Local Stabilization of Multi-valued Logical Control Networks via State Feedback

Hui Tian[1,2], Huaguang Zhang[1], Zhanshan Wang[1], Yanfang Hou[3]

1. School of Information Science and Engineering, Northeastern University, Shenyang, 110819
E-mail: tianhui1980@hpu.edu.cn
E-mail: zhanghuaguang@mail.neu.edu.cn
E-mail: wangzhanshan@ise.neu.edu.cn

2. School of Mathematics and Information Science, Henan Polytechnic University, Jiaozuo 454000, China

3. School of Computer Science and Technology, Henan Polytechnic University, Jiaozuo 454000, China
E-mail: yfhou@hpu.edu.cn

Abstract: A more general stabilization problem of k-valued logical control networks (LCNs) is presented and a reverse-transfer (RT) method is proposed to investigate this problem. Firstly, the RT method is introduced and the concept of stabilizability of k-valued LCNs is generalized. Secondly, using the RT method, some equivalent criteria are proved for the largest stability domain of a fixed point of a k-valued LCN under a constant control. Then some necessary and sufficient local/global stabilizability conditions are derived directly. Based on the local stabilizability condition, a constructive method of designing state feedback stabilizers for k-valued LCNs is provided. Thirdly, according to the results obtained above, some algorithms are developed. Finally, an example is given to show the significance of our proposed problem and the effectiveness of our results.

Key Words: State Feedback stabilization, k-valued logical control network (LCN), Reverse-transfer (RT), Semi-tensor product of matrices (STP).

1 INTRODUCTION

Boolean network (BN), which is of two possible states 1 and 0, has received more and more attention of researchers from many fields, such as biology, physics, system science and so on. Some research indicates that many realistic biological questions may be answered within the seemingly simplistic Boolean formalism [1].

However, when node expression is not limited to 1 or 0, it is not reasonable to take advantage of a BN since binary values may lose the precision. In this case, a k-valued logical network (LN), which is an extension of the traditional BN, should be adopted to establish the mathematics model more precisely. Compared with BN, the k-valued LN has the similar topological structure, while it allows its nodes to take values from a finite set $\mathscr{D}_k = \{\frac{i}{k-1} | i = 0, 1, \cdots, k-1\}$. k-valued LNs can be found in some other complex systems, for example chemical reactions [2], cognitive sciences [3], etc. Besides, the k-valued LN is an effective tool for solving some problems in game theory [4], such as the infinitely repeated prisoners' dilemma. Particularly, recently, much attention is put on the networked evolutionary game that is usually in the form of multi-valued logical network [5, 6].

This work was supported in part by NNSF of China (Nos. 61433004, 61473070), in part by the Fundamental Research Funds for the Central Universities (Nos. N130104001, N130504002), in part by the Scientific Research Fund of Henan Provincial Education Department (14A110004), and in part by the State Key Laboratory of Integrated Automation for Process Industries Fundamental Research Funds (Nos. 2013ZCX01, 2013ZCX05).

Naturally, various control problems of k-valued logical control networks (LCNs) need to be considered, which may help us to gain insights into the intrinsic control in complex systems and enable us to develop strategies to manipulate complex systems by using exogenous inputs. However, as we know, it is difficult to deal with the k-valued LN, including BN as its special case, due to the shortage of efficient tools until the introduction of the semi-tensor product (STP) developed in the last decade [7, 8, 9]. Using this mathematical tool, much significant and substantial progress has been made, see [10]-[18], etc. Among them, [10] investigated the global stability and stabilization of BNs and gave some necessary and sufficient conditions. Then the study was extended to switched Boolean networks [11, 12]. However, no design approach to any state feedback stabilizer was available in these papers. Thus, [13] studied the design problem of state feedback stabilizer and provided an effective design method of global state feedback stabilizer for BCNs. Later, based on the results in [13], Li et al presented a method of designing global output feedback stabilizers for BCNs [14]. It should be pointed out that most of the existing results mainly focus on the case of global stabilization. However, global stabilization is not always required. In some cases, it is enough to stabilize the scope we are interested in. For example, in the context of gene therapy applications, the researchers are only interested in whether a gene regulatory network can be driven to a desirable location (corresponding to a healthy state) from a given undesirable one (corresponding to a diseased state). So the local stabilization of logic systems deserves more attention. In this respect, some work has been done in [15], but the relevant results are s bit special and no method for designing state feedback stabilizer has been provided either. To the best of our knowledge, there has not been any general design method of local state feedback stabilizers for k-valued LCNs in the literature yet. Motivated by the above analysis, we generalize the stabilizability concept of k-valued LCNs and propose a new method of investigating the stabilization problem. By using this method, some interesting results are derived. The main contributions are summarized as follows.

1. Some equivalent criteria are proposed to calculate the largest stability domain of a fixed point for a k-valued LCN and a corresponding algorithm is provided.
2. Some necessary and sufficient local/global stabilizability conditions are derived and an algorithm is developed to check whether a set of given initial states Ω can be stabilized to a particular state. Although Theorem 3.5 in [15] has also given a local stabilizability condition for k-valued LCNs, it is only sufficient, and only applicable to the case when Ω includes one state.
3. A constructive method of designing feasible state feedback stabilizers for k-valued LCNs is proved and a corresponding algorithm is presented.

The rest of this paper is organized as follows. Section 2 introduces some necessary preliminaries and gives the problem formulation. The main results are presented in Section 3. An illustrative example is provided in Section 4, which is followed by a conclusion in Section 5.

2 PRELIMINARIES AND PROBLEM FORMULATION

We introduce some notations first.

