 + \cdots + c_n x_n + c_{n+1} x_1 x_2 + \cdots + c_{2^n-1} x_1 x_2 \cdots x_n,$$

(i) Express the SBN (4.1) as (4.3).

(3.11) $$[c_0, c_1, \dots, c_{2^n-1}]^T = P_n^{-1}[a_1, a_2, \dots, a_{2^n}]^T$$

(ii) Solve the set of inequalities

$$(4.9) \quad \begin{cases} a_{2^n} = 0, \ a_i > 0, \quad 1 \leqslant i \leqslant 2^n - 1, \\ [a_1, a_2, \dots, a_{2^n}] Col_i(L_j - I_{2^n}) < 0, \\ \quad i = 1, 2, \dots, 2^n - 1, \ j = 1, 2, \dots, w, \end{cases}$$

$$V(x_1, x_2, \dots, x_n) = c_1 x_1 + c_2 x_2 + \cdots + c_n x_n + c_{n+1} x_1 x_2 + \cdots + c_{2^n-1} x_1 x_2 \cdots x_n \quad (4.8)$$

and obtain a solution $(a_1, a_2, \dots, a_{2^n-1}, 0)$.

(iii) With the obtained solution, find out $(0, c_1, \dots, c_{2^n-1})$ by (3.11). Then, a common strict-Lyapunov function of the SBN (4.1) is given as (4.8).

Representative results [8]

Model

Consider the following SBN:

(3)
$$\begin{cases} x_1(t+1) = f_1^{\sigma(t)}(x_1(t), \ldots, x_n(t)), \\ x_2(t+1) = f_2^{\sigma(t)}(x_1(t), \ldots, x_n(t)), \\ \quad \vdots \\ x_n(t+1) = f_n^{\sigma(t)}(x_1(t), \ldots, x_n(t)), \end{cases} \Rightarrow x(t+1) = L_{\sigma(t)}x(t).$$

where $\sigma : \mathbb{N} \mapsto W = \{1, 2, \ldots, w\}$ is the switching signal, $x_i \in \mathcal{D}$, $i = 1, 2, \ldots, n$, are logical variables, and $f_i^j : \mathcal{D}^n \mapsto \mathcal{D}$, $i = 1, 2, \ldots, n$, $j = 1, 2, \ldots, w$, are logical functions.

Given a switching signal $\sigma : \mathbb{N} \mapsto W$, assume that $\{t_1, t_2, \ldots, t_s, \ \ldots\}$ is the switching time sequence. We refer to the sequence $\{\sigma(t_0) = i_0, \sigma(t_1) = i_1, \ldots, \sigma(t_s) = i_s, \ldots\}$ as its switching index sequence, where $t_0 = 0$. Let $h_i = t_{i+1} - t_i$, $i = 0, 1, \ldots, s, \ldots$; then we obtain the following switching sequence:

$$\pi = \{(i_0, h_0), (i_1, h_1), \ldots, (i_s, h_s), \ldots\}.$$

[8] Li H, Wang Y, Liu Z. Stability analysis for switched Boolean networks under arbitrary switching signals. IEEE Transactions on Automatic Control, 2014, 59(7): 1978-1982.

Representative results [8]

Main results (state transition matrix method)

Proposition 2: Let $M = \sum_{i=1}^{m} L_i$. Then

$$\sum_{i=1}^{2^n} (M^k)_{ij} = m^k, \; \forall\, j = 1, 2, \cdots, 2^n, \; k \in \mathbb{Z}_+ \tag{10}$$

where m is the number of sub-networks of the system (3).

Theorem 3: The system (3) is globally stable at $x_e = \delta_{2^n}^{i^*}$ under arbitrary switching signal, if and only if there exists a positive integer $k^* \leq 2^n$ such that

$$Row_{i^*}\left(M^{k^*}\right) = \Big[\underbrace{m^{k^*} \cdots m^{k^*}}_{2^n}\Big] \tag{12}$$

where $M = \sum_{i=1}^{m} L_i$, and m is the number of sub-networks for the system (3).

Remark: The necessary and sufficient condition given in Theorem 3 only needs to calculate matrix M^{K^*}, thus avoiding the tedious step of constructing common strict Lyapunov functions.

[8] Li H, Wang Y, Liu Z. Stability analysis for switched Boolean networks under arbitrary switching signals. IEEE Transactions on Automatic Control, 2014, 59(7): 1978-1982.

Representative results [9]

Main results (Set stabilization)

Let $\mathcal{S} = \{\delta_{2^n}^{i_1}, \delta_{2^n}^{i_2}, \dots, \delta_{2^n}^{i_s}\}$ be a subset of Δ_{2^n}.

Definition 3.1 (*Set Stabilization*). The SBCN (1) is said to be $\mathcal{S}$-stabilizable if, for any switching signal and any initial state $x_0 \in \Delta_{2^n}$, there exists a control sequence u and an integer $T \geq 0$ such that $x(t; x_0, u) \in \mathcal{S}, \forall t \geq T$.

$$\begin{cases} x_1(t+1) = f_1^{\sigma(t)}(u_1(t), \dots, u_m(t), x_1(t), \dots, x_n(t)), \\ \vdots \\ x_n(t+1) = f_n^{\sigma(t)}(u_1(t), \dots, u_m(t), x_1(t), \dots, x_n(t)), \end{cases} \quad (1)$$

$$u_i(t) = h_i(x_1(t), \dots, x_n(t)), \quad i = 1, 2, \dots, m, \quad (4)$$

Definition 3.2. The set $\tilde{\mathcal{S}} \subseteq \mathcal{S}$ is the control invariant set of $\mathcal{S}$ for SBCN (1), if for any switching signal, there exists a control sequence u such that $x(t) \in \tilde{\mathcal{S}}$ implies $x(t+1) \in \tilde{\mathcal{S}}$. A set $\mathcal{S}^*$ is called the largest control invariant set of $\mathcal{S}$ for SBCN (1), if it contains the largest number of elements among all the control invariant sets of $\mathcal{S}$.

$$\begin{cases} x_1(t+1) = f_1^{\sigma(t)}(u_{1,\sigma(t)}(t), \dots, u_{m,\sigma(t)}(t), x_1(t), \dots, x_n(t)), \\ \vdots \\ x_n(t+1) = f_n^{\sigma(t)}(u_{1,\sigma(t)}(t), \dots, u_{m,\sigma(t)}(t), x_1(t), \dots, x_n(t)), \end{cases} \quad (5)$$

$$u_{i,\sigma(t)}(t) = h_{i,\sigma(t)}(x_1(t), \dots, x_n(t)), \quad i = 1, 2, \dots, m, \quad (6)$$

Theorem 3.1. *The SBCN (1) is $\mathcal{S}$-stabilizable under arbitrary switching signal by a state feedback controller (4), if and only if,*

(i) $\mathcal{S}^* \neq \emptyset$;
(ii) *there exists an integer T such that $E_T(\mathcal{S}^*) = \Delta_{2^n}$.*

Theorem 3.3. *The SBCN (5) is $\mathcal{S}$-stabilizable under arbitrary switching signals by a state feedback controller in the form (6), if and only if,*

(i) $\mathcal{S}^* \neq \emptyset$;
(ii) *there exists an integer T such that $E_T(\mathcal{S}^*) = \Delta_{2^n}$.*

[9] Li F, Tang Y. Set stabilization for switched Boolean control networks. Automatica, 2017, 78: 223-230.

Representative results [9]

Main results (Set stabilization)

Set stabilization in the case of the control input is independent of the switching signal

Theorem 3.2. *Suppose that conditions* (i) *and* (ii) *in Theorem 3.1 hold. Every* $1 \leq i \leq 2^n$ *corresponds a unique integer* $1 \leq l_i \leq N$ *such that* $\delta_{2^n}^i \in E_{l_i}(\mathcal{S}^*) \setminus E_{l_i-1}(\mathcal{S}^*)$ *where* $E_0(\mathcal{S}^*) = \emptyset$. *Let* $1 \leq h_i \leq 2^m$ *satisfy*

$$\begin{cases} \sum_{\beta=1_{\mathcal{B}}}^{\omega} \mathrm{Col}_i(\mathrm{Blk}_{h_i}(L_\beta)) \leq [\mathcal{S}^*], & \text{for } l_i = 1, \\ \sum_{\beta=1_{\mathcal{B}}}^{\omega} \mathrm{Col}_i(\mathrm{Blk}_{h_i}(L_\beta)) \leq [E_{l_i-1}(\mathcal{S}^*)], & \text{for } l_i \geq 2. \end{cases}$$

Then the feedback control law (4) *with the state feedback matrix H is given by*

$$H = \delta_{2^m}[h_1, h_2, \ldots, h_{2^n}].$$

Set stabilization in the case of the control sequence is dependent on the switching signal

Theorem 3.4. *Suppose that conditions* (i) *and* (ii) *in Theorem 3.3 hold. Every* $1 \leq i \leq 2^n$ *corresponds a unique integer* $1 \leq l_i \leq N$ *such that* $\delta_{2^n}^i \in E_{l_i}(\mathcal{S}^*) \setminus E_{l_i-1}(\mathcal{S}^*)$ *where* $E_0(\mathcal{S}^*) = \emptyset$. *Let* $1 \leq h_i^\beta \leq 2^m$ *satisfy*

$$\begin{cases} \sum_{\beta=1_{\mathcal{B}}}^{\omega} \mathrm{Col}_i(\mathrm{Blk}_{h_i^\beta}(L_\beta)) \leq [\mathcal{S}^*], & \text{for } l_i = 1, \\ \sum_{\beta=1_{\mathcal{B}}}^{\omega} \mathrm{Col}_i(\mathrm{Blk}_{h_i^\beta}(L_\beta)) \leq [E_{l_i-1}(\mathcal{S}^*)], & \text{for } l_i \geq 2, \end{cases}$$

where $\beta = 1, 2, \ldots \omega$. *Then the feedback control law* (6) *with the state feedback matrix H is given by*

$$H_\beta = \delta_{2^m}[h_1^\beta, h_2^\beta, \ldots, h_{2^n}^\beta].$$

Representative results [10]

Model

Consider a switched Boolean control network with n nodes, m controllers, and s switched signals as follows:

$$\begin{cases} x_1(t+1) = l_1^{\theta(t)}(x_1(t), \ldots, x_n(t), u_1(t), \ldots, u_m(t)) \\ x_2(t+1) = l_2^{\theta(t)}(x_1(t), \ldots, x_n(t), u_1(t), \ldots, u_m(t)) \\ \qquad \cdots \\ x_n(t+1) = l_n^{\theta(t)}(x_1(t), \ldots, x_n(t), u_1(t), \ldots, u_m(t)) \end{cases} \quad (1)$$

with controllers

$$\begin{cases} u_1(t+1) = f_1^{\theta(t)}(x_1(t), x_2(t), \ldots, x_n(t)) \\ u_2(t+1) = f_2^{\theta(t)}(x_1(t), x_2(t), \ldots, x_n(t)) \\ \qquad \cdots \\ u_m(t+1) = f_m^{\theta(t)}(x_1(t), x_2(t), \ldots, x_n(t)) \end{cases} \quad (2)$$

where $x_i(t) \in \mathcal{D}$, $l_i^{\theta(t)} : \mathcal{D}^n \to \mathcal{D}$, and $f_j^{\theta(t)} : \mathcal{D}^m \to \mathcal{D}$ are logical functions for $i = 1, 2, \ldots, n, j = 1, 2, \ldots, m$. The switched signal $\theta(t) \in \{1, 2, \ldots, s\}$ and s is a positive integer denoting the number of subnetworks. The switched signal $\theta(t)$ is assumed to be an independent identically distributed (i.i.d) process and with probability distribution $P(\theta(0) = i) = p_i^\theta$, $i \in \{1, 2, \ldots, s\}$, where $0 < p_i^\theta < 1$ and $\Sigma_{i=1}^s p_i^\theta = 1$.

$$\begin{cases} x(t+1) = L_{\theta(t)} u(t) x(t) \\ u(t) = F_{\theta(t)} x(t). \end{cases} \quad (3)$$

[10] Fang M, Wang L, Wu Z G. Asynchronous stabilization of Boolean control networks with stochastic switched signals. IEEE Transactions on Systems, Man, and Cybernetics: Systems, 2019.

Representative results [10]

Main results

Definition 4: Given $x_d \in \Delta_{2^n}$, the switched Boolean control network (3) with stochastic switched signals $\{\theta(t)\}$ is called x_d stabilization in stochastic sense if for any initial value and any initial distribution of $\theta(t)$, there exists control u, such that

$$\lim_{t\to\infty} \mathrm{E}x(t; x_0, \theta(0), u) = x_d.$$

Theorem 1: Assume the initial switched signal of $\theta(t)$ follows a probability distribution $\mathbf{p}^\theta$, then switched Boolean control network (3) is $\delta_{2^n}^r$ stabilized with stochastic switched signals if and only if there exist switched state feedback controllers u, such that:

a) $\delta_{2^n}^r$ is a fixed point, that is $[\mathbf{P}^x]_{r,r} = 1$;

b) for any $x_0 \in \Delta_{2^n}$, there is an admissible path of length $l < 2^n$ from x_0 to $\delta_{2^n}^r$. Equivalently, $\mathrm{Row}_r[(\mathbf{P}^x)^{2^n-1}] > 0$.

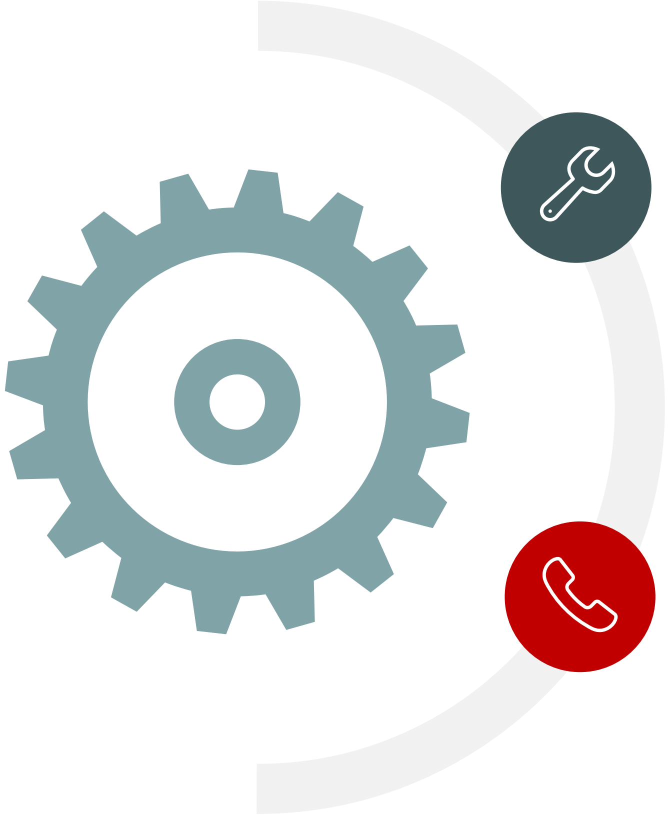
[10] Fang M, Wang L, Wu Z G. Asynchronous stabilization of Boolean control networks with stochastic switched signals. IEEE Transactions on Systems, Man, and Cybernetics: Systems, 2019.

