Zhipeng Zhang; Zengqiang Chen; Xiaoguang Han; Zhongxin Liu

On the static output feedback stabilization of deterministic finite automata based upon the approach of semi-tensor product of matrices

Kybernetika, Vol. 54 (2018), No. 1, 41–60

Persistent URL: http://dml.cz/dmlcz/147150

KYBERNETIKA — VOLUME *54* (2018), NUMBER 1, PAGES 41–60

ON THE STATIC OUTPUT FEEDBACK STABILIZATION OF DETERMINISTIC FINITE AUTOMATA BASED UPON THE APPROACH OF SEMI-TENSOR PRODUCT OF MATRICES

Zhipeng Zhang, Zengqiang Chen, Xiaoguang Han, Zhongxin Liu

In this paper, the static output feedback stabilization (SOFS) of deterministic finite automata (DFA) via the semi-tensor product (STP) of matrices is investigated. Firstly, the matrix expression of Moore-type automata is presented by using STP. Here the concept of the set of output feedback feasible events (OFFE) is introduced and expressed in the vector form, and the stabilization of DFA is defined in the sense of static output feedback (SOF) control. Secondly, SOFS problem of DFA is investigated within the framework of STP, including single-equilibrium-based SOFS, multi-equilibrium-based SOFS, and further limit cycle-based SOFS. Then the necessary and sufficient conditions for the existence of the three types SOFS are proposed respectively. Meanwhile the efficient and systematic procedures based on the matrix theory to seek the corresponding SOF controller are provided for the three types SOFS problem. Finally, two examples are presented to illustrate the effectiveness of the proposed approach.

Keywords: discrete event dynamic systems, finite automata, static output feedback stabilization, semi-tensor product, output feedback feasible events

Classification: 93D15, 93C65

1. INTRODUCTION

Finite automata are one of the rapidly developing areas in systems and control. In finite automata, states and events are a finite logical or discrete set. There are many kinds of finite automata including deterministic finite automata (DFA) [12], non-deterministic finite automata (NFA) [9] fuzzy finite automata [18] and other [7]. Among these finite automata, DFA have been one of the most applied finite automata, and have been shown to be a powerful and synthesis tool for modeling and analyzing discrete event systems(DESs) [16].

There is an increasing interest in the problem of stability and stabilization of finite automata, especially output feedback stabilization which has received so much attention in the control theory. A variety of stability and stabilization of finite automata have been studied by many researchers in some literature [15, 16]. For example, in [16], the Lyapunov stability and asymptotic stability of finite automata is studied by proposing

DOI: 10.14736/kyb-2018-1-0041

a Lyapunov function-based approach; in [15], the output stabilizability of DFA under partial observation is investigated and a necessary and sufficient condition for output stabilization has been proposed. In this case, the supervisor is of dynamic feedback type by means of constructing a dynamic state observer. As we all know, static output feedback (SOF) control is the simplest closed-loop control and it can be realized easier than dynamic output feedback in practice. Now what is the static output feedback stabilization (SOFS) of DFA? We set the SOF controller as $u(t) = Ky(t)$, where $y(t)$ denotes the output of DFA and K is a constant feedback gain matrix. The problem of SOFS aims at finding an SOF controller above to ensure the closed-loop system has desirable behaviors [17]. There have been significant advances recently in the study of SOFS of Boolean control network (BCN) which is a special form of DFA in [1, 13]. In [13], an sufficient condition (Theorem 2) has been proposed to seek the SOF matrix, that is, designing SOF controller is converted to solving a matrix equation; And later, [1] investigated the SOFS of BCN through both the time-invariant output feedback (TIOF) law and the time-varying output feedback (TVOF) law, and proposed an sufficient and necessary condition for the existence of the TIOF law. Meanwhile, two algorithms to seek the SOF matrix have been given: algorithm 1 which is to solve the nilpotent matrix is based on the matrix theory, and the other which is to find paths with no TIOF-compatible cycle is based on the labeled digraph.

As can be seen, the literature [1, 13] use the theory of Semi-tensor product (STP) [6] which provides a nice systematic approach to mix-valued logical network in recent years. The research findings enrich and develop the theories and methods of various other systems such as BCN [3, 8], nonlinear system [14], fuzzy control systems [21], Petri net systems [10], networked evolutionary games [4] and so on. There have existed some literature concerning the use of STP in finite automata. For example, in [19], the matrix expression and the reachability for finite automata have been investigated; [11] investigated the topological structure properties of DFA and proposed a necessary and sufficient condition for DFA stabilized to a limit cycle by the state feedback controller.