- $\mathbf{1}_n$ (respectively, $\mathbf{0}_n$) denotes the n-dimensional row vector whose entries are equal to 1 (respectively, 0). That is, $\mathbf{1}_n := [\underbrace{1\ 1 \cdots 1}_{n}]$.
- $\Delta_n := \{\delta_n^i \mid i = 1,2,\cdots,n\}$, where δ_n^i is the ith column of the $n \times n$ identity matrix.
- $\delta_n[i_1,i_2,\cdots,i_s] := [\delta_n^{i_1}\ \delta_n^{i_2}\ \cdots\ \delta_n^{i_s}]$, called a logical matrix.
- $\mathscr{L}_{m\times n}$ is the set of all $m \times n$ logical matrices. In particular, $\mathscr{L}_{1\times n} := \mathscr{B}_{1\times n}$, called a Boolean vector.
- $Row_i(L)$ and $Col_i(L)$ denote the ith row and the ith column of a matrix L, respectively.
- $X \le Y := x_i \le y_i$ for all $1 \le i \le n$, where $X = (x_1,x_2,\cdots,x_n)$, $Y = (y_1,y_2,\cdots,y_n)$.
- $\sigma[V]$ denotes the number of all elements of a set V or the total number of positive entries of a vector V.
- $[V]_i$ is the ith entry of a vector V.

Our research object is the k-valued LCN (1) with a state feedback control (2), whose dynamics is described as follows:

$$x_i(t+1) = f_i(x_1(t),\cdots,x_n(t),u_1(t),\cdots,u_m(t)),\quad i = 1,2,\cdots,n; \tag{1}$$

$$u_j(t) = h_j(x_1(t),\cdots,x_n(t)),\quad j = 1,2,\cdots,m, \tag{2}$$

where x_i, $u_j \in \mathscr{D}_k = \{\frac{i}{k-1} | i = 0,1,\cdots,k-1\}$; x_i are state variables; u_j are controls; f_i and h_j are logical functions.
The main mathematical tool used in this paper is the semi-tensor product of matrices (STP), which is a generation of conventional matrix product. Now we identify $\frac{k-i}{k-1}$ with δ_k^i. Thus the space $\mathscr{D}_k$ is equivalent to Δ_k. Denoting $x(t) = \ltimes_{i=1}^n x_i(t) \in \Delta_{k^n}$, $u(t) = \ltimes_{i=1}^m u_i(t) \in \Delta_{k^m}$ and referring to [16], there exists a unique matrix $\bar{L} \in \mathscr{L}_{k^n\times k^{n+m}}$, such that the algebraic form of (1) is

$$x(t+1) = \bar{L}u(t)x(t). \tag{3}$$

Similarly, there exists a unique matrix $H \in \mathscr{L}_{k^m\times k^n}$, such that the algebraic form of (2) is

$$u(t) = Hx(t). \tag{4}$$

In this paper, we only investigate models (3) and (4) to highlight our methods.
Denote the set $\{x_0 \mid x_d = \bar{L}\delta_{k^m}^r x_0\}$ by $\mho_r^1(x_d)$, where x_d and $\delta_{k^m}^r$ are the given state and constant control, respectively. In the following, we give two new concepts that play a key role in our discussion.

Definition 1

1. *The combination $(\mho_r^1(x_d), x_d, \delta_{k^m}^r)$, denoted by $R_r^1(x_d)$, is called the first RT of x_d under a RT control $v = \delta_{k^m}^r$.*
2. *Let $\mho_{r_1,\cdots,r_q}^q(x_d) = \{x_{i_1},\cdots,x_{i_s}\}$, where $q \ge 1$ is an integer. Given a RT control $v = \delta_{k^m}^{r_{q+1}}$, the combination $(\mho_{r_1,\cdots,r_q,r_{q+1}}^{q+1}(x_d), x_d, \{\delta_{k^m}^{r_{t+1}} | 0 \le t \le q\})$, denoted by $R_{r_1,\cdots,r_q,r_{q+1}}^{q+1}(x_d)$, is called the $(q+1)$th RT of x_d under the RT control path $v(t) = \delta_{k^m}^{r_{t+1}}$, $0 \le t \le q$, where $\mho_{r_1,\cdots,r_q,r_{q+1}}^{q+1}(x_d) = \bigcup_{j=1}^{s} \mho_{r_{q+1}}^1(x_{i_j})$.*

Definition 2 *$R_{r_1,\cdots,r_q}^q(x_d)$ is called a fine qth RT of x_d if there is no any qth RT $R_{\bar{r}_1,\cdots,\bar{r}_q}^q(x_d)$ satisfying $\mho_{r_1,\cdots,r_q}^q(x_d) \subsetneq \mho_{\bar{r}_1,\cdots,\bar{r}_q}^q(x_d)$.*

We denote

$$G^q(x_d) = \{\mho_{r_1,\cdots,r_q}^q(x_d) | R_{r_1,\cdots,r_q}^q(x_d) \text{ is a fine } q\text{th RT}\}.$$

The following lemma is straightforward and thus omitted.

Lemma 1 *For the system (3), all initial states $x(0) \in \mho_{r_1,\cdots,r_q}^q(x_d)$ can be driven to x_d by the input sequence $u(0) = \delta_{k^m}^{r_q}$, $u(1) = \delta_{k^m}^{r_{q-1}},\cdots,u(q-1) = \delta_{k^m}^{r_1}$. That is, $x_d = \bar{L}\delta_{k^m}^{r_1}\cdots\bar{L}\delta_{k^m}^{r_{q-1}}\bar{L}\delta_{k^m}^{r_q}x(0)$.*

Finally, we give the definition of stabilizability that is a natural generalization of Definition 1 of [13] and Definition 3.2 of [15].