Boolean networks with time delays

- Time delay phenomena is very common in real world, for instance, economic, biological and physiological systems and so on. It is well known that, in many cases, time delay cannot be avoided in practice and it often results in some poor performance.

- Thus, it is necessary for us to investigate Boolean networks with time delays.

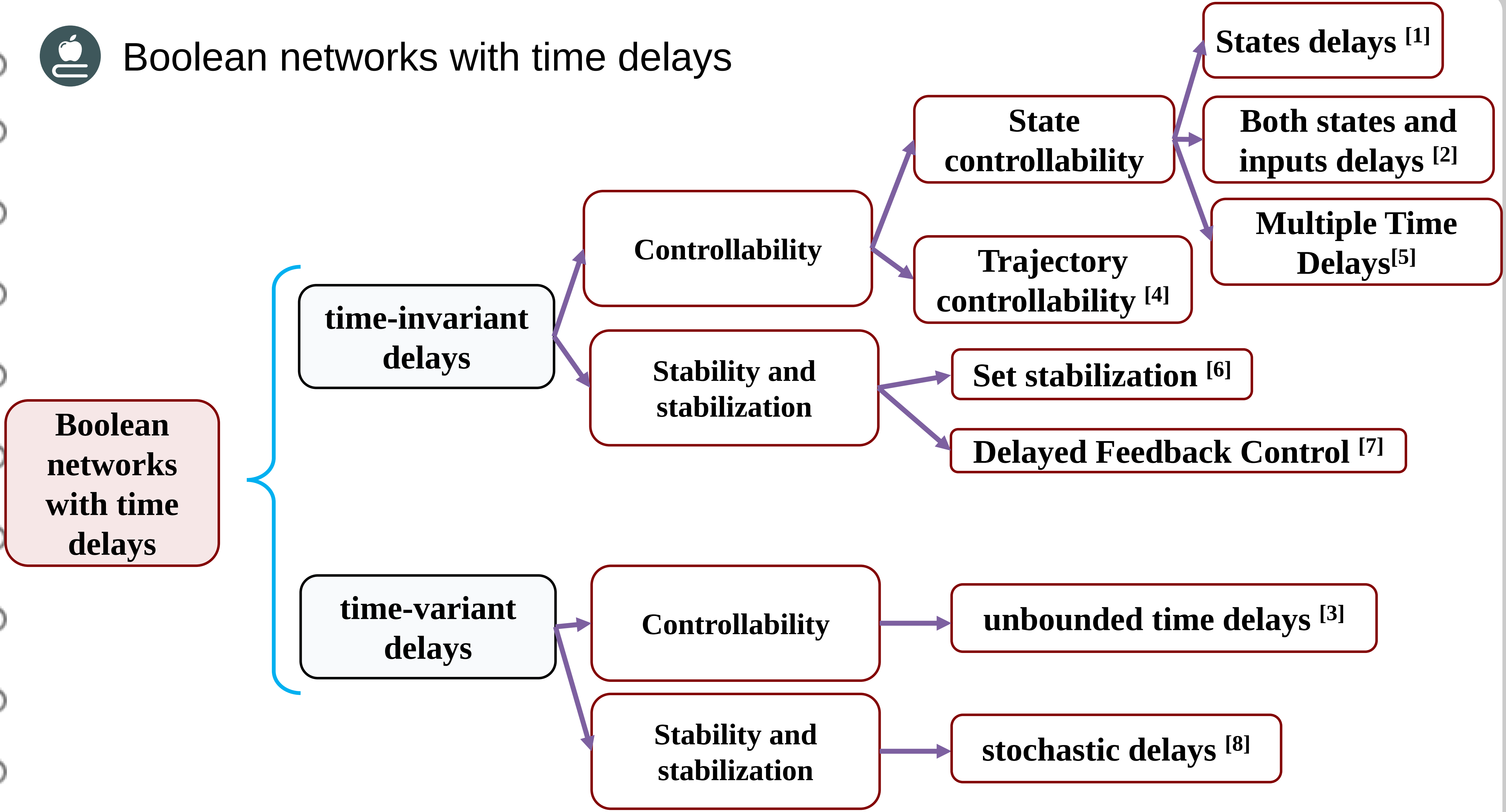
Boolean networks with time delays



Controllability

Stability and stabilization

Boolean networks with time delays



Representative results [1]

Model

Consider Boolean control networks with time-invariant integer delays in states as follows:

$$\begin{cases} A_1(t+1) = f_1(u_1(t), \ldots, u_m(t), A_1(t-\tau), \ldots, A_n(t-\tau)), \\ A_2(t+1) = f_2(u_1(t), \ldots, u_m(t), A_1(t-\tau), \ldots, A_n(t-\tau)), \\ \vdots \\ A_n(t+1) = f_n(u_1(t), \ldots, u_m(t), A_1(t-\tau), \ldots, A_n(t-\tau)), \end{cases} \quad (4)$$

where τ is a positive integer delay.

(1) The controls are logical variables satisfying certain logical rules, called input networks such as:

$$\begin{cases} u_1(t+1) = g_1(u_1(t),\ u_2(t), \ldots, u_m(t)), \\ u_2(t+1) = g_2(u_1(t),\ u_2(t), \ldots, u_m(t)), \\ \vdots \\ u_m(t+1) = g_n(u_1(t),\ u_2(t), \ldots, u_m(t)), \end{cases} \quad (5)$$

$$\begin{cases} u(t+1) = Gu(t), \\ x(t+1) = Lu(t)x(t-\tau) \end{cases} \quad (8)$$

(2) The control is a free Boolean sequence.

[1] Li F, Sun J. Controllability of Boolean control networks with time delays in states. Automatica, 2011, 47(3): 603-607.

Representative results [1]

Main results (under control (1))

Theorem 3.2. *Consider system (4) with control (5), equivalently (8), where G is fixed.* x_d *is s step reachable from* $x(i-\tau)$, $i \in \{0, 1, \ldots, \tau\}$, *if and only if*

$$x_d \in \mathrm{Col}\{\Theta^G(s+i)W_{[2^n,2^m]}x(b-1-\tau)\}$$

where and hereafter "Col" is the column set, also there exist unique $a \in \{0, 1, 2, \ldots\}$ *and* $b \in \{1, 2, \ldots, \tau+1\}$ *such that* $s+i$ *satisfies:*

$$s+i = a(\tau+1)+b$$

and

$$\begin{aligned}\Theta^G(s+i) = {} & LG^{a(\tau+1)+(b-1)}(I_{2^m} \otimes LG^{(a-1)(\tau+1)+(b-1)}) \\ & \times (I_{2^{2m}} \otimes LG^{(a-2)(\tau+1)+(b-1)}) \cdots (I_{2^{am}} \otimes LG^{(b-1)}) \\ & \times (I_{2^{(a-1)m}} \otimes \Phi_m) \cdots (I_{2^m} \otimes \Phi_m)\Phi_m,\end{aligned}$$

where Φ_m *is defined as* $\Phi_m = \ltimes_{i=1}^{m} I_{2^{i-1}} \otimes [(I_2 \otimes W_{[2,2^{m-i}]})M_r]$, $M_r = \delta_4[1, 4]$.

$$\begin{aligned}
x(1) &= Lu(0)x(-\tau),\\
x(2) &= Lu(1)x(1-\tau) = LGu(0)x(1-\tau),\\
&\vdots\\
x(\tau+1) &= Lu(\tau)x(0) = LG^{\tau}u(0)x(0),\\
x(\tau+2) &= LG^{\tau+1}u(0)x(1) = LG^{\tau+1}(I_{2^m} \otimes L)\Phi_m u(0)x(-\tau),\\
x(\tau+3) &= LG^{\tau+2}u(0)x(2)\\
&= LG^{\tau+2}(I_{2^m} \otimes LG)\Phi_m u(0)x(1-\tau),\\
&\vdots\\
x(2(\tau+1)) &= LG^{2\tau+1}u(0)x(\tau+1)\\
&= LG^{2\tau+1}(I_{2^m} \otimes LG^{\tau})\Phi_m u(0)x(0),\\
x(2\tau+3) &= LG^{2\tau+2}u(0)x(\tau+2)\\
&= LG^{2\tau+2}u(0)LG^{\tau+1}(I_{2^m} \otimes L)\Phi_m u(0)x(-\tau)\\
&= LG^{2\tau+2}(I_{2^m} \otimes LG^{\tau+1})(I_{2^{2m}} \otimes L)\\
&\quad \times (I_{2^m} \otimes \Phi_m)\Phi_m u(0)x(-\tau),\\
x(2\tau+4) &= LG^{2\tau+3}u(0)x(\tau+3)\\
&= LG^{2\tau+3}u(0)LG^{\tau+2}(I_{2^m} \otimes LG)\Phi_m u(0)x(1-\tau),\\
&= LG^{2\tau+3}(I_{2^m} \otimes LG^{\tau+2})(I_{2^{2m}} \otimes LG)\\
&\quad \times (I_{2^m} \otimes \Phi_m)\Phi_m u(0)x(1-\tau),\\
&\vdots\\
x(3(\tau+1)) &= LG^{3\tau+2}u(0)x(2\tau+2)\\
&= LG^{3\tau+2}u(0)LG^{2\tau+1}(I_{2^m} \otimes LG^{\tau})\Phi_m u(0)x(0)\\
&= LG^{3\tau+2}(I_{2^m} \otimes LG^{2\tau+1})(I_{2^{2m}} \otimes LG^{\tau})\\
&\quad \times (I_{2^m} \otimes \Phi_m)\Phi_m u(0)x(0),
\end{aligned}$$

[1] Li F, Sun J. Controllability of Boolean control networks with time delays in states. Automatica, 2011, 47(3): 603-607.

Representative results [1]

Main results (under control (2))

Theorem 3.5. *x_d is reachable from $x(i-\tau)$, $i \in \{0, 1, \ldots, \tau\}$ at s steps by controls of Boolean sequences $u(s+i-k-(k-1)\tau)u(s+i-(k-1)-(k-2)\tau)\cdots u(s+i-1)$ if and only if*

$$x_d \in \mathrm{Col}\{\tilde{L}^k x(j-\tau)\}$$

where there exists unique j and k such that

$$s+i-k-k\tau = j-\tau, \quad j \in \{0, 1, \ldots, \tau\}.$$

$$x(t+1) = \tilde{L}x(t-\tau)u(t).$$

It yields:

$$\begin{aligned} x(s+i) &= \tilde{L}x(s+i-1-\tau)u(s+i-1) \\ &= \tilde{L}^2 x(s+i-2-2\tau)u(s+i-2-\tau)u(s+i-1) \\ &= \tilde{L}^3 x(s+i-3-3\tau)u(s+i-3-2\tau) \\ &\quad \times u(s+i-2-\tau)u(s+i-1) \\ &= \cdots \\ &= \tilde{L}^k x(s+i-k-k\tau)u(s+i-k-(k-1)\tau) \\ &\quad \times u(s+i-(k-1)-(k-2)\tau)\cdots u(s+i-1). \end{aligned}$$

Assume that

$$s+i-k-k\tau = j-\tau, \quad \text{where } j \in \{0, 1, \ldots, \tau\}.$$

[1] Li F, Sun J. Controllability of Boolean control networks with time delays in states. Automatica, 2011, 47(3): 603-607.

Representative results [2]

Model

Consider Boolean control network with time-invariant integer delays in both states and controls as follows:

$$\begin{cases} A_1(t+1)=f_1(u_1(t-\tau),\ldots,u_m(t-\tau), \\ A_1(t-\tau),\ldots,A_n(t-\tau)), \\ A_2(t+1)=f_2(u_1(t-\tau),\ldots,u_m(t-\tau), \\ A_1(t-\tau),\ldots,A_n(t-\tau)), \\ \vdots \\ A_n(t+1)=f_n(u_1(t-\tau),\ldots,u_m(t-\tau), \\ A_1(t-\tau),\ldots,A_n(t-\tau)), \end{cases}$$

[2] Han M, Liu Y, Tu Y. Controllability of Boolean control networks with time delays both in states and inputs. Neurocomputing, 2014, 129: 467-475.

Representative results [3]

Model

Consider the following Boolean control networks with n nodes, m inputs, q outputs, and time delays in states:

$$\begin{cases} x_j(t+1) = f_j(u_1(t), \ldots, u_m(t) \\ x_1(t-\tau(t)), \ldots, x_n(t-\tau(t))), \quad j = 1, \ldots, n \\ y_i(t) = h_i(x_1(t), \ldots, x_n(t)), \quad i = 1, \ldots, q \end{cases} \quad (4)$$

$$\begin{cases} x(t+1) = Lu(t)x(t-\tau(t)) \\ y(t) = Hx(t) \end{cases} \quad (5)$$

where $x_1, \ldots, x_n, u_1, \ldots, u_m, y_1, \ldots, y_q \in \mathcal{D}$; $t_0 \in \mathcal{Z}$; $t = t_0, t_0+1, \ldots$; $f_1, \ldots, f_n : \mathcal{D}^{n+m} \to \mathcal{D}$; $h_1, \ldots, h_q : \mathcal{D}^n \to \mathcal{D}$ are logical functions; and $\tau : \{t \in \mathcal{Z} : t \geq t_0\} \to \mathcal{N}$ is a mapping, called the time delay function. Throughout this brief, without loss of generality, we assume that $t - \tau(t) \geq t_0 - \tau(t_0)$ $\forall t \geq t_0$ to ensure that (4) has a starting point. Then the trajectory of (4) is determined by its initial state sequence $x(t_0 - \tau(t_0)), x(t_0 - \tau(t_0) + 1), \ldots, x(t_0)$ (and the control sequence).

[3] Zhang L, Zhang K. Controllability and observability of Boolean control networks with time-variant delays in states. IEEE transactions on neural networks and learning systems, 2013, 24(9): 1478-1484.

Representative results [3]

Main results

Definition 2: A directed graph $G(V, E)$ is said to be the constructed forest of (5) if the vertex set $V = \{t \in \mathcal{Z} : t \geq t_0 - \tau(t_0)\}$, i.e., the time sequence of (5), and the edge set $E = \{(t', t'') : t' = t'' - 1 - \tau(t'' - 1)\} \subset V \times V$.