Definition 3 *The system (3) is said to be locally (globally) stabilizable to a state $x_d \in \Delta_{k^n}$ with a stability domain Ω (when $\Omega = \Delta_{k^n}$), if for every initial state $x_0 \in \Omega$, there exists an input sequence $u(t), t = 0, 1, \cdots$ and a positive integer N such that $t \geq N$ implies $x(t, x_0, u(t)) = x_d$.*

Remark 1 *If the system (3) is stabilizable to x_d, the state x_d must be a fixed point of (3) under a constant input.*

Remark 2 *When Ω includes only one state, our Definition 4 is actually the definition of local stabilizability in [15]. Therefore, the stabilization problem we discuss in this paper is more general.*

The goal of this paper is to provide a criterion for checking whether a k-valued LCN (3) starting from a given set Ω can be stabilized to a particular state and develop a design approach to a feasible state feedback stabilizer (4) when local stabilizability is guaranteed.

3 MAIN RESULTS

3.1 Stabilizability Criteria and Stabilization Design

For convenience, the states in Δ_{k^n} are denoted as $P_1 = \delta_{k^n}^1$, $P_2 = \delta_{k^n}^2, \cdots$, $P_{k^n} = \delta_{k^n}^{k^n}$ and we denote $\Gamma_h^{(s,i)} = \bigcup_{\mho^s_{r_1,\cdots,r_h,\cdots,r_s}(P_i) \in G^s(P_i)} \mho^h_{r_1,\cdots,r_h}(P_i), 1 \leq h \leq s$ and $\Gamma_s^{(s,i)} = \bigcup^{(s,i)}$. The following proposition always holds because the number of different states in the state space is finite.

Proposition 1 *Assume that P_i is a fixed point of (3) under a constant control $u = \delta_{k^m}^{r^*}$. Then the following statements hold.*

(a) For any $\mho^s_{r_1,\cdots,r_s}(P_i) \in G^s(P_i)$, there exists a $\mho^{s+1}_{\bar{r}_1,\cdots,\bar{r}_{s+1}}(P_i) \in G^{s+1}(P_i)$, such that $\mho^s_{r_1,\cdots,r_s}(P_i) \subseteq \mho^{s+1}_{\bar{r}_1,\cdots,\bar{r}_{s+1}}(P_i)$.

(b) $\bigcup^{(s,i)} \subseteq \bigcup^{(s+1,i)}$ holds for any positive integer s.

(c) There is a positive integer $s \leq k^n$ such that $\bigcup^{(s,i)} = \bigcup^{(s+1,i)}$.

(d) If $\bigcup^{(s,i)} = \bigcup^{(s+1,i)}$ holds, then $\bigcup^{(s,i)} = \bigcup^{(s+r,i)}$ also holds for any $r \geq 2$.

Remark 3 *Assume that P_i is a fixed point of (3). From items (b), (c) and (d) in Proposition 1, there must exist a positive integer $s \leq k^n$ such that*

$$\bigcup^{(1,i)} \subset \bigcup^{(2,i)} \subset \cdots \subset \bigcup^{(s,i)} = \bigcup^{(s+1,i)} = \cdots. \quad (5)$$

In the following, we present the first main theorem that is very helpful for solving the largest stability domain of system (3) associated with a given fixed point.

Theorem 1 *For a fixed point P_i of (3) under a constant control, the following statements are equivalent.*

(a) $\bigcup^{(s,i)} = \bigcup^{(s+1,i)}$.

(b) $\sigma[\bigcup^{(s,i)}] = \sigma[\bigcup^{(s+1,i)}]$.

(c) $\bigcup^{(s,i)}$ is the largest scope of P_i. That is, for any point $P_j \in \Delta_{k^n}$, if $P_j \in \bigcup^{(s,i)}$, there exists such a control sequence that leads P_j to P_i and makes it stay at P_i forever, otherwise P_j can not be stabilized to P_i.

Proof. From (5), the proof of (a) $\leftrightarrow$ (b) is trivial.
Now we prove (a) $\leftrightarrow$ (c). As we know, if a point P_j is stabilizable to the fixed point P_i, then there exists an input sequence, say $u(t) = \delta_{k^m}^{r_{q-t}}$, $0 \leq t \leq q$ such that $P_i = \bar{L}\delta_{k^m}^{r_1} \cdots \bar{L}\delta_{k^m}^{r_{q-1}} \bar{L}\delta_{k^m}^{r_q} P_j$, which implies $P_j \in \mho^q_{r_1,\cdots,r_q}(P_i)$ and then $P_j \in \bigcup^{(q,i)}$. Therefore, for any P_j in the stability domain of P_i, there is a $\bigcup^{(q,i)}$ such that $P_j \in \bigcup^{(q,i)}$. Combining this fact with (5) implies that (a) $\to$ (c) holds. As for (c) $\to$ (a), it can be derived from Lemma 1 and item (b) in Proposition 1. Summarizing the above argument, we have (a) $\leftrightarrow$ (c).
According to the transitivity property of equivalence relationship, we have (b) $\leftrightarrow$ (c). □

Remark 4 *Note that item (c) in Proposition 1 ensures the existence of the largest stability domain of a fixed point. Furthermore, Theorem 1 provides some methods for solving such a stability domain, which will be further discussed in Subsection 3.2.*

By using Proposition 1 and Theorem 1, we give a stabilizability condition for the system (3).

Theorem 2 *Let P_i be a fixed point of (3) under a constant control and Ω be a nonempty set. The system (3) is locally stabilizable to P_i with the stability domain Ω if and only if there is a positive integer $s \leq k^n$ such that $\Omega \subseteq \bigcup^{(s,i)}$. In particular, the system (3) is globally stabilizable to P_i if and only if there is a positive integer $s \leq k^n$ such that $\bigcup^{(s,i)} = \Delta_{k^n}$ or $\sigma[\bigcup^{(s,i)}] = k^n$.*

Now we provides a constructive design approach to state feedback stabilizer for the system (3).

Theorem 3 *Assume that P_i is a fixed point of (3) under a constant control and $\Omega \subseteq \bigcup^{(s,i)}$ holds for an integer s. Then all trajectories of the system (3), starting from Ω, can be stabilized to P_i by any state feedback controller (4) with H designed as follows. For any $Col_j(H)$, if $P_j \in \bigcup^{(s,i)}$, then find a fine sth RT, say $R^s_{r_1,\cdots,r_h,\cdots,r_s}(P_i)$, satisfying $P_j \in (\mho^h_{r_1,r_2,\cdots,r_h}(P_i) \setminus \bigcup_{q=1}^{h-1} \Gamma_q^{(s,i)})$ for a certain integer h, where $\bigcup_{q=1}^{0} \Gamma_q^{(s,i)} = \emptyset$, and then choose $Col_j(H)$ as $\delta_{k^m}^{r_h}$; otherwise, $Col_j(H)$ can be chosen arbitrarily.*

Proof. Without loss of generality, assume that $\bigcup^{(s,i)}$ satisfies $\bigcup^{(h,i)} \subset \bigcup^{(s,i)}$ for any positive integer $h < s$. We write $\bigcup^{(s,i)}$ as the union of s disjoint sets

$$\begin{aligned}\bigcup^{(s,i)} =& \Gamma_1^{(s,i)} \cup (\Gamma_2^{(s,i)} \setminus \Gamma_1^{(s,i)}) \cup \cdots \cup (\Gamma_h^{(s,i)} \setminus \bigcup_{q=1}^{h-1} \Gamma_q^{(s,i)}) \\ & \cup \cdots \cup (\Gamma_s^{(s,i)} \setminus \bigcup_{q=1}^{s-1} \Gamma_q^{(s,i)}).\end{aligned} \quad (6)$$

Clearly, for any initial state $x(0) \in \Omega$, say P_j, we have $P_j \in \bigcup^{(s,i)}$. Then there exists a unique positive integer $1 \le h \le s$, such that $P_j \in \Gamma_h^{(s,i)} \setminus \bigcup_{q=1}^{h-1} \Gamma_q^{(s,i)}$, where $\bigcup_{q=1}^{0} \Gamma_q^{(s,i)} = \emptyset$. From the definition of $\Gamma_h^{(s,i)}$, there must be at least a fine sth RT, say $R^s_{r_1,\cdots,r_h,\cdots,r_s}(P_i)$, satisfying $P_j \in \mho^h_{r_1,\cdots,r_h}(P_i)$. So $P_j \in (\mho^h_{r_1,\cdots,r_h}(P_i) \setminus \bigcup_{q=1}^{h-1} \Gamma_q^{(s,i)})$. According to the design approach given in Theorem 3, $\delta_{k^m}^{r_h}$ is taken as $Col_j(H)$. Note that from Lemma 1 the trajectory of (3) starting from P_j can be driven into $\mho^{h-1}_{r_1,\cdots,r_{h-1}}(P_i)$ by the constant control $u = \delta_{k^m}^{r_h}$ and this control can be obtained by $H\delta_{k^n}^j$. Therefore, P_j is driven into $\mho^{h-1}_{r_1,\cdots,r_{h-1}}(P_i)$ and then into $\Gamma_{h-1}^{(s,i)}$ by the state feedback controller (4) with H designed as in Theorem 3. We notice that h is finite and $P_i \in \Gamma_1^{(s,i)}$. We can use the same technique developed above to show that P_j will be driven into $\Gamma_1^{(s,i)}$ and then eventually be stabilized at P_i by controller (4) with H designed as in Theorem 3. The arbitrariness of choosing initial state $x(0)$ shows the effectiveness of our design approach. □

In the following, we are ready to give matrix forms of the above results. First, we split the transition matrix $\bar{L}$ into k^m square blocks as $\bar{L} = [L_1 \ \cdots \ L_r \ \cdots \ L_{k^m}]$, where $L_r \in \mathscr{L}_{k^n \times k^n}, r = 1, 2, \cdots, k^m$.

Proposition 2 *For this system (3), $P_j \in \mho^q_{r_1,r_2,\cdots,r_q}(P_i)$ if and only if $[Row_i(L_{r_1})L_{r_2} \cdots L_{r_q}]_j = 1$.*

Proof. According to Definition 1, $P_j \in \mho^q_{r_1,\cdots,r_q}(P_i)$ if and only if $\bar{L}\delta_{k^m}^{r_1} \cdots \bar{L}\delta_{k^m}^{r_q} P_j = P_i$ or $Row_i(L_{r_1} \cdots L_{r_q} P_j) = 1$, equivalent to $[Row_i(L_{r_1}) \cdots L_{r_q}]_j = 1$. □

From Proposition 2 (and Lemma 1), we have the following two corollaries.