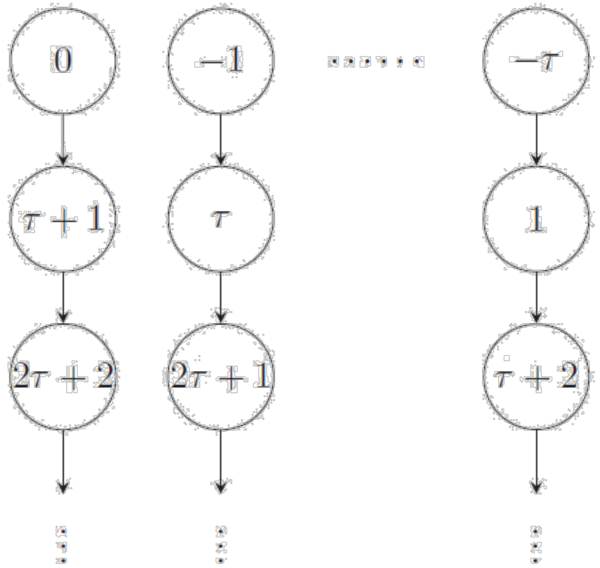


Fig. 1. Constructed forest of (5) with $\tau(t)$ constant that is studied in [7], where the number in each circle denotes the time step.

Denote by

$$\{\mathrm{t}_0, t_1, \ldots, t_N\}(\text{or } \{\mathrm{t}_0, t_1, \ldots\}) \subset \mathcal{Z} \tag{7}$$

one of the controllability constructed paths of (5), where $t_0 - \tau(t_0) \leq \mathrm{t}_0 \leq t_0$, $t_{i+1} > t_i$ for all $i \geq 0$.

$$\begin{cases} x(t_{k+1}) = Lu(t_{k+1} - 1)x(t_k) \\ y(t_k) = Hx(t_k) \end{cases} \tag{8}$$

Theorem 4: Consider (8). Let $L\mathbf{1}_{2^m} := M$. Then:

1) $\delta_{2^n}^i$ is reachable from $\delta_{2^n}^j$ at the sth step if and only if $(M^s)_{ij} > 0$;
2) (8) is controllable from $\delta_{2^n}^j$ if and only if all the entries of $\mathrm{Col}_j(\sum_{k=1}^{\min\{2^{n+m}-1,N\}} M^k)$ are positive;
3) (8) is controllable if and only if all the entries of $\sum_{k=1}^{\min\{2^{n+m}-1,N\}} M^k$ are positive,

where N is the length of (7).

[3] Zhang L, Zhang K. Controllability and observability of Boolean control networks with time-variant delays in states. IEEE transactions on neural networks and learning systems, 2013, 24(9): 1478-1484.

[7] Li F, Sun J. Controllability of boolean control networks with time delays in states, Automatica, 47(3): 603–607, 2011.

Representative results [3]

Main results

Theorem 5: Consider (5). Let $L\mathbf{1}_{2^m} := M$, and set

$$M_{t_0-\tau(t_0)+s} = \begin{cases} \sum_{k=1}^{\min\{N_s, 2^{n+m}-1\}} M^k, & \text{if } N_s > 0 \\ \mathbf{0}_{2^n \times 2^n}, & \text{if } N_s = 0 \end{cases}$$

$s = 0, 1, \ldots, \tau(t_0)$. Then:

1) $\delta_{2^n}^i$ is reachable from $X_0 = (\delta_{2^n}^{i_0}, \delta_{2^n}^{i_1}, \ldots, \delta_{2^n}^{i_{\tau(t_0)}})$ if and only if $\sum_{l=0}^{\tau(t_0)} (M_{t_0-\tau(t_0)+l})_{i,i_l} > 0$;
2) (5) is controllable from $X_0 = (\delta_{2^n}^{i_0}, \delta_{2^n}^{i_1}, \ldots, \delta_{2^n}^{i_{\tau(t_0)}})$ if and only if $\sum_{l=0}^{\tau(t_0)} (M_{t_0-\tau(t_0)+l})_{i,i_l} > 0$ for all $i = 1, 2, \ldots, 2^n$;
3) (5) is controllable if and only if (8) is controllable;
4) (5) is controllable if and only if all the entries of $\sum_{k=1}^{\min\{N_c, 2^{n+m}-1\}} M^k$ are positive.

[3] Zhang L, Zhang K. Controllability and observability of Boolean control networks with time-variant delays in states. IEEE transactions on neural networks and learning systems, 2013, 24(9): 1478-1484.

Representative results [4]

On controllability of delayed Boolean control networks

This paper is devoted to studying the trajectory and state controllability of BCNs with time delay. In contrast to BCNs without time delay, the dynamics of delayed BCNs are determined by a sequence of initial states, named here trajectories. Trajectory controllability means that there exists a control signal steering a system from an initial trajectory to a desired trajectory, while state controllability means that there exists a control signal steering an initial state to a given state.

Here, both trajectory controllability and state controllability are studied. It should be noted that in this paper, trajectory controllability does not mean tracking or following a given trajectory. In fact it means to control BCNs to a destination trajectory of length μ at the k-th step.

[4] J.Q. Lu*, J. Zhong, D.W.C. Ho, Y. Tang and J.D. Cao. On controllability of delayed Boolean control networks. SIAM Journal on Control and Optimization, 54(2): 475-494, 2016.

Representative results [4]

Model

A BCN with high order time delay:

$$\begin{cases} x_1(t+1) &= f_1\big(u_1(t),\dots,u_m(t),x_1(t-\mu+1), \\ &\quad \dots,x_n(t-\mu+1),\dots,x_1(t),\dots,x_n(t)\big) \\ x_2(t+1) &= f_2\big(u_1(t),\dots,u_m(t),x_1(t-\mu+1), \\ &\quad \dots,x_n(t-\mu+1),\dots,x_1(t),\dots,x_n(t)\big) \\ &\vdots \\ x_n(t+1) &= f_n\big(u_1(t),\dots,u_m(t),x_1(t-\mu+1), \\ &\quad \dots,x_n(t-\mu+1),\dots,x_1(t),\dots,x_n(t)\big) \end{cases} \quad (1)$$

where μ is a positive integer denoting the length of the initial states sequence.

Let $u(t)=\ltimes_{i=1}^{m}u_i(t),\ x(t)=\ltimes_{i=1}^{n}x_i(t),\ y(t)=\ltimes_{i=t-\mu+1}^{t}x(i),$ then, we get algebraic form:

$$x(t+1)=L_0u(t)y(t), \quad (2)$$

where $\mathrm{Col}_j(L_0)=\ltimes_{i=1}^{n}\mathrm{Col}_j(M_i),\ L_0\in\mathcal{L}_{2^n\times 2^{\mu n+m}}$ Further, we obtain

$$y(t+1)\triangleq Lu(t)y(t), \quad (3)$$

[4] J.Q. Lu*, J. Zhong, D.W.C. Ho, Y. Tang and J.D. Cao. On controllability of delayed Boolean control networks. SIAM Journal on Control and Optimization, 54(2): 475-494, 2016.

Representative results [4]

Model

Remarks:

(a) Here, we call $x(t)$ the state of the delayed BCN (1) and $(x(t-\mu+1),\ldots,x(t))$ as the trajectory of length μ (abbreviated as trajectory).

(b) Since the dynamics of the delayed BCN (1) is determined by its initial state sequence $(x(1-\mu),\ldots,x(0))$ (or named as initial trajectory), it is meaningful to study the trajectory controllability of the delayed BCN (1).

(c) Here, the trajectory means a state sequence of length "μ", and the t-th trajectory of delayed BCN (1) or (1) is denoted by $X(t)=(x(t-\mu+1),\ldots,x(t))$.

[4] J.Q. Lu*, J. Zhong, D.W.C. Ho, Y. Tang and J.D. Cao. On controllability of delayed Boolean control networks. SIAM Journal on Control and Optimization, 54(2): 475-494, 2016.

Representative results [4]

Main results

Definition Consider the delayed BCN (1). Given initial trajectory $X(0) = (x(1-\mu), x(2-\mu), \dots, x(0))$ and destination trajectory $X_d = (x_d^1, x_d^2, \dots, x_d^\mu)$:

1. X_d is said to be **trajectory controllable** (or **trajectory reachable**) from $X(0)$ at the k-th step, if we can find a sequence of control $U(k) = (u(0), \dots, u(k-1))$ such that $X(0)$ can be driven to the destination trajectory X_d, that is $X(k) = X_d$.
2. The set of all trajectories that are trajectory reachable from $X(0)$ at the k-th step is said to be the k-step trajectory reachable set of $X(0)$, denoted by $R_k^t(X(0))$.
3. The set of all trajectories that are trajectory reachable from $X(0)$ is said to be the trajectory reachable set of $X(0)$, denoted by $R^t\big(X(0)\big)$.
4. System (1) is said to be **trajectory controllable** from $X(0)$ if $R^t(x(0)) = \Delta_{2^{\mu n}}$. And it is said to be **trajectory controllable** if it is trajectory controllable from any initial trajectory $X(0)$.

[4] J.Q. Lu*, J. Zhong, D.W.C. Ho, Y. Tang and J.D. Cao. On controllability of delayed Boolean control networks. SIAM Journal on Control and Optimization, 54(2): 475-494, 2016.

Representative results [4]

Main results

Theorem 4

1. The destination trajectory $X_d = (x_d^1, x_d^2, \dots, x_d^{\mu})$ with $x_d^i \in \Delta_{2^n}$ is reachable from the initial trajectory $X(0)$ at the k-th step by $u(0), \dots, u(k-1)$ if and only if $y_d = \ltimes_{i=1}^{\mu} x_d^i \in Col\{(\bar{L})^k y(0)\}$, where $\bar{L} = LW_{[2^{\mu n}, 2^m]}$;
2. If we assume that s^* is the smallest $s > 0$ such that $Col\{\bar{L}^{s+1} \ltimes_{i=1}^{\mu} x(i)\} \subseteq Col\{\bar{L}^{s+1} \ltimes_{i=1}^{\mu} x(i) | j = 1, 2, \dots, s\}$, then the reachable trajectory set of $\ltimes_{i=1-\mu}^{0} x(i)$ is
$$R(\ltimes_{i=1-\mu}^{0} x(i)) = \cup_{j=1}^{s^*} Col\{\bar{L}^j \ltimes_{i=1-\mu}^{0} x(i)\}.$$

Let $Q = L \ltimes \mathbf{1}_{2^m}$. The number of different control sequences from $y(0) = y_a$ to $y(k) = y_b$ is given.

$$N_1(k; y_a, y_b) = y_b^T Q^k y_a.$$

Theorem 5: Consider the BCN (1), it is **trajectory controllable** if and only if Q is **irreducible**.

[4] J.Q. Lu*, J. Zhong, D.W.C. Ho, Y. Tang and J.D. Cao. On controllability of delayed Boolean control networks. SIAM Journal on Control and Optimization, 54(2): 475-494, 2016.

Representative results [4]

Main results

If the state $x(t-k+\mu)=\delta_{2^{\mu n}}^{p}$, then the trajectory $y(t)$ belongs to the following set:

$$\Xi_k^p = \{\delta_{2^{\mu n}}^{(i-1)2^{(\mu-k+1)n}+(p-1)2^{(\mu-k)n}+j} : i=1,\ldots,2^{(\mu-k)n}, j=1,\ldots,2^{(\mu-k)n}\}.$$

Theorem 6: The destination state $\delta_{2^{\mu n}}^{p}$ is reachable from the initial state sequence $a = \ltimes_{i=1-\mu}^{0} x(i)$ at the k-th step by a free control sequence $u(0), u(1), \ldots, u(k-1)$ if and only if

$$Col\{\Xi_\mu^p\} \cap Col\{(\bar{L})^k \ltimes_{i=1-\mu}^{0} x(i)\} \neq \emptyset.$$

Theorem 7 $N_2(k;\ a, b_s)$ is the number of different control sequences from $a = \ltimes_{i=1-\mu}^{0} x(i)$ to x(k)=b_s is

$$\begin{aligned} N_2(k;a,b_s) &= \sum_{b\in\Xi_\mu^{p_s}} N_1(k;a,b) \\ &= \sum_{b\in\Xi_\mu^{p_s}} b^T Q^k a. \end{aligned}$$

Remarks: Consider the delayed BCN (1) without any forbidden states or trajectories. If it is trajectory controllable, then it must be state controllable.

[4] J.Q. Lu*, J. Zhong, D.W.C. Ho, Y. Tang and J.D. Cao. On controllability of delayed Boolean control networks. SIAM Journal on Control and Optimization, 54(2): 475-494, 2016.

Representative results [5]

Model (multiple time delays)

A BCN with multiple time-varying delays can be expressed as follows:

$$
\begin{cases}
X_1(t+1) = f_1 & (X_1(t-\tau_1(t)),\dots,X_1(t-\tau_q(t)),\dots,\\
& X_n(t-\tau_1(t))\dots,X_n(t-\tau_q(t)),U_1(t),\\
& \dots,U_m(t))\\
& \vdots \\
X_n(t+1) = f_n & (X_1(t-\tau_1(t)),\dots,X_1(t-\tau_q(t)),\dots,\\
& X_n(t-\tau_1(t)),\dots,X_n(t-\tau_q(t)),U_1(t),\\
& \dots,U_m(t))
\end{cases}
$$

[5] Ding Y, Xie D, Guo Y. Controllability of Boolean control networks with multiple time delays. IEEE Transactions on Control of Network Systems, 2017, 5(4): 1787-1795.

Representative results [6]

Model (set stabilization)

The dynamics of delayed Boolean control networks can be described as follows:

$$\begin{cases} x_1(t+1) = f_1(X(t-\mu+1), \ldots, X(t), U(t)), \\ x_2(t+1) = f_2(X(t-\mu+1), \ldots, X(t), U(t)), \\ \quad \vdots \\ x_n(t+1) = f_n(X(t-\mu+1), \ldots, X(t), U(t)), \end{cases} \tag{3.1}$$

where $\mu \in \mathbb{Z}_+$ denotes the time delay, $X(i) := (x_1(i), x_2(i), \ldots, x_n(i)) \in \mathcal{D}^n$, $i = t-\mu+1, \ldots, t$ the state at time i, $U(t) := (u_1(t), \ldots, u_m(t)) \in \mathcal{D}^m$ the control input at time t, and $f_j : D^{\mu n+m} \to D$, $j = 1, \ldots, n$ the Boolean function. Given an initial trajectory $Z_0 := (X(-\mu+1), \ldots, X(0)) \in \mathcal{D}^{\mu n}$ and a control sequence $U : \mathbb{N} \to \mathcal{D}^m$, denote the state of the DBCN (3.1) at time $t \in \mathbb{N}$ by $X(t; Z_0, U)$.

[6] Zheng Y, Li H, Ding X, et al. Stabilization and set stabilization of delayed Boolean control networks based on trajectory stabilization. Journal of the Franklin Institute, 2017, 354(17): 7812-7827.

Representative results [6]

Main results (set stabilization)

Given the following nonempty set

$$A = \{\delta_{2^n}^{\alpha_1}, \ldots, \delta_{2^n}^{\alpha_w}\} \tag{3.23}$$

with $\alpha_1 < \cdots < \alpha_w$. Construct the following set:

$$\begin{aligned} B &= \{\delta_{2^{\mu n}}^{\gamma} = \delta_{2^n}^{\gamma_1} \ltimes \cdots \ltimes \delta_{2^n}^{\gamma_\mu} : \gamma_i \in \{\alpha_1, \ldots, \alpha_w\}, i = 1, \ldots, \mu\} \\ &:= \{\delta_{2^{\mu n}}^{\chi_1}, \ldots, \delta_{2^{\mu n}}^{\chi_{w^\mu}}\}, \end{aligned} \tag{3.24}$$

where $\chi_1 < \cdots < \chi_{w^\mu}$. For example, given $A = \{\delta_4^1, \delta_4^2\}$ and $\tau = 2$, one can obtain $B = \{\delta_4^1 \ltimes \delta_4^1, \delta_4^1 \ltimes \delta_4^2, \delta_4^2 \ltimes \delta_4^1, \delta_4^2 \ltimes \delta_4^2\} = \{\delta_{16}^1, \delta_{16}^2, \delta_{16}^5, \delta_{16}^6\}$.

Theorem 3.15. *The DBCN (3.1) is stabilizable to the set A (given in Eq. (3.23)) with respect to state by a state feedback control, if and only if it is stabilizable to the set B (defined in Eq. (3.24)) with respect to trajectory by a state feedback control.*

[6] Zheng Y, Li H, Ding X, et al. Stabilization and set stabilization of delayed Boolean control networks based on trajectory stabilization. Journal of the Franklin Institute, 2017, 354(17): 7812-7827.

Representative results [7]

Delayed feedback control for stabilization of Boolean control networks with state delay

In this brief, we study the delayed feedback stabilization problem for Boolean control networks (BCNs) with state delay.

Using the semi-tensor product of matrices, some necessary and sufficient conditions are obtained. For the stabilization of BCNs, detailed procedure to construct the feedback controllers is also presented.

We further derive the number of different feedback controllers, which can successfully stabilize the BCN in a finite time.

[7] R. J. Liu, J. Q. Lu*, Y. Liu, J. D. Cao, and Z.G. Wu. Delayed feedback control for stabilization of Boolean control networks with state delay, IEEE Transactions on Networks and Learning Systems, 29(7):3283-3288, 2018.

Representative results [7]

Model

A BCN with state delay and input delay can be described as

$$\begin{cases} x_1(t+1) = f_1(x_1(t-\tau), \ldots, x_n(t-\tau) \\ \qquad\qquad u_1(t-\tau), \ldots, u_m(t-\tau)) \\ \qquad\qquad \vdots \\ x_n(t+1) = f_n(x_1(t-\tau), \ldots, x_n(t-\tau) \\ \qquad\qquad u_1(t-\tau), \ldots, u_m(t-\tau)) \end{cases} \quad (1)$$

Objective: design state feedback controller in the form

$$\begin{cases} u_1(t) = k_1(x_1(t), \ldots, x_n(t)) \\ \qquad \vdots \\ u_m(t) = k_m(x_1(t), \ldots, x_n(t)) \end{cases}$$

Definition 1. The BCN (1) is said to be globally stabilized to a given state $X_g = (x_1^g, x_2^g, \ldots, x_n^g) \in \mathcal{D}^n$,if for any initial state $X(-\tau), X(-\tau+1), \ldots, X(0) \in \mathcal{D}^n$, there exist a control sequence $U(t), t \in \mathbb{N}^+$, and a positive integer N such that $X(t) = X_g$, for $\forall t \geq N$.

[7] R. J. Liu, J. Q. Lu*, Y. Liu, J. D. Cao, and Z.G. Wu. Delayed feedback control for stabilization of Boolean control networks with state delay, IEEE Transactions on Networks and Learning Systems, 29(7):3283-3288, 2018.

Representative results [7]

Main results

With the help of STP:

$$x(t+1) = Fx(t-\tau)u(t-\tau) \qquad (2)$$

$$u(t) = Kx(t) \qquad (3)$$

Key procedure:

1) $S_0 = \{\delta_{2^n}^{\theta}\}$
2) $S_1 = E(S_0)\backslash S_0$
3) $S_2 = E(S_1)\backslash \bigcup_{i=0}^{1} S_i$

$\vdots$

4) $S_M = E(S_{M-1})\backslash \bigcup_{i=0}^{M-1} S_i.$

Theorem 3: The system can be globally stabilized to state $x_g = \delta_{2^n}^{\theta}$ by a state feedback controller (3) if and only if the following conditions are satisfied:

1) $\delta_{2^n}^{\theta} \in E(\delta_{2^n}^{\theta})$;
2) $\sum_{i=0}^{M} |S_i| = 2^n$.

[7] R. J. Liu, J. Q. Lu*, Y. Liu, J. D. Cao, and Z.G. Wu. Delayed feedback control for stabilization of Boolean control networks with state delay, IEEE Transactions on Networks and Learning Systems, 29(7):3283-3288, 2018.

Representative results [8]

Model (Stochastic Delays)

Consider a Boolean network with n modes and stochastic delays as follows:

$$\begin{cases} x_1(k+1) = f_1(X(k-\tau(k))) \\ x_2(k+1) = f_2(X(k-\tau(k))) \\ \vdots \\ x_n(k+1) = f_n(X(k-\tau(k))) \end{cases} \quad (2)$$

$$x(k+1) = Fx(k-\tau(k)) \quad (4)$$

this paper, $\tau(k)$ is viewed as a time-homogeneous Markov chain that attains values in a finite set $\Omega = \{0, 1, \ldots, \tau\}$ with τ a positive integer. The $(i+1, j+1)$-element of the transition probability matrix $\Pi \in \mathbb{R}^{(\tau+1)\times(\tau+1)}$ of Markov chain $\{\tau(k)|k \geq 0\}$ is given as

$$\pi_{ij} = \mathbf{Pr}\{\tau(k+1) = j|\tau(k) = i\} \quad (3)$$

where $\pi_{ij} \geq 0$ for $i, j \in \Omega$ and $\sum_{j=0}^{\tau} \pi_{ij} = 1$ for any $i \in \Omega$. It can be observed that the trajectory of Boolean network (2) can be uniquely determined by the initial state sequence $x(-\tau), x(-\tau+1), \ldots, x(0)$.

[8] Meng M, Lam J, Feng J E, et al. Stability and stabilization of Boolean networks with stochastic delays. IEEE Transactions on Automatic Control, 2018, 64(2): 790-796.

Representative results [8]

Main results

Definition 2: Boolean network (2) with random delays is said to be globally stochastically stable at $X_e \in \mathcal{D}^n$ if for any initial condition and any initial distribution of $\tau(k)$

$$\lim_{k\to\infty} \mathbf{E}\{X(k)|X(0), X(-1), \ldots, X(-\tau), \tau(0)\} = X_e$$

where $X(k) = (x_1(k), x_2(k), \ldots, x_n(k))$.

To proceed, split the state $x(k) \in \Delta_{2^n}$ as $x(k) = (w^T(k), v(k))^T$ with $w(k) \in \mathbb{R}^{2^n-1}$ and $v(k) \in \mathcal{D}$, then one can easily obtain that

$$w(k+1) = F_{11}w(k-\tau(k)) + F_{12}v(k-\tau(k)) \quad (5)$$

$$v(k+1) = F_{21}w(k-\tau(k)) + F_{22}v(k-\tau(k)) \quad (6)$$

where

$$\begin{bmatrix} F_{11} & F_{12} \\ F_{21} & F_{22} \end{bmatrix} = F, \; F_{11} \in \mathcal{M}_{(2^n-1)\times(2^n-1)}. \quad (7)$$

Let $W(k) = (w^T(k), w^T(k-1), \ldots, w^T(k-\tau))^T$, and $V(k) = (v(k), v(k-1), \ldots, v(k-\tau))^T$. Now, Boolean network (4) with stochastic delay can be converted equivalently into two coupled stochastic switched systems as follows:

$$W(k+1) = (M_1 + G_{11}^{\langle\tau(k)\rangle})W(k) + G_{12}^{\langle\tau(k)\rangle}V(k) \quad (8)$$

$$V(k+1) = (M_2 + G_{22}^{\langle\tau(k)\rangle})V(k) + G_{21}^{\langle\tau(k)\rangle}W(k) \quad (9)$$

Theorem 1: Boolean network (2) is globally stochastically stable at X_e if and only if $F_{22} = 1$ and there exist vectors $\lambda_i \in \mathbb{R}^{(\tau+1)(2^n-1)}$, $i \in \Omega$, satisfying the following LP problem:

$$\sum_{i=0}^{\tau} \pi_{ij}(M_1 + G_{11}^{\langle i\rangle})\lambda_i - \lambda_j < \mathbf{0} \quad (14)$$

$$\lambda_j > \mathbf{0}, \; j \in \Omega. \quad (15)$$

[8] Meng M, Lam J, Feng J E, et al. Stability and stabilization of Boolean networks with stochastic delays. IEEE Transactions on Automatic Control, 2018, 64(2): 790-796.

Conjunctive Boolean Networks (CBNs)

- A BN is **conjunctive** if the associated value update rule is comprised of **only AND operations.**
- A **canalyzing function** is such that if an input of the function holds a certain value, called the "canalyzing value", then the output value of the function is **uniquely** determined regardless of the other values of the inputs.
- The conjunctive Boolean network is **monotonic**.

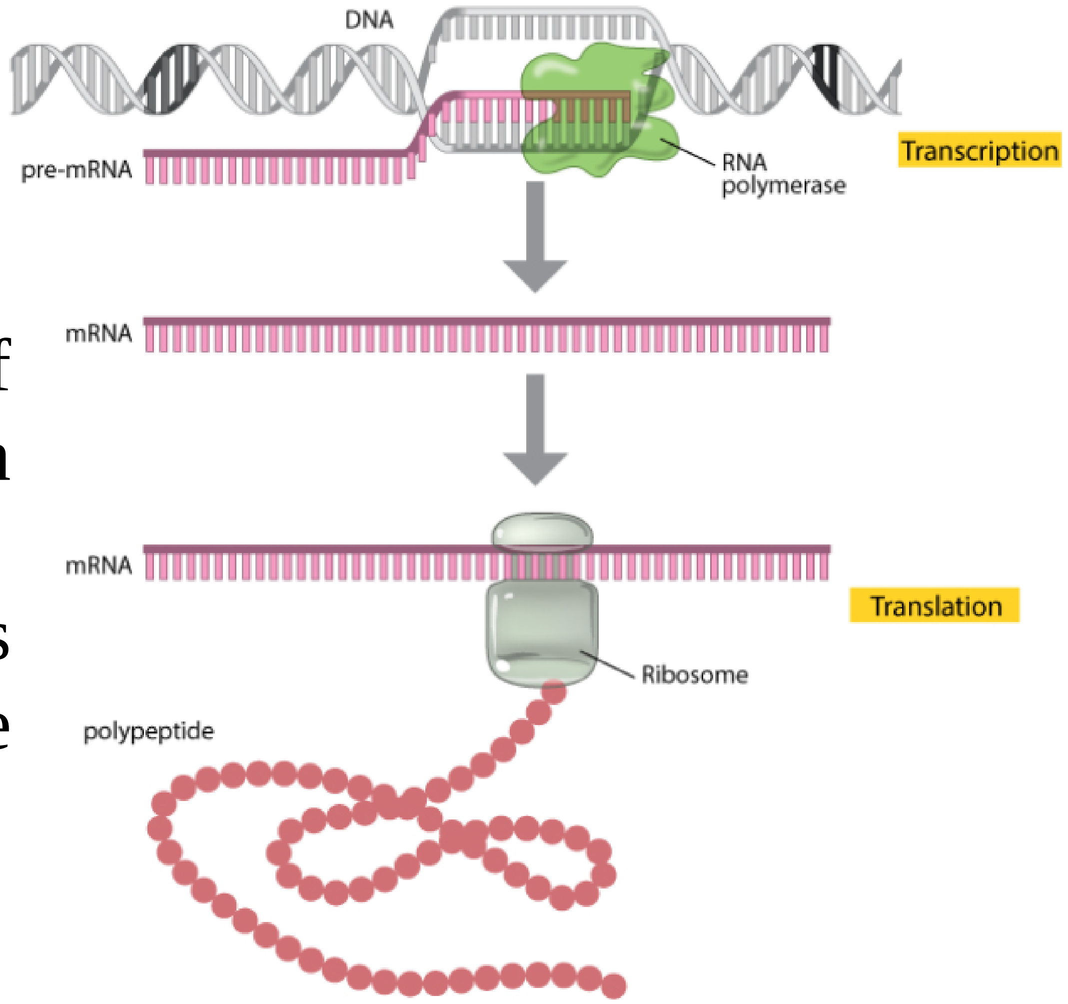


Figure 1.

Conjunctive Boolean Networks (CBNs)

Stability[1]

Controllability[2]

[1] Z. Gao, X. Chen and T. Basar, Stability structures of conjunctive Boolean networks, *Automatica*, 2018, 89: 8-20.

[2] Z. Gao, X. Chen and T. Basar, Controllability of conjunctive Boolean networks with application to gene regulation, *IEEE Transactions on Control of Network Systems*, 2017, 5(2): 770-781.

Conjunctive Boolean Networks (CBNs)

- Let D = (V, E) be a directed graph.

We denote by $v_i v_j$ an edge from v_i to v_j in D. We say that v_i is an *in-neighbor* of v_j and v_j is an *out-neighbor* of v_i. The sets of in-neighbors and out-neighbors of vertex v_i are denoted by $\mathcal{N}_{\text{in}}(v_i)$ and $\mathcal{N}_{\text{out}}(v_i)$, respectively. The *in-degree* and *out-degree* of vertex v_i are defined as $|\mathcal{N}_{\text{in}}(v_i)|$ and $|\mathcal{N}_{\text{out}}(v_i)|$, respectively.