Corollary 1 *For this system (3), $P_j \in (\mho^s_{r_1,r_2,\cdots,r_s}(P_i) \cup \mho^{\bar{s}}_{\bar{r}_1,\bar{r}_2,\cdots,\bar{r}_{\bar{s}}}(P_i))$ if and only if $[Row_i(L_{r_1})L_{r_2} \cdots L_{r_s} + Row_i(L_{\bar{r}_1})L_{\bar{r}_2} \cdots L_{\bar{r}_{\bar{s}}}]_j \ge 1$.*

Corollary 2 *If $[Row_i(L_{r_1})L_{r_2} \cdots L_{r_s}]_j = 1$, the state P_j can be driven to P_i by the input sequence $u(0) = \delta_{k^m}^{r_s}$, $u(1) = \delta_{k^m}^{r_{s-1}}, \cdots, u(s-1) = \delta_{k^m}^{r_1}$.*

Based on Proposition 2, $\mho^q_{r_1,r_2,\cdots,r_q}(P_i)$ is identified with $Row_i(L_{r_1})L_{r_2} \cdots L_{r_q}$. Even we identify a general set $\Omega \subseteq \Delta_{k^n}$ with a k^n-dimensional Boolean vector. Precisely speaking, $P_j \in \Omega$ if and only if the jth entry of the vector form of Ω equals to 1. We utilize the same notation Ω to express its vector form and use them alternatively without explanation. Denote $\Upsilon_h^{(s,i)} = \sum_{\mho^s_{r_1,\cdots,r_h,\cdots,r_s}(P_i) \in G^s(P_i)} Row_i(L_{r_1})L_{r_2} \cdots L_{r_h}$, $1 \le h \le s$ and $\Upsilon_s^{(s,i)} = \Sigma^{(s,i)}$. Then by using Proposition 2 and Corollaries 1 and 2, we rewrite Theorems 1, 2 and 3 as follows.

Theorem 4 *For a fixed point P_i of (3) under a constant control, the following statements are equivalent.*
(a) For any positive integer $j \le k^n$, $[\Sigma^{(s,i)}]_j > 0$ if and only if $[\Sigma^{(s+1,i)}]_j > 0$.
(b) $\sigma[\Sigma^{(s,i)}] = \sigma[\Sigma^{(s+1,i)}]$ holds.
(c) $\bigcup^{(s,i)}$ is the largest scope of P_i.

Theorem 5 *Let P_i be a fixed point of (3) under a constant control and Ω be a nonempty set. The system (3) is locally stabilizable to P_i with the stability domain Ω if and only if there is a positive integer $s \le k^n$ such that $\Omega \le \Sigma^{(s,i)}$. In particular, the system (3) is globally stabilizable to P_i if and only if there is a positive integer $s \le k^n$ such that $\mathbf{1}_{k^n} \le \Sigma^{(s,i)}$ or $\sigma[\Sigma^{(s,i)}] = k^n$.*

Theorem 6 *Assume that P_i is a fixed point of (3) under a constant control and $\Omega \le \Sigma^{(s,i)}$ holds for an integer s. Then all trajectories of the system (3), starting from Ω, can be stabilized to P_i by any state feedback controller (4) with H designed as follows. For any $Col_j(H)$, if $[\Sigma^{(s,i)}]_j > 0$, then find a fine sth RT, say $Row_i(L_{r_1})L_{r_2} \cdots L_{r_h} \cdots L_{r_s}$, satisfying $[Row_i(L_{r_1})L_{r_2} \cdots L_{r_h} - \sum_{q=1}^{h-1} \Upsilon_q^{(s,i)}]_j > 0$ for a certain integer h, where $\sum_{q=1}^{0} \Upsilon_q^{(s,i)} = \mathbf{0}_{k^n}$, and then choose $Col_j(H)$ as $\delta_{k^m}^{r_h}$; otherwise, $Col_j(H)$ can be chosen arbitrarily.*

3.2 Algorithms

Theorems 4, 5 and 6 are effective tools for solving the stabilization problem of the system (3), but the difficulty in using them is to calculate $G^s(P_i)$. To calculate $G^s(P_i)$ as soon as possible, we observe that if $Row_i(L_{r_1})L_{r_2} \cdots L_{r_q} \le Row_i(L_{\bar{r}_1})L_{\bar{r}_2} \cdots L_{\bar{r}_q}$ holds, then $R^q_{r_1,r_2,\cdots,r_q}(P_i)$ can be deleted because it will not lead to any change of $G^s(P_i)$. For statement ease, both $R^q_{r_1,r_2,\cdots,r_q}(P_i)$ and $\mho^s_{r_1,r_2,\cdots,r_s}(P_i)$ are briefly expressed as $Row_i(L_{r_1})L_{r_2} \cdots L_{r_q}$ if no confusion arises. $\bigcup^{(s,i)}$ is identified with $\Sigma^{(s,i)}$. That is, $P_j \in \bigcup^{(s,i)}$ if and only if $[\Sigma^{(s,i)}]_j > 0$.