Let v_i and v_j be two vertices of D. A *walk* from v_i to v_j, denoted by w_{ij}, is a sequence $v_{i_0} v_{i_2} \cdots v_{i_m}$ (with $v_{i_0} = v_i$ and $v_{i_m} = v_j$) in which $v_{i_k} v_{i_{k+1}}$ is an edge of D for all $k \in \{0, 1, \ldots, m-1\}$. A walk is said to be a *path* if all the vertices in the walk are pairwise distinct. A *closed walk* is a walk w_{ij} such that the starting vertex and ending vertex are the same, i.e., $v_i = v_j$. A walk is said to be a *cycle* if there is no repetition of vertices in the walk other than the repetition of the starting- and ending-vertex. The *length* of a path/cycle/walk is defined to be the number of edges in that path/cycle/walk.

A *strongly connected graph* is a directed graph such that for any two distinct vertices v_i and v_j in the graph, there is a path from v_i to v_j. A *cycle digraph* is a directed graph that consists of a single cycle.

Conjunctive Boolean Networks (CBNs)

- A **binary necklace** of length p is an equivalence class of p-character strings over the binary set $\mathbb{F}_2 = \{0, 1\}$, taking all rotations (circular shifts) as equivalent. For example, in the case of $p = 4$, there are six different binary necklaces, as illustrated in Fig. 1. A *necklace with fixed density* is a necklace in which the number of zeros (and hence, ones) is fixed. The **order** of a necklace is the cardinality of the corresponding equivalence class, and it is always a divisor of p. An *aperiodic necklace* (see, for example, Varadarajan & Wehrhahn, 1990) is a necklace of order p, i.e., no two distinct rotations of a necklace from such a class are equal. Thus, an aperiodic necklace cannot be partitioned into more than one sub-strings which have the same alphabet pattern. For example, a necklace

---

K. Varadarajan and K. Wehrhahn, Aperiodic rings, necklace rings, and Witt vectors, Advances in Mathematics, 81(1): 1-29, 1990.

Conjunctive Boolean Networks (CBNs)

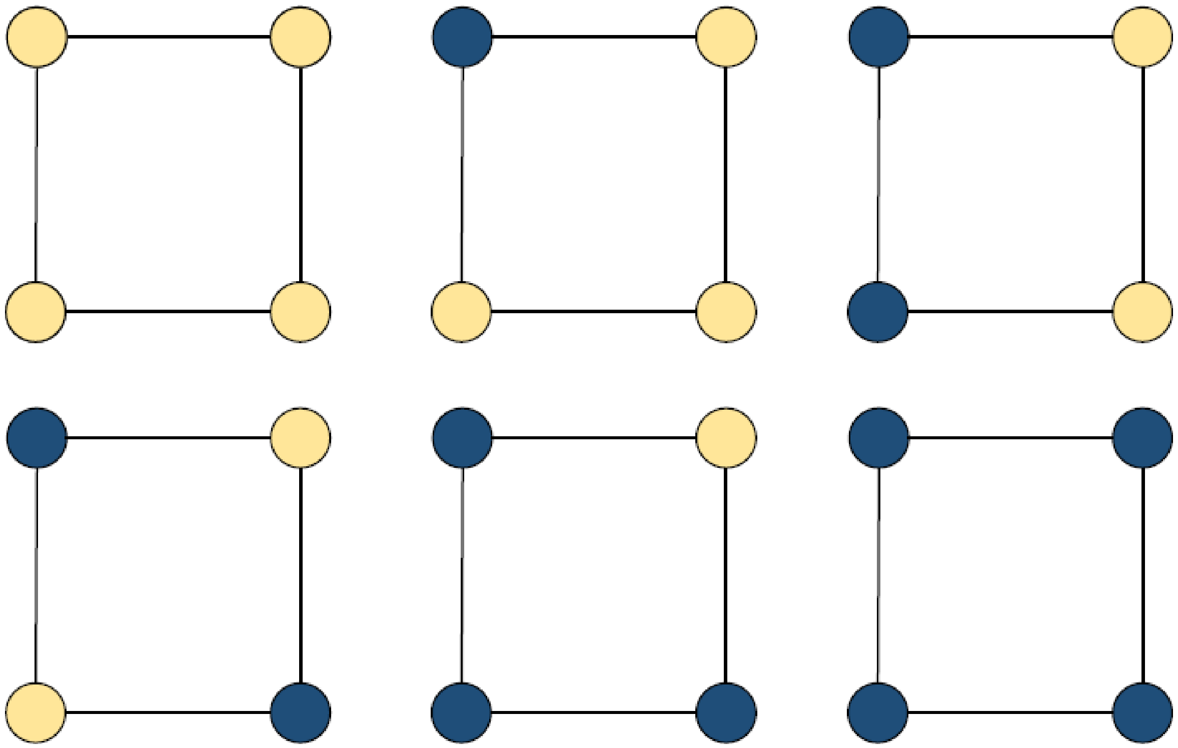


Fig. 1. All binary necklaces of length 4. If the bead is plotted in dark blue (resp. light yellow), then it holds value “1” (resp, “0”). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

Conjunctive Boolean Networks (CBNs)

Stability[1]

Conjunctive Boolean Networks (CBNs) $\longrightarrow$ Stability

Definition 1 (*Conjunctive Boolean Network* *Jarrah et al., 2010*)**.** A Boolean network $f = (f_1, \ldots, f_n)$ is **conjunctive** if each Boolean function f_i, for all $i = 1, \ldots, n$, can be expressed as follows:

$$f_i(x_1, \ldots, x_n) = \prod_{j=1}^{n} x_j^{\epsilon_{ji}} \qquad (1)$$

with $\epsilon_{ji} \in \{0, 1\}$ for all $j = 1, \ldots, n$.

Note that if we let $I_i := \{j \mid \epsilon_{ji} = 1\}$, then f_i is nothing but an AND operator on the variables x_j, for $j \in I_i$.

Jarrah, A.S., Laubenbacher, R. and Veliz-Cuba, A. The Dynamics of Conjunctive and Disjunctive Boolean Network Models, *Bull. Math. Biol.*, 72**:** 1425–1447, 2010.

Conjunctive Boolean Networks (CBNs)

Definition 2 (*Dependency Graph* *Jarrah et al., 2010*). Let $f = (f_1, \ldots, f_n)$ be the value update rule associated with a conjunctive Boolean network. The associated **dependency graph** is a directed graph $D = (V, E)$ of n vertices. An edge from v_i to v_j, denoted by $v_i v_j$, exists in E if and only if $i \in I_j$.

- We assume that the dependency graph D is **strongly connected.**

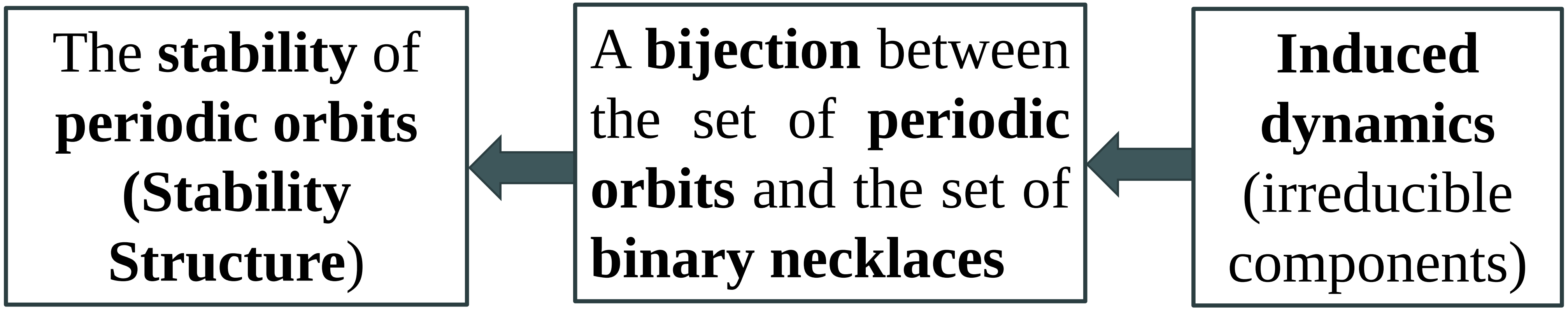
➡ It can be written as **the union of its cycles**, i.e. $D = \left(\cup_{i=1}^{N} V_i, \cup_{i=1}^{N} E_i\right)$ $D_1 = (V_1, E_1), \ldots, D_N = (V_N, E_N)$, are the cycles of D.

- Let $\boldsymbol{n_i}$ be the length of $\boldsymbol{D_i}$. $p^* := \gcd\{n_1, n_2, \ldots, n_N\}$.

Jarrah, A.S., Laubenbacher, R. and Veliz-Cuba, A. The Dynamics of Conjunctive and Disjunctive Boolean Network Models, *Bull. Math. Biol.*, 72**:** 1425–1447, 2010.

Conjunctive Boolean Networks (CBNs) ⟶ Stability

➤ **Framework:**



Conjunctive Boolean Networks (CBNs) ⟶ Stability

Definition 3. Let p divide the lengths of cycles of the dependency graph D. We say that a vertex v_i is **related to** another vertex v_j (or simply write $v_i \sim_p v_j$) if there exists a walk w_{ij} from v_i to v_j such that p divides $l(w_{ij})$. We denote by $l(w_{ij})$ the length of w_{ij}.

- $[v_i]_p := \{v_j \in V \mid v_j \sim_p v_i\}$. $[v_0]_p = [v_0]$ if $p = p^*$.

Proposition 1. *The following two properties hold:*

(1) *If* $v_i \sim_p v_j$, *then* $[v_i]_p = [v_j]_p$. *If* $v_i \nsim_p v_j$, *then* $[v_i]_p \cap [v_j]_p = \varnothing$.
(2) *Let* $v_0 \in V$, *and choose vertices* $v_1, \ldots, v_{p-1}$ *such that* $v_1 \in \mathcal{N}_{\text{out}}(v_0), \ldots, v_{p-1} \in \mathcal{N}_{\text{out}}(v_{p-2})$. *Then, the subsets* $[v_0]_p, \ldots, [v_{p-1}]_p$ *form a partition of* V:

$$V = \sqcup_{i=0}^{p-1} [v_i]_p.$$

Conjunctive Boolean Networks (CBNs) ⟶ Stability

Definition 4 (*Irreducible Components*). Let $D = (V, E)$ be a strongly connected digraph, and p^* be its loop number. Choose a vertex v_0 of D, and let $v_1 \in \mathcal{N}_{\text{out}}(v_0), \ldots, v_{p^*-1} \in \mathcal{N}_{\text{out}}(v_{p^*-2})$. The subsets $[v_0], \ldots, [v_{p^*-1}]$ then form a partition of V. The **irreducible components** of D are digraphs $G_0 = (U_0, F_0), \ldots, G_{p^*-1} = (U_{p^*-1}, F_{p^*-1})$, with their vertex sets U_k's given by

$$U_k := [v_k], \quad \forall k = 0, \ldots, p^* - 1.$$

The edge set F_k of G_k is determined as follows: Let u_i and u_j be two vertices of G_k. Then, $u_i u_j$ is an edge of G_k if there is a walk w_{ij} from u_i to u_j in D with $l(w_{ij}) = p^*$.

Proposition 2. *Each G_k, for $k = 0, \ldots, p^* - 1$, is strongly connected and irreducible.*

Conjunctive Boolean Networks (CBNs) $\longrightarrow$ Stability

Definition 5. A digraph D is a **rose** if all the cycles of D satisfy the following two conditions:

(1) They have the same length.
(2) They share at least one common vertex of D.

Definition 6 (*Induced Dynamics*). An **induced dynamics** on G_k is a conjunctive Boolean network whose dependency graph is G_k.

- Let $U_k = \{u_1, \ldots, u_m\}$, and $(y_1, \ldots, y_m)$ be the state of the network.

$$g_{k_i}(y_1, \ldots, y_m) = \prod_{u_j \in U_k} y_j^{\epsilon_{ji}}$$

where $\epsilon_{ji} = 1$ if u_j is an in-neighbor of u_i and $\epsilon_{ji} = 0$ otherwise.

Conjunctive Boolean Networks (CBNs) ⟶ Stability

Theorem 1. *Let* $G_k = (U_k, F_k)$ *be an irreducible component of* D. *Then, the following hold:*

(1) *Let* g_k *be the induced dynamics on* G_k. *Then,*

$$g_k(x_{U_k}) = f_{U_k}^{p^*}(x), \qquad \forall x \in \mathbb{F}_2^n.$$

(2) *Suppose that* $x(t_0)$ *is in a periodic orbit; then,*

$$x_{U_{(k+1 \bmod p^*)}}(t_0 + 1) = x_{U_k}(t_0). \tag{5}$$

Conjunctive Boolean Networks (CBNs) ⟶ Stability

Theorem 1. *Let $G_k = (U_k, F_k)$ be an irreducible component of D. Then, the following hold:*

(1) *Let g_k be the induced dynamics on G_k. Then,*

$$g_k(x_{U_k}) = f_{U_k}^{p^*}(x), \quad \forall x \in \mathbb{F}_2^n.$$

(2) *Suppose that $x(t_0)$ is in a periodic orbit; then,*

$$x_{U_{(k+1 \bmod p^*)}}(t_0 + 1) = x_{U_k}(t_0).$$

Proposition 5. *There is a bijection between the set of periodic orbits and the set of binary necklaces of length p^*. Moreover, such a bijection maps a periodic orbit of period p to a necklace of order p.*

Conjunctive Boolean Networks (CBNs) $\longrightarrow$ Stability

- Let $\sigma(s)$ be the number of "1"s in the string $s = y_0 \dots y_{p^*-1}$.

Definition 7 (*Successor*). Let s and s' be two periodic orbits. Let $x \in \mathbb{F}_2^n$ be a state in s, and $x' \in \mathcal{I}(x)$. If the trajectory of the dynamics, with x' the initial condition, enters into s' (in finite time steps), then we say that s' is a successor of s.

Definition 8 (*Stability Structure*). The **stability structure** of a conjunctive Boolean network is a digraph $H = (S, A)$, with the vertex set being the set of periodic orbits. The edge set of H is defined as follows: Let s_i and s_j be in S. Then, $s_i s_j$ is an edge of H if s_j is a successor of s_i. Furthermore, an edge $s_i s_j$ of H is a **down-edge** (resp. an **up-edge**) if $\sigma(s_i) > \sigma(s_j)$ (resp, $\sigma(s_i) < \sigma(s_j)$).

Conjunctive Boolean Networks (CBNs) ⟶ Stability

➤ Our goal here is to determine the edge set A of H. To proceed, we first introduce a *partial order* on the set of binary necklaces of length p^*: Let $s = y_0 \dots y_{p^*-1}$ and $s' = y'_0 \dots y'_{p^*-1}$ be two binary necklaces. We say that s is **greater than** s', or simply write $s \succ s'$, if we can obtain s by replacing at least one "0" in s' with "1".

Theorem 2. *Let D be the dependency graph associated with a conjunctive Boolean network, and $H = (S, A)$ be the stability structure. Let s_i and s_j be two vertices of H. Then, there is an edge from s_i to s_j if and only if one of the following three conditions is satisfied:*

(1) *Down-edges:* $s_i \succ s_j$ *and* $\sigma(s_i) - \sigma(s_j) = 1$.
(2) *Up-edges:* $s_i \prec s_j$, $\sigma(s_j) - \sigma(s_i) = 1$, *and D has to be a rose.*
(3) *Self-loops:* $s_i = s_j$, $s_i \neq 1 \dots 1$, *and D is not a cycle digraph.*

Conjunctive Boolean Networks (CBNs)

Controllability[2]

- **Orbit-controllability**
- **State-controllability**

Conjunctive Boolean Networks (CBNs) ⟶ Controllability

- Two questions:

(1) How can one steer the system from any initial state to any **desired periodic orbit**?

- If this is possible, we say that the system is **orbit-controllable** and the subset of variables whose values are determined by external inputs (the controls) is termed the **orbit-controlling set**.

(2) How can one steer the system from any initial state to any **desired final state**?

- If this is possible, we say that the system is **state-controllable** and the subset of variables whose values are determined by external inputs (the controls) is termed the **state-controlling set**.

Conjunctive Boolean Networks (CBNs) ⟶ Controllability

- Let $D = (V, E)$ be the dependency graph of a CBN. A node v_i of D is said to be a *control node* if its value at any time step is determined completely by an external control input.

- We denote by V^* the subset of V,

$$x_i(t) = \begin{cases} u_i(t) & \text{if } v_i \in V^*, \\ f_i(x(t-1)) & \text{otherwise} \end{cases} \tag{3}$$

Definition (Derived Graph [42]): Let $D = (V, E)$ be the dependency graph associated with a CBN. Let $V^* \subset V$ be the set of control nodes associated with system (3). The *derived graph* $D' = (V, E')$ is a digraph, with V the node set and $E' = E \setminus \cup_{u \in V^*} \mathcal{E}_{\text{in}}(u)$ the edge set.

[42] Z. Gao, X. Chen, and T. Basar, State-controlling sets for conjunctive Boolean networks, in *Proc. 20th IFAC World Congr.*, Toulouse, France, 14855–14860, 2017.

Conjunctive Boolean Networks (CBNs) ⟶ Controllability

Definition 3 (Orbit-Controlling Set): A subset $V^* \subseteq V$ is an *orbit-controlling* set for **(1)** if for any initial condition $x \in \mathbb{F}_2^n$ and any periodic orbit $\mathcal{O}$ of system **(1),** there exists a time T and a set of control laws $u_i(t)$, for $v_i \in V^*$ and $0 \leq t \leq T$, such that the trajectory generated by system (3) with $x(0) = x$, reaches a state in $\mathcal{O}$ at $t = T$.

Definition 4 (State-Controlling Set): A subset $V^* \subseteq V$ is a *state-controlling set* for **(1)** if for any initial condition x and any final state x^*, there exists a time T and a set of control laws $u_i(t)$ for $v_i \in V^*$ and $0 \leq t \leq T$, such that the trajectory generated by system (3) with $x(0) = x$, reaches x^* at $t = T$.

Conjunctive Boolean Networks (CBNs) → Controllability

♣ Orbit-controllability

Theorem 1: Let the dependency graph $D = (V, E)$ of a CBN be strongly connected. Then, a subset V^* is an orbit-controlling set if and only if the associated derived graph D' is acyclic.

Recall that $V_1, \ldots, V_N$ are the vertex sets of the cycles of D. Then, the statement of Theorem 1 is equivalent to the following statement: $V^* \subseteq V$ is an orbit-controlling set if and only if

$$V^* \cap V_i \neq \varnothing, \quad \forall i = 1, \ldots, N. \tag{5}$$

Conjunctive Boolean Networks (CBNs) → Controllability

♣ Orbit-controllability

➤ The desired periodic orbit is $s = y_0 \dots y_{p^*-1}$.

Algorithm 1: Control Law for Orbit-Controlling.

procedure CONTROL(V^*, s)
    $t \leftarrow 0$
    while $x(t) \neq (1, \dots, 1) \in \mathbb{F}_2^n$ **do**
        $x_{V^*}(t) \leftarrow (1, \dots, 1)$
        $t \leftarrow t + 1$
    end while
    $\tau \leftarrow t$
    pick any $v_i \in V^*$
    for $t' := 0$ **to** $p^* - 1$ **do**
        $x_i(\tau + t') \leftarrow y_{p^*-1-t'}$;
    end for
end procedure

Conjunctive Boolean Networks (CBNs) ⟶ Controllability

♣ State-controllability

Theorem 2: Let $D = (V, E)$ be the dependency graph associated with a CBN. A subset $V^* \subseteq V$ is a state-controlling set if and only if the associated derived graph D' satisfies the following conditions.

1) The derived subgraph D' is acyclic.
2) For any $v \in V$, there exists a control node $u \in V^*$ and an integer $k \geq 0$, such that $\mathcal{N}^k_{\mathrm{out}}(u; D') = \{v\}$.

- We do ***not*** require that the dependency graph *D* be strongly connected.

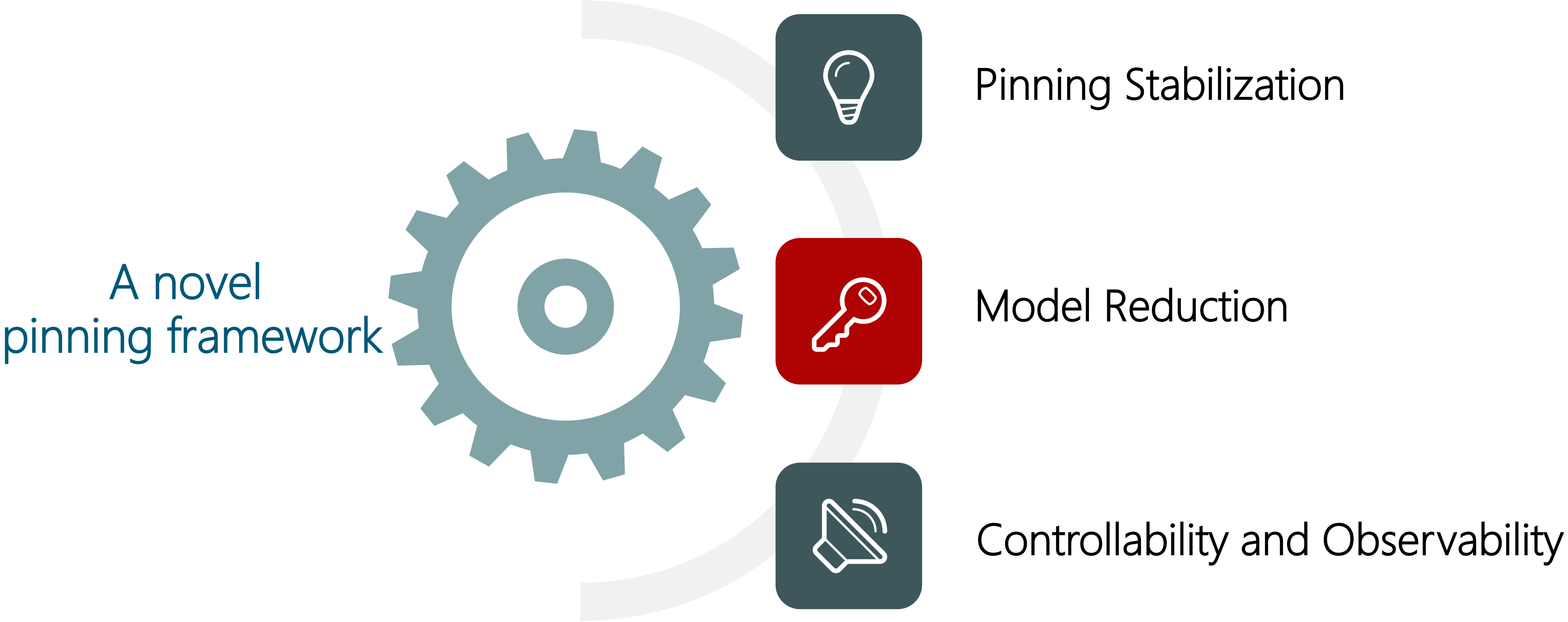
Conjunctive Boolean Networks (CBNs) ⟶ Controllability

♣ State-controllability

Algorithm 2: Control Law for State-Controlling.

procedure CONTROL(V^*, x^*)
    $T \leftarrow$ length of the longest path in D'
    for $t := 0$ **to** T **do**
        for $v_i \in V^*$ **do**
            if $|\mathcal{N}_{\text{out}}^{T-t}(v_i; D')| == 1$ **then**
                if $x^*_{\mathcal{N}_{\text{out}}^{T-t}(v_i; D')} == 0$ **then**
                    $u_i(t) \leftarrow 0$
                    continue
                end if
            end if
            $u_i(t) \leftarrow 1$
        end for
    end for
end procedure

Large-scale Boolean Networks



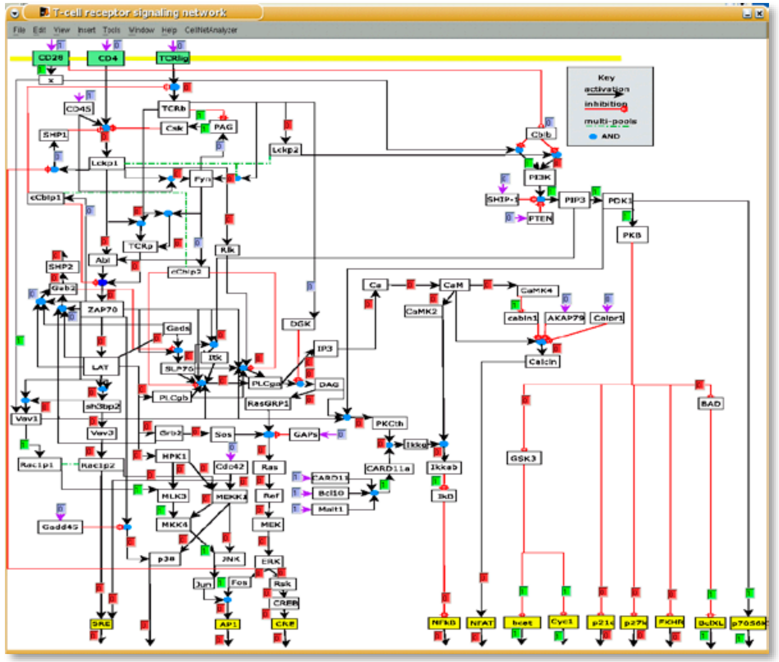
A Novel Pinning Framework

Disadvantages of using $\boldsymbol{x}(\boldsymbol{t}+\mathbf{1}) = \boldsymbol{L}\boldsymbol{x}(\boldsymbol{t})$ (2)

- Information of network structure is missing
- $\boldsymbol{M}$ grows exponentially with network size
- Difficult for large dimension BNs

The existing largest BN model (90 nodes) [27]:

$2^{90} = 1.24 \times 10^{27}$!

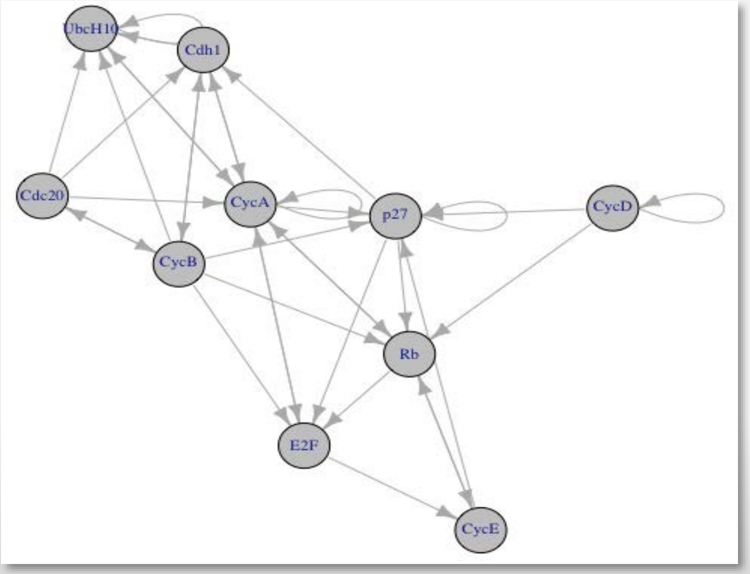


Why do we need a novel framework for pinning control?

[27] J. Saez-Rodriguez, *et al.* A logical model provides insights into T-cell receptor signaling, PLoS Computational Biology 3(8): e163, 2007.

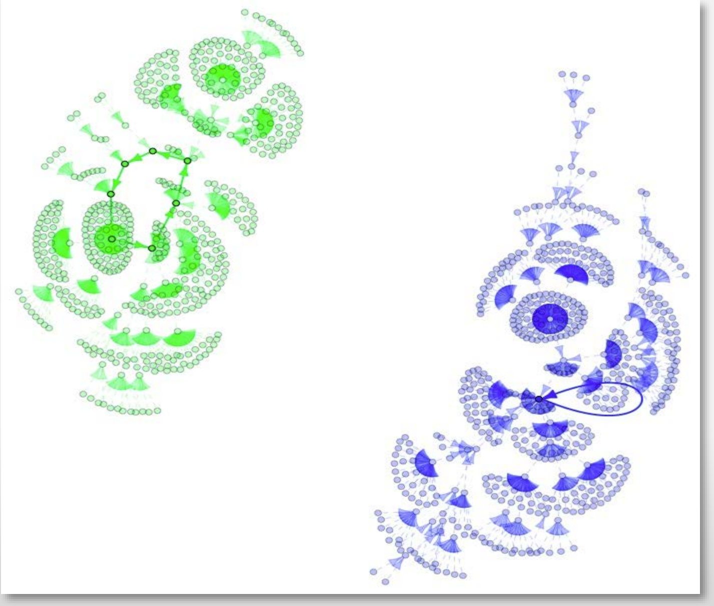
A Novel Pinning Framework

Definition 2 [28]: (**Interaction digraph**) The interaction digraph of a BN (2) is a digraph denoted by $\boldsymbol{G = (V, E)}$, $V = \{x_1, \dots, x_n\}$. An edge $x_i \rightarrow x_j$ exists in $G = (V, E)$ if and only if **logical function f_j is dependent on x_i**.