Algorithm 1 *The largest stability domain of a fixed point P_i of the system (3) under a constant control can be calculated by the following steps.*

- Step 1. *Set $q = \overline{q} = max\{\sigma[Row_i(L_r)] | r = 1, \cdots, k^m\}$ and $\underline{q} = min\{\sigma[Row_i(L_r)] | r = 1, \cdots, k^m\}$.*
- Step 2. *If $q < \underline{q}$, go to Step 5. Else, denote $\Phi_q = \{Row_i(L_r) | \sigma[Row_i(L_r)] = q; r = 1, \cdots, k^m\}$ and go to Step 3.*
- Step 3. *Check whether all RTs have been checked. If yes, go to Step 4. Else, choose an unchecked RT $Row_i(L_{r_1})$ with $\sigma[Row_i(L_{r_1})] \le q$. If there exists a RT $Row_i(L_{r_2})$ in Φ_q, different from $Row_i(L_{r_1})$, such that $Row_i(L_{r_1}) \le Row_i(L_{r_2})$, then $Row_i(L_{r_1})$ is deleted. In this case, replace Φ_q by $\Phi_q := \Phi_q \setminus \{Row_i(L_{r_1})\}$ and go back to Step 3. Else, $Row_i(L_{r_1})$ keeps unchanged and go back to Step 3.*
- Step 4. *Replace q by $q =: q - 1$. If $\Phi_q = \emptyset$, go back to Step 4. Else, set $\underline{q} = min\{\sigma[V] | V$ is an unchanged RT in Step 3$\}$ and go back to Step 2.*

- Step 5. *Set $s=1$. Denote $G^s(P_i)=\bigcup_{\Phi_j\neq\emptyset}\Phi_j$ and calculate $\Sigma^{(s,i)}=\sum_{V\in G^s(P_i)}V$. Check whether the equality $\sigma[\Sigma^{(s-1,i)}]=\sigma[\Sigma^{(s,i)}]$ holds. If yes, stop, $\Sigma^{(s,i)}$ is the largest stabilization domain of P_i. Else, we branch from $G^s(P_i)$. Precisely speaking, for each vector of $G^s(P_i)$, say $Row_i(L_{r_1})L_{r_2}\cdots L_{r_s}$, calculate $Row_i(L_{r_1})L_{r_2}\cdots L_{r_s}L_r, r=1,2,\cdots,k^m$. Replace all old RTs by these new RTs and s by $s:=s+1$. Go back to Step 1.*

From Theorem 5, we can also use Algorithm 1 to test the stabilizability of (3). Next, according to Theorem 6, we present an algorithm for designing feasible stabilizers.

Algorithm 2 *Assume that P_i is a fixed point of (3) under a constant control and $\Omega\leq\Sigma^{(s,i)}$ holds for a positive integer s. Let $G^s(P_i)=\{Row_i(L_{r_1^{(q)}})L_{r_2^{(q)}}\cdots L_{r_h^{(q)}}\mid 1\leq q\leq l\}$. Then all trajectories of the system (3), starting from Ω, can be stabilized to P_i by any state feedback controller (4) with H designed as follows.*

- Step 1. *Set $q=1$ and $h=1$. For any integer j satisfying $[Row_i(L_{r_1^{(q)}})L_{r_2^{(q)}}\cdots L_{r_h^{(q)}}-\sum_{q=1}^{h-1}\Upsilon_q^{(s,i)}]_j>0$, if $Col_j(H)$ has not been solved yet, then $Col_j(H)$ is chosen as $\delta_{2^m}^{r_h}$, else $Col_j(H)$ keeps unchanged .*
- Step 2. *Set $h=h+1$. If $h\leq s$, go back Step 1, else go to Step 3.*
- Step 3. *Set $q=q+1$. If $q\leq l$, go back Step 1, else go to Step 4.*
- Step 4. *The other columns of H can be chosen arbitrarily.*

4 AN ILLUSTRATIVE EXAMPLE

In this section, we give an example to show the significance of local stabilization and the effectiveness of our methods.

Example 1 *Consider a game (S_4,G,Π)[5], where $S_4=(N,E)$ is a network, $N=\{w,x_1,x_2,x_3\}$ is the set of nodes and E is the set of edges; G is Benoit-Krishna Game with strategies {1: Deny, 2: Waffle, 3: Confess}; Π is the unconditional imitation with fixed priority, which provides a strategy updating rule. This model can be explained as a game of four players w, $x_i, i=1,2,3$. The player x_2 is the head who contacts only x_1 and x_3. The player w is a detective who sneaked in and can also contact only x_1 and x_3. The purpose of w is to make all x_i confess.*

The strategy i is identified with δ_3^i. Under the vector forms of logical variables, we define $x(t)=x_1(t)x_2(t)x_3(t)$, $u(t)=w(t)$. We calculate by the technique provided by [5] that $x(t+1)=\bar{L}u(t)x(t)$ with

$$\begin{aligned}\bar{L}=\delta_{27}[&1\ 1\ 9\ 1\ 1\ 9\ 27\ 27\ 27\ 1\ 1\ 9\ 1\ 14\ 18\ 27\ 9\ 27\ 25\ 25\\ &27\ 25\ 26\ 27\ 27\ 27\ 27\ 1\ 1\ 9\ 1\ 5\ 3\ 27\ 27\ 27\ 1\ 11\\ &18\ 13\ 14\ 14\ 27\ 14\ 14\ 25\ 26\ 27\ 25\ 14\ 17\ 27\ 14\\ &27\ 21\ 21\ 27\ 21\ 24\ 27\ 27\ 27\ 27\ 21\ 19\ 27\ 24\ 14\\ &14\ 27\ 24\ 27\ 27\ 27\ 27\ 27\ 14\ 27\ 27\ 27\ 27].\end{aligned}$$

Clearly, the target state is $x_d=\delta_{27}^{27}$, which is a fixed point under any constant control δ_3^r. Now there are two problems of interest: (1) What is the largest stability domain of x_d? (2) How do we design a state feedback controller to stabilize all trajectories of the game starting from the largest stability domain to x_d? We are ready to deal with them by using Algorithms 1 and 3.