Interaction digraph (10 nodes)

VS



State transition digraph ($2^{10} = 1024$ nodes)

[28] Faure A, et al. Dynamical analysis of a generic Boolean model for the control of the mammalian cell cycle, Bioinformatics, 2006.

A Novel Pinning Framework

Steady state is unknown!

Lemma 1 [28]: BNs are globally stable if the corresponding **interaction digraph** is **acyclic**.

[29]Global Stabilization w.r.t. Steady state $\delta_{2^n}^{\gamma}$

Step One: Delete the minimum number of edges in $G=(V,E)$, such that $G=(V,E)$ becomes acyclic.

Step Two: Transform structure matrices of controlled nodes, such that $(a_1, \dots, a_n) \sim \delta_{2^n}^{\gamma}$ will be the unique steady state.

Problem 1: How to find control nodes?

Problem 2: How to control these nodes?

Problem 3: How to control Steady State?

[28] Faure A, et al. Dynamical analysis of a generic Boolean model for the control of the mammalian cell cycle, Bioinformatics, 2006.

[29] J. Zhong, D.W.C. Ho and J. Lu. A New Framework for Pinning Control of Boolean Networks, https://arxiv.org/abs/1912.01411.

A Novel Pinning Framework

Problem 1: How to Find Control Nodes

For given set of cycles $C = \{C_1, \dots, C_p\}$ and fixed points $F = \{F_1, \dots, F_q\}$, for **every** $i \in [1, p]$ and $j \in [1, q]$, we try to find edges $e_i \in \Upsilon(C_i)$, $E_j \in \Upsilon(F_i)$, minimize the cost function

$$\mathbf{C}_1 = \min\{|\omega = \bigcup_{i\in[1,p],j\in[1,q]} \mathbf{O}_+(\mathbf{e}_i) \cup \mathbf{O}_+(\mathbf{E}_j)|\},$$

Control nodes

subject to constraint

$$\mathbf{C}_2 = \min\{|\varphi = \bigcup_{i\in[1,p],j\in[1,q]} \mathbf{e}_i \cup \mathbf{E}_j|\}.$$

find the **minimum number of edges** that should be deleted

depth-first search algorithm
\+
linear programming method

find the minimum number of **ending points** for those selected edges

[29] J. Zhong, D.W.C. Ho and J. Lu. A New Framework for Pinning Control of Boolean Networks, https://arxiv.org/abs/1912.01411.

A Novel Pinning Framework

Step 2.1: Solve Feasible Structure Matrices

Consider nodes x_i, $i \in \{\omega_1, \dots, \omega_{|C_1|}\}$, that are selected to be controlled, find matrices $\tilde{A}_{\omega_i} \in \mathcal{L}_{2\times 2^{|N_{\omega_i}|}}$, $\hat{A}_{\omega_i} \in \mathcal{L}_{2\times 2^{|\bar{N}_{\omega_i}|}}$, such that

$$\tilde{A}_{\omega_i} = \hat{A}_{\omega_i}(I_{2^{|\bar{\mathcal{N}}_{\omega_i}|}} \otimes \mathbf{1}^\top_{2^{|\mathcal{N}^C_{\omega_i}|}}), i \in [1, |\mathbf{C}_1|].$$

where

$$\begin{cases} \bar{\mathcal{N}}_{\omega_1} = \mathcal{N}_{\omega_1} \backslash \{\nu_1^1, \dots, \nu_1^{\varepsilon_1}\}, & \dots, \quad \bar{\mathcal{N}}_{\omega_{|\mathbf{C}_1|}} = \mathcal{N}_{\omega_{|\mathbf{C}_1|}} \backslash \{\nu^1_{|\mathbf{C}_1|}, \dots, \nu^{\varepsilon_{|\mathbf{C}_1|}}_{|\mathbf{C}_1|}\}, \\ \mathcal{N}^C_{\omega_1} = \{\nu_1^1, \dots, \nu_1^{\varepsilon_1}\}, & \dots, \quad \mathcal{N}^C_{\omega_{|\mathbf{C}_1|}} = \{\nu^1_{|\mathbf{C}_1|}, \dots, \nu^{\varepsilon_{|\mathbf{C}_1|}}_{|\mathbf{C}_1|}\}. \end{cases}$$

In-neighbors of controlled nodes

Step 2.2: Update Structure Matrices

in-neighbors of controlled nodes that are deleted in $G = (V, E)$

Lemma 1: For $x_{\omega_j}(t+1) = A_{\omega_j} \ltimes_{j \in N_{\omega_j}} x_j(t)$, $j \in [1, C_1]$, swap the positions of neighbors in the following form:

$$\begin{aligned} x_{\omega_j}(t+1) = & \quad A_{\omega_j} \mathbf{W}_j \ltimes_{j \in \bar{\mathcal{N}}_{\omega_j}} x_j(t) \ltimes_{j \in \mathcal{N}^C_{\omega_j}} x_j(t), \\ \triangleq & \quad \bar{A}_{\omega_j} \ltimes_{j \in \bar{\mathcal{N}}_{\omega_j}} x_j(t) \ltimes_{j \in \mathcal{N}^C_{\omega_j}} x_j(t), \end{aligned}$$

where $\mathbf{W}_j \triangleq [\ltimes_{i=1}^{\omega_j} W_{[2, 2^{|\lfloor v_j^i \rfloor \mathcal{N}_{\omega_j}|}]} W_{[2^{|\mathcal{N}^C_{\omega_j}|}, 2^{|\mathcal{N}_{\omega_j}|}]}] \otimes I_{2^{|\mathcal{N}_{\omega_j}| - |\mathcal{N}^C_{\omega_j}|}}$, and $\bar{A}_{\omega_j} \triangleq A_{\omega_j} \mathbf{W}_j$.

A Novel Pinning Framework

Step 2.3: Solve Matrix Equations

Solve matrices $M_{\oplus_j} \in \mathcal{L}_{2\times 4}$ and $K_j \in \mathcal{L}_{2\times 2^{|N_j|}}, j \in \{\omega_1, \dots, \omega_{|C_1|}\}$, from:

$$\begin{cases} \tilde{A}_{\omega_1} = M_{\oplus_{\omega_1}} \mathbf{K}_{\omega_1} (I_{2^{|\mathcal{N}_{\omega_1}|}} \otimes \bar{A}_{\omega_1}) \Phi_{2^{|\mathcal{N}_{\omega_1}|}}, \\ \vdots \\ \tilde{A}_{\omega_{|\mathbf{C}_1|}} = M_{\oplus_{\omega_{|\mathbf{C}_1|}}} \mathbf{K}_{\omega_{|\mathbf{C}_1|}} (I_{2^{|\mathcal{N}_{\omega_{|\mathbf{C}_1|}}|}} \otimes \bar{A}_{\omega_{|\mathbf{C}_1|}}) \Phi_{2^{|\mathcal{N}_{\omega_{|\mathbf{C}_1|}}|}}. \end{cases}$$

Must be solvable by [30]

Feasible structure matrix in Step 2.1

The updated structure matrix in Step 2.2

[30] F. Li, Pinning control design for the stabilization of Boolean networks, " IEEE Trans. Neural Netw. Learn. Syst, vol. 27, pp. 1585–1590, 2015.

A Novel Pinning Framework

Problem 3 :
How to control Steady State

Let $\delta_{2^n}^{\gamma} = \ltimes_{i=1}^{n} \delta_2^{\gamma_i}$. Find matrices $\check{A}_1 \in \mathcal{L}_{2\times 2^{|\bar{N}_1|}}, \dots, \check{A}_n \in \mathcal{L}_{2\times 2^{|\bar{N}_n|}}$ and binary variables $\delta_1, \dots, \delta_n \in \mathcal{D}$, minimize the cost function

$$\mathbf{C}_3 \triangleq \sum_{i=1}^{n} \delta_i,$$

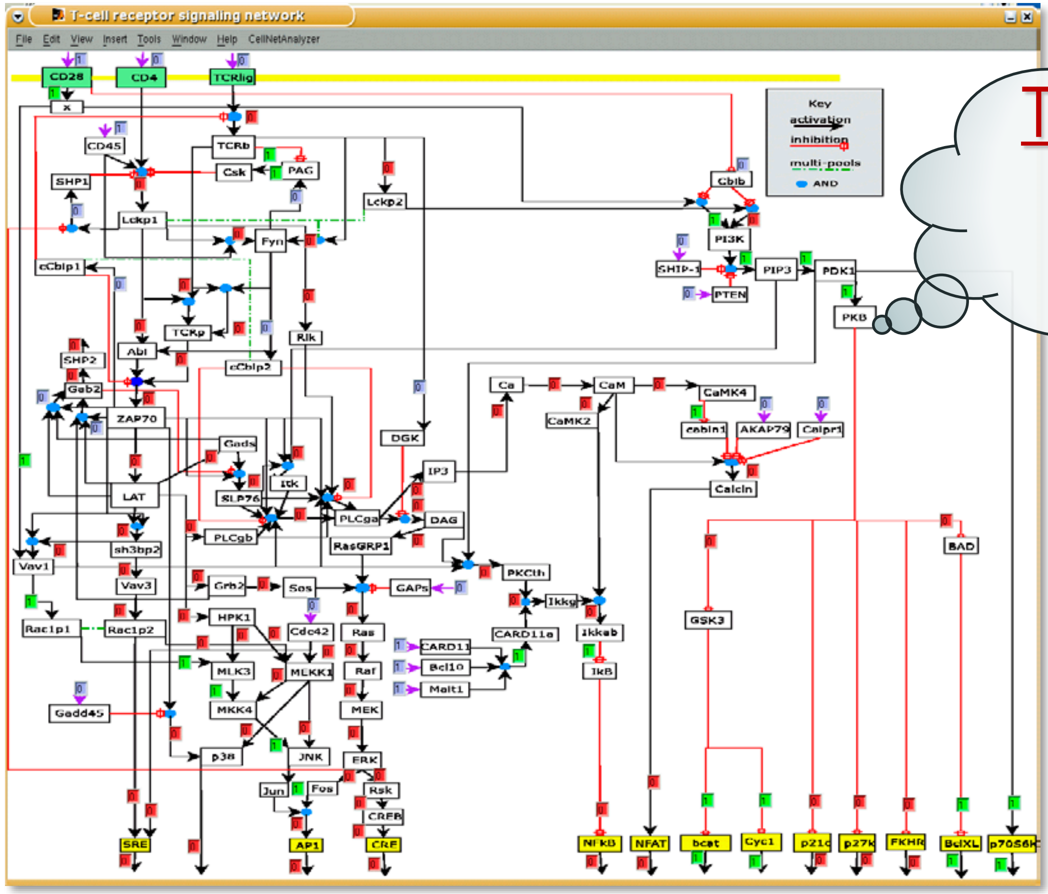
subject to the following conditions:

$$\delta_2^{\gamma_i} = [(1-\delta_i)\hat{A}_i + \delta_i \check{A}_i] \ltimes_{j\in\bar{N}_i} \delta_2^{\gamma_j}, i \in [1, n].$$

- If $\delta_1 = 1$, $\delta_2^{\gamma_1} = [(1-\delta_1)\hat{A}_1 + \delta_1 \check{A}_1] \ltimes_{j\in\bar{N}_1} \delta_2^{\gamma_j}$ reduces to $\delta_2^{\gamma_1} = \check{A}_1 \ltimes_{j\in\bar{N}_1} \delta_2^{\gamma_j}$. This implies that node x_1 should be controlled and its structure matrix $\hat{A}_1$ should be changed to $\check{A}_1$.
- $\Sigma_{i=1}^{n}\delta_i$: the number of further controlled nodes in Problem 3.

A Novel Pinning Framework

Example 2: T-Cell receptor signaling BNs (90 nodes) [27]



The largest modeled BN

Cycles exist!

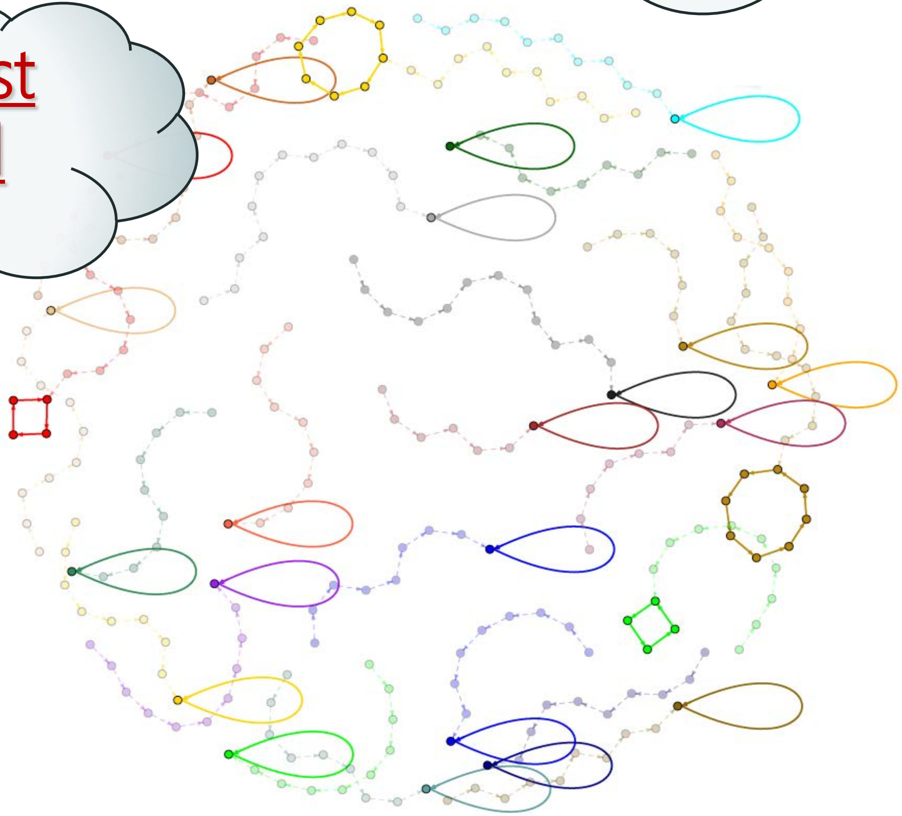


Fig. The interaction digraph.

[27] J. Saez-Rodriguez, *et al.* A logical model provides insights into T-cell receptor signaling, PLoS Computational Biology 3(8): e163, 2007.

A Novel Pinning Framework

The designed pinning control for this network

$$\begin{cases} x_1(*) = u_1 \vee x_1, \quad x_2(*) = u_2 \vee x_2, \quad x_3(*) = u_3 \vee x_3, \\ x_4(*) = u_4 \wedge x_4, \quad x_9(*) = u_9 \vee [x_2 \wedge x_4 \wedge \neg x_6 \wedge \neg x_7], \\ x_{12}(*) = u_{12} \vee x_{17}, \quad x_{21}(*) = u_{21} \wedge x_{21}, \quad x_{22}(*) = u_{22} \vee x_{22}, \quad x_{38}(*) = u_{38} \vee x_{38}, \\ x_{47}(*) = u_{47} \vee x_{47}, \quad x_{52}(*) = u_{52} \vee x_{49}, \quad x_{68}(*) = u_{68} \vee x_{68}, \quad x_{69}(*) = u_{69} \wedge x_{69}, \\ x_{78}(*) = u_{78} \vee x_{78}, \quad x_{79}(*) = u_{79} \vee x_{79}; \end{cases}$$

$$\begin{cases} u_1 = \neg x_1, \quad u_2 = \neg x_2, \quad u_3 = \neg x_3, \quad u_4(*) = \neg x_4, \\ u_9 = x_2 \wedge x_4 \wedge \neg x_6, \quad u_{12} = \neg x_{17}, \quad u_{21} = \neg x_{21}, \\ u_{22} = \neg x_{22}, \quad u_{38} = \neg x_{38}, \quad u_{47} = \neg x_{47}, \quad u_{52} = \neg x_{49}, \\ u_{68} = \neg x_{68}, \quad u_{69} = \neg x_{69}, \quad u_{78} = \neg x_{78}, \quad u_{79} = \neg x_{79}. \end{cases}$$

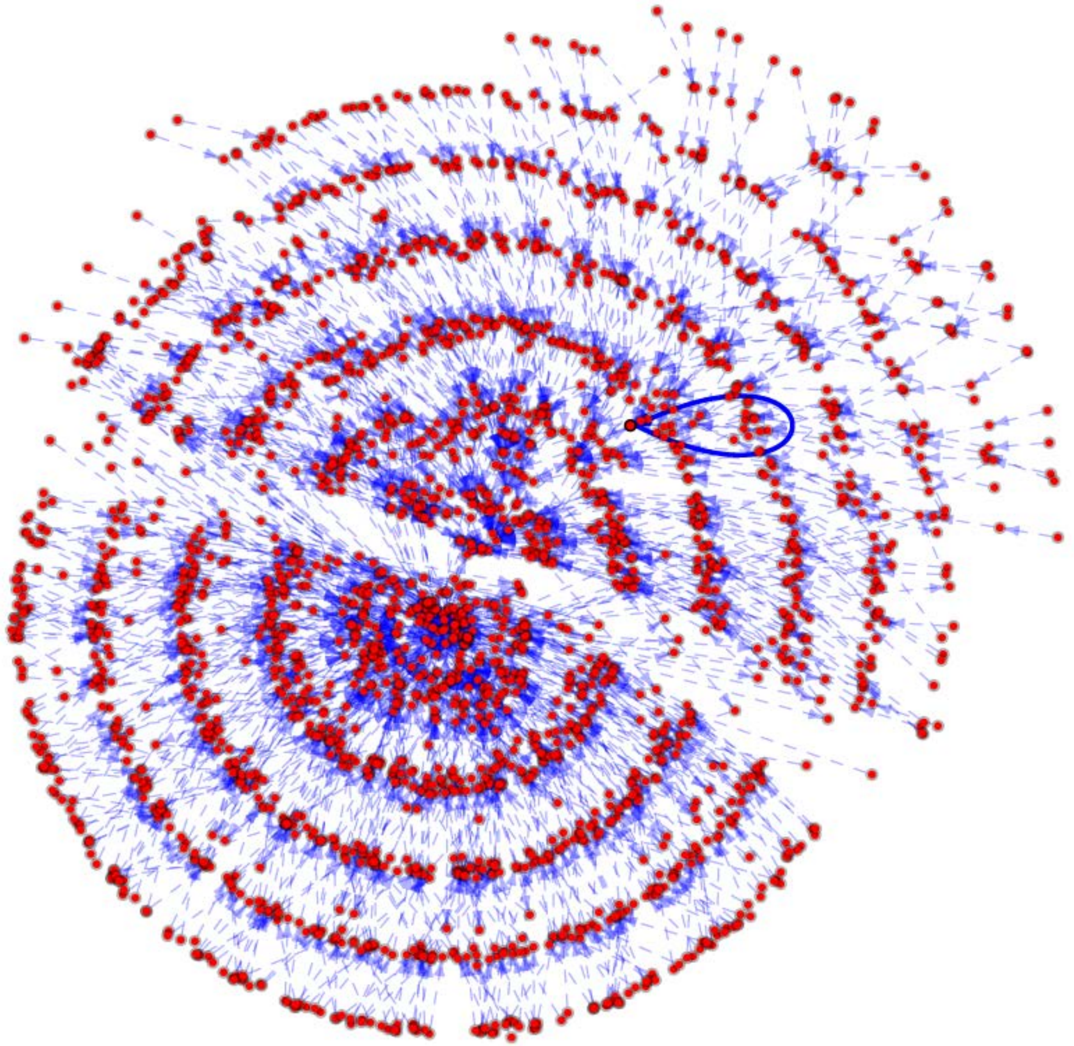


Fig. The state transition graph after control.

[29] J. Zhong, D.W.C. Ho and J. Lu. A New Framework for Pinning Control of Boolean Networks, https://arxiv.org/abs/1912.01411.

Model Reduction

Model Reduction Algorithm

model reduction algorithm [41], shown in **Algorithm 1**.

1) Delete variables that are nonfunctional, i.e., find variables x_i, such that $f_j(\dots, x_i, \dots) = f_j(\dots, \neg x_i, \dots)$ for any $(x_1, \dots, x_{i-1}, x_{i+1}, \dots, x_n) \in \mathcal{D}^{n-1}$.
2) Delete nodes without self-loop. Let x_i be a functional variable in function f_i, which has no self-loop in **WD** $\mathbf{G} = (\mathbf{V}, \mathbf{E})$ of BNs. Find all logical functions f_j dependent on variable x_i, then replace function $f_j(\dots, x_i, \dots)$ by $f_j(\dots, f_i, \dots)$.
3) Use Boolean algebra to simplify function $f_j(\dots, f_i, \dots)$.

Lemma 8 [41]: Consider the original BN (1) and reduced BN (5). Define a projection $\Pi : \{1,0\}^n \to \{1,0\}^k$ as $\Pi(x_1, \dots, x_n) = (x_1, \dots, x_k)$. Thus, the projection Π is a bijection between the set of steady states of reduced BN (5) and the set of steady states of original BN (1).

We can design controller based on the reduced Boolean networks

$$\begin{cases} x_1(t+1) = f_1\left(\left[x_j(t)\right]_{j\in \mathbf{N}_1}\right) \\ \dots \\ x_n(t+1) = f_n\left(\left[x_j(t)\right]_{j\in \mathbf{N}_n}\right). \end{cases} \quad (1)$$

$$\begin{cases} x_1(t+1) = \hat{f}_1\left(\left[x_j(t)\right]_{j\in \hat{\mathbf{N}}_1}\right) \\ \dots \\ x_k(t+1) = \hat{f}_k\left(\left[x_j(t)\right]_{j\in \hat{\mathbf{N}}_k}\right). \end{cases} \quad (5)$$

[41] A. Veliz-Cuba, Reduction of Boolean network models, J. Theor. Biol., 289: 167–172, 2011.

Pinning Controllability and Observability

Controllability Criterion

Theorem 1. *The BCN is controllable if the following conditions are both satisfied: 1) its NS diagram is acyclic; and 2) the in-neighbor set of every state vertex is nonempty and only contains single-source generators and channels.*

Observability Criterion

Lemma 3.1 (See [45], [53]): The BN is observable if its NS diagram G is of both Properties P_1 and P_2:

P_1: for each non-directly observable vertex X_i, there exists a distinct vertex X_j satisfying that $\mathcal{N}_{f_j} = \{i\}$.

P_2: for each cycle C composed entirely of non-directly observable vertices, there exists a vertex X_i in C such that X_i is the unique in-neighbor of some other vertex X_j not the part of C, that is, $\mathcal{N}_{f_j} = \{i\}$.

To this end, the *NS diagram* (also called *wiring digraph*) of BN (3) is constructed as an ordered pair $G := (V, E)$. Vertex set V of NS diagram G consists of two parts: state vertex set $\mathscr{X}$ and output vertex set $\mathscr{Y}$, i.e., $V := \mathscr{X} \cup \mathscr{Y}$, where $\mathscr{X} := \{X_1, X_2, \cdots, X_n\}$ and $\mathscr{Y} := \{Y_1, Y_2, \cdots, Y_p\}$. Arc set $E \subseteq V \times V$ is defined as the set of ordered pairs (X_j, X_i) satisfying that $j \in \mathcal{N}_{f_i}$ and ordered pairs (X_k, Y_k) with $k \in [1, p]$.

- Zhu S, Lu J, Zhong J, et al. A novel pinning observability strategy for Boolean networks[J]. arXiv preprint arXiv:1912.02394, 2019.
- Zhu S, Lu J, Azuma S. Pinning Controllability of Boolean Networks: Application to Large-Scale Genetic Networks[J]. arXiv preprint arXiv:2007.00171, 2020.

[45] E. Weiss and M. Margaliot, A polynomial-time algorithm for solving the minimal observability problem in conjunctive Boolean networks, IEEE Transactions on Automatic Control, 64(7): 2727–2736, 2019.
[53] E. Weiss and M. Margaliot, Output selection and observer design for Boolean control networks: A sub-optimal polynomial-complexity algorithm, IEEE Control Systems Letters, 3(1): 210–215, 2019.

Some of our recent work!

1. J. Zhong, D.W.C. Ho and **J. Lu**. A New Framework for Pinning Control of Boolean Networks, *https://arxiv.org/abs/1912.01411.*
2. Zhu S, **Lu J**, Zhong J, et al. A novel pinning observability strategy for Boolean networks[J]. arXiv preprint *arXiv:1912.02394, 2019*
3. **J.Q. Lu***, R.J. Liu, J.G. Lou, and Y. Liu, Pinning Stabilization of Boolean Control Networks via Minimum Number of Controllers, *IEEE Transactions on Cybernetics*, doi:10.1109/TCYB.2019.2944659, in press.
4. B.W. Li, and **J.Q. Lu***, Boolean-Network-Based Approach for the Construction of Filter Generators, *Science China: Information Sciences*, doi:10.1007/s11432-019-2813-7, in press.
5. S.Y. Zhu, **J.Q. Lu***, T.W. Huang, and Y. Liu, The output robustness of probabilistic Boolean control networks with respect to one-bit perturbation, *IEEE Transactions on Control of Network Systems*, doi:10.1109/TCNS.2020.2999456, in press, 2020.
6. S.Y. Zhu, **J.Q. Lu***, and Y. Liu. Asymptotical Stability of Probabilistic Boolean Networks with State Delays. *IEEE Transactions on Automatic Control*, 65(4): 1779-1784, April 2020.
7. B.W. Li, **J.Q. Lu***, Y. Liu, and Z.G. Wu. The outputs robustness of Boolean control networks via pinning control. *IEEE Transactions on Control of Network Systems*, 7(1):201-209, March 2020.
8. S.Y. Zhu, **J.Q. Lu***, and D.W.C. Ho. Finite-Time Stability of Probabilistic Logical Networks: A Topological Sorting Approach. *IEEE Transactions on Circuits and Systems II: Express Briefs*, 67(4):695-699, 2020.
9. R.J. Liu, **J.Q. Lu***, W.X. Zheng, and J. Kurths. Output feedback control for set stabilization of Boolean control networks. *IEEE Transactions on Neural Networks & Learning Systems*, 31(6): 2129-2139, 2020.
10. B.W. Li, **J.Q. Lu***, J. Zhong, Y. Liu. Fast-time stability of temporal Boolean networks. *IEEE Transactions on Neural Networks and Learning Systems*, 30(8): 2285-2294, August 2019.

Some of our recent work!

11. S.Y. Zhu, **J.Q. Lu***, Y. Liu, T.W. Huang, and J. Kurths. Output Tracking of Probabilistic Boolean Networks by Output Feedback Control. *Information Sciences*, 483:96-105, 2019.
12. **J.Q. Lu***, L.J. Sun, Y. Liu, D.W.C. Ho, J.D. Cao. Stabilization of Boolean control networks under aperiodic sampled-data control. *SIAM Journal on Control and Optimization*, 56(6):4385-4404, 2018.
13. **J.Q. Lu***, M.L. Li, T.W. Huang, Y. Liu, J.D. Cao. The transformation between the Galois NLFSRs and the Fibonacci NLFSRs via semi-tensor product of matrices. *Automatica*, 96:393-397, 2018.
14. **J.Q. Lu***, M.L. Li, Y. Liu, D.W.C. Ho, J. Kurths. Nonsingularity of Grain-like cascade FSRs via semi-tensor product. *Science China: Information Sciences*, 61(1):010204, 2018.
15. R.J. Liu, **J.Q. Lu***, Y. Liu, J.D. Cao, and Z.G. Wu. Delayed Feedback Control for Stabilization of Boolean Control Networks with State Delay. *IEEE Transactions on Neural Networks and Learning Systems*, 29(7):3283-3288, 2018.
16. **J.Q. Lu***, H.T. Li, Y. Liu and F.F. Li. Survey on semi-tensor product method with its applications in logical networks and other finite-valued systems. *IET Control Theory & Applications*, 11(13): 2040-2047, 2017.
17. J. Zhong, **J.Q. Lu***, T.W. Huang, D.W.C. Ho. Controllability and Synchronization Analysis of Identical-Hierarchy Mixed-valued Logical Control Networks. *IEEE Trans. Cybernetics*, 47(11): 3482-3493, 2017.
18. **J.Q. Lu***, J. Zhong, D.W.C. Ho, Y. Tang and J.D. Cao. On controllability of delayed Boolean control networks. *SIAM Journal on Control and Optimization*, 54(2): 475-494, 2016.
19. **J.Q. Lu***, J. Zhong, C. Huang and J.D. Cao. On pinning controllability of Boolean control networks. *IEEE Transactions on Automatic Control*, 61(6):1658-1663, 2016.
20. J. Zhong, **J.Q. Lu***, Y. Liu and J.D. Cao. Synchronization in an array of output-coupled Boolean networks with time delay. *IEEE Transactions on Neural Networks and Learning Systems*, 25(12): 2288-2294, 2014.

Thanks for Attention!