First of all, from $\bar{L}$ we get the first RTs of x_d as follows:

$$\begin{aligned}Row_{27}(L_1)=[&0\ 0\ 0\ 0\ 0\ 0\ 1\ 1\ 1\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 1\ 0\ 1\\ &0\ 0\ 1\ 0\ 0\ 1\ 1\ 1\ 1],\end{aligned}$$

$$\sigma[Row_{27}(L_1)]=10,$$

$$\begin{aligned}Row_{27}(L_2)=[&0\ 0\ 0\ 0\ 0\ 0\ 1\ 1\ 1\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 1\ 0\ 0\\ &0\ 0\ 1\ 0\ 0\ 0\ 1\ 0\ 1],\end{aligned}$$

$$\sigma[Row_{27}(L_2)]=7,$$

$$\begin{aligned}Row_{27}(L_3)=[&0\ 0\ 1\ 0\ 0\ 1\ 1\ 1\ 1\ 0\ 0\ 1\ 0\ 0\ 0\ 1\ 0\ 1\\ &1\ 1\ 1\ 1\ 0\ 1\ 1\ 1\ 1],\end{aligned}$$

$$\sigma[Row_{27}(L_3)]=16.$$

Because $\sigma[Row_{27}(L_3)]$ is the largest, we start with it to make comparisons and get $Row_{27}(L_1)\leq Row_{27}(L_3)$, $Row_{27}(L_2)\leq Row_{27}(L_3)$. Then $Row_{27}(L_1)$ and $Row_{27}(L_2)$ are deleted and we have $G^1(P_{27})=\{Row_{27}(L_3)\}$. Next, we branch from $G^1(P_{27})$ as follows:

$$\begin{aligned}Row_{27}(L_3)L_1=[&0\ 0\ 1\ 0\ 0\ 1\ 1\ 1\ 1\ 0\ 0\ 1\ 0\ 0\ 1\ 1\ 1\ 1\\ &1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1],\end{aligned}$$

$$\sigma[Row_{27}(L_3)L_1]=19,$$

$$\begin{aligned}Row_{27}(L_3)L_2=[&0\ 0\ 1\ 0\ 0\ 1\ 1\ 1\ 1\ 0\ 0\ 1\ 0\ 0\ 0\ 1\ 0\ 0\\ &1\ 1\ 1\ 1\ 0\ 0\ 1\ 0\ 1],\end{aligned}$$

$$\sigma[Row_{27}(L_3)L_2]=13,$$

$$\begin{aligned}Row_{27}(L_3)L_3=[&1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 0\ 0\ 1\ 1\ 1\\ &1\ 1\ 1\ 1\ 0\ 1\ 1\ 1\ 1],\end{aligned}$$

$$\sigma[Row_{27}(L_3)L_3]=24.$$

For these new RTs, we use the same technique and find $Row_{27}(L_3)L_1\nleq Row_{27}(L_3)L_3$, $Row_{27}(L_3)L_2\leq Row_{27}(L_3)L_3$. So we delete $Row_{27}(L_3)L_2$ and have $G^2(P_{27})=\{Row_{27}(L_3)L_1, Row_{27}(L_3)L_3\}$. Obviously, $\sigma[\Sigma^{(1,27)}]\neq\sigma[\Sigma^{(2,27)}]$, where $\Sigma^{(1,27)}=Row_{27}(L_3)$, $\Sigma^{(2,27)}=Row_{27}(L_3)L_1+Row_{27}(L_3)L_3$. We have to branch from $G^2(P_{27})$. Similar to the above procedure, we will have $G^3(P_{27})=\{Row_{27}(L_3)L_3L_1\}$, $\Sigma^{(3,27)}=Row_{27}(L_3)L_3L_1$. Because $\sigma[\Sigma^{(2,27)}]=\sigma[\Sigma^{(3,27)}]$, we conclude that the largest stability domain of x_d is

$$\Sigma^{(2,27)}=[1\ 1\ 2\ 1\ 1\ 2\ 2\ 2\ 2\ 1\ 1\ 2\ 1\ 0\ 1\ 2\ 2\ 2\ 2\ 2\ 2\ 2\ 2\ 1\ 2\ 2\ 2\ 2],$$

equivalently $\bigcup^{(2,27)}=\Delta_{27}\backslash\{P_{14}\}$.

In the following, we design a feasible state feedback stabilizer by Algorithm 3. As we know, $G^2(P_{27})=\{Row_{27}(L_3)L_1, Row_{27}(L_3)L_3\}$. We find a fine second RT arbitrarily, say $Row_{27}(L_3)L_3$. According to the positions of positive elements in $Row_{27}(L_3)$, we choose some columns of H as follows:

$$Col_j(H) = \delta_3^3 \quad j = 3,6,7,8,9,12,16,18,19,20,21,22,24,25,26,27. \tag{7}$$

We calculate

$$Row_{27}(L_3)L_3 - \Sigma^{(1,27)} = [1\ 1\ 0\ 1\ 1\ 0\ 0\ 0\ 0\ 1\ 1\ 0\ 1\ 0\ 0\ 0\ 1\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0].$$

Then from the positions of positive elements in $Row_{27}(L_3) - \Sigma^{(1,27)}$, some of the other columns can be chosen as follows:

$$Col_j(H) = \delta_3^3, \quad j = 1,2,4,5,10,11,13,17. \tag{8}$$

Now let's see another fine second RT $Row_{27}(L_3)L_1$. Because $Row_{27}(L_3)$ has been considered above, we don't discuss it any more. We calculate

$$Row_{27}(L_3)L_1 - (\Sigma^{(1,27)} + Row_{27}(L_3)L_3) = [-1\ -1\ -1\ -1\ -1\ -1\ -1\ -1\ -1\ -1\ -1\ -1\ -1\ 0\ 1\ -1\ 0\ -1\ -1\ -1\ -1\ -1\ 1\ -1\ -1\ -1\ -1]$$

and get

$$Col_j(H) = \delta_3^1, \quad j = 15,23. \tag{9}$$

The rest of columns, precisely $Col_{14}(H)$, can be chosen arbitrarily, say δ_3^2. Combining it with (7), (8) and (9) yields a state feedback controller

$$u(t) = Hx(t), \quad H = \delta_3[3\ 3\ 3\ 3\ 3\ 3\ 3\ 3\ 3\ 3\ 3\ 3\ 3\ 2\ 1\ 3\ 3\ 3\ 3\ 3\ 3\ 3\ 1\ 3\ 3\ 3\ 3]. \tag{10}$$

It is verifiable that all initial states $x(0) \in \bigcup^{(2,27)}$ can be stabilized to x_d via our state feedback controller (10). For example, when the initial state of the game is chosen to be $x(0) = \delta_{27}^{15}$, equivalently (Waffle, Waffle, Confess), the state-control trajectory (x,u) of the game is

$$(x(0),u(0)) = (\delta_{27}^{15}, \delta_3^1),\ (x(1),u(1)) = (\delta_{27}^{18}, \delta_3^3),$$
$$(x(t),u(t)) = (\delta_{27}^{27}, \delta_3^3),\ t \geq 2.$$

That is, if the detective take actions Deny first and Confess later, he/she must make all other players confess.

5 CONCLUSION

In this paper, we have considered the local stabilization problem of logical systems and derived some interesting results, which are listed as follows. (1) A more general concept of stabilizability of k-valued LCNs has been given. (2) Some necessary and sufficient criteria for the largest stability domain of a fixed point of a k-valued LCN under a constant control have been obtained. (3) An equivalent local stabilizability condition (including global case) has been provided. (4) A constructive design approach to a feasible state feedback stabilizer has been presented. (5) According to the above results, three algorithms have been developed.

REFERENCES

[1] I. Shmulevich, E. Dougherty, W. Zhang, From Boolean to probabilistic Boolean networks as models of genetic regulatory networks, Proceedings of the IEEE, Vol.90, No. 11, 1778-1792, 2002.

[2] A. Adamatzky, On dynamically non-trivial three-valued logics: oscillatory and bifurcatory species, Chaos Solitons and Fractals, Vol.18, No.5, 917-936, 2003.

[3] L. Volkert, M. Conrad, The role of weak interactions in biological systems: The dual dynamics model, J. Theoretical Biology, Vol.193, No.2, 287-306, 1998.

[4] Y. Zhao, Z. Li, D. Cheng, Optimal control of logical control networks, IEEE Trans. Autom. Control, Vol.56, No.8, 1766-1776, 2011.

[5] D. Cheng, F. He, H. Qi, T. Xu, Modeling, analysis and control of networked evolutionary games, IEEE Trans. Autom. Control, Vol.60, No.9, 2402-2415, 2015.

[6] P. Guo, Y. Wang, H. Li, Algebraic formulation and strategy optimization for a class of evolutionary networked games via a semi-tensor product method, Automatica, Vol.49, No.11, 3384-3389, 2013.

[7] D. Cheng, H. Qi, A linear representation of dynamics of Boolean networks, IEEE Trans. Autom. Control, Vol.55 No.10, 2251-2258, 2010.

[8] D. Cheng, H. Qi, Controllability and observability of Boolean control networks, Automatica, Vol.45, No.7, 1659-1667, 2009.

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

[10] D. Cheng, H. Qi, Z. Li, J. Liu, Stability and stabilization of Boolean networks, Int. J. Robust Nonlinear Control, Vol.21, No.2, 134-156, 2011.

[11] H. Chen, J. Sun, Global stability and stabilization of switched Boolean network with impulsive effects, Appl. Math. Comput., Vol.224, No.4, 625-634, 2013.

[12] H. Li, Y. Wang, Z. Liu, Stability analysis for switched Boolean networks under arbitrary switching signals, IEEE Trans. Autom. Control, Vol.59, No. 1978-1982, 2014.

[13] R. Li, M. Yang, T. Chu, State feedback stabilization for Boolean control networks, IEEE Trans. Autom. Control, Vol.58, No.7, 1853-1857, 2013.

[14] H. Li, Y. Wang, Output feedback stabilization control design for Boolean control networks, Automatica, Vol.49, No.12, 3641-3645, 2013.

[15] F. Li, J. Sun, Stability and stabilization of multivalued logical networks, Nonlinear Anal. RWA., Vol.12, No.6, 3701-3712, 2011.

[16] Z. Li, D. Cheng, Algebraic approach to dynamics of multivalued networks, Int. J. Bifurcation Chaos, Vol.20, No.3, 561-582, 2010.

[17] H. Zhang, H. Tian, Z. Wang, Y. Hou, Synchronization analysis and design of coupled Boolean networks based on periodic switching sequences, IEEE Trans. Neural Netw. Learn. Syst., (in press). Doi: 10.1109/TNNLS.2015.2499446.

[18] H. Tian, Z. Wang, Y. Hou, and H. Zhang, State feedback controller design for synchronization of master-slave Boolean networks based on core input-state cycles, Neurocomputing, Vol.174, Part B, 1031-1037, 2016.