Up to now, there is little result about the solution for the SOFS problem of DFA. In this paper, the problem of SOFS of DFA is investigated via the STP and the algorithm is introduced based on the matrix theory. We aim at finding the algebraic conditions for the existence of SOFS. Compared with the literature [1, 13], a wider DFA system is studied and the conditions and algorithms are given; on the other hand, the concept of the set of output feedback feasible events (OFFE) is introduced and expressed in the vector form, and then we investigate not only the single-equilibrium-based SOFS of DFA, but also the multi-equilibriums and limit cycle. In summary, some of the key contributions of the present paper include the following:

(1) The problem of single-equilibrium-based SOFS of DFA is investigated. A necessary and sufficient condition and an efficient algorithm are proposed.

(2) A necessary and sufficient condition of multi-equilibrium-based SOFS is investigated, and an efficient algorithm to seek the SOF controller is given here.

(3) The equilibrium is generalized to the limit cycle and the SOFS of limit cycle in DFA is investigated. A necessary and sufficient condition for the problem and an efficient algorithm are also proposed.

The remainder section are organized as follows. In section 2, Some preliminaries including STP theory, FEM and prereachability set are introduced. In section 3, the main results of the paper are given here; In subsection 3.1, firstly, the definition of SOFS of DFA is introduced and a necessary and sufficient condition about equilibrium-based SOFS is presented and then we provide an efficient algorithm to seek the SOF controller; In subsection 3.2, a necessary and sufficient condition and algorithm about multi-equilibrium-based SOFS are presented; In subsection 3.3, we investigate the limit cycle-based SOFS and give an efficient algorithm. In section 4, two illustrative examples are given to validate the results, and the examples are analyzed in detail from many different prospectives. In the final section, the present thesis is summarized and we give an outlook of the future research on this topic.

Notation.

- $\mathbb{N}^+$ denotes the set of positive integers.
- $|X|$ denotes the cardinality of set X.
- $k \in [1, n]$ denotes $1 \leq k \leq n$.
- $\mathbb{R}^{m\times n}$ denotes an $m \times n$ real matrices, especially if $n = 1, \mathbb{R}^m := \mathbb{R}^{m\times 1}$.
- $Col(A)$ is the set of all columns of matrix A, $Col_i(A)$ is the ith column of matrix A.
- δ_n^k denotes the kth canonical vector of size k.
- $\delta_n[k_1, k_2, \dots k_p] := [\delta_n^{k_1}, \delta_n^{k_2}, \dots, \delta_n^{k_p}]$.
- $\Delta_n := \{\delta_n^1, \delta_n^2, \dots, \delta_n^n\}$.
- A is a logical matrices if $Col_i(A) \in \Delta_n$, $\mathcal{L}_{m\times n}$ denotes a set of logical matrices respectively..
- $\prod^*$ denotes the set of all finite strings of element of input event set $\prod$.
- $\mathbf{1}_i$ denotes the $i-$dimensional vector with all entries equal to 1.
- $\mathcal{D} := \{0, 1\}$.
- $M_{(i,j)}$ is the (i, j) element of matrix M, A_i is the ith element of vector A.
- A vector A is a Boolean vector if $A_i \in \mathcal{D}$.
- Assume that $A \in \mathbb{R}^m$, $B \in \mathbb{R}^m$. Then $A \wedge B := A_i \wedge B_i$, where the symbol $\wedge$ denotes the logical operator AND.

2. PRELIMINARIES

2.1. STP of matrices

STP which can convert a logical function into an algebraic function is first proposed by Cheng [6]. The theory generalizes the use of conventional matrix product and all the main properties of the conventional matrix product remain true for this generalization. It provides a nice systematic approach to mix-valued logical network in recent years and the brief introduction is given in the following.

Definition 2.1. (Cheng and Qi [6]) For $M \in \mathbb{R}^{m\times n}$, $N \in \mathbb{R}^{m\times n}$, the corresponding STP is defined as:

$$M \ltimes N := (M \otimes I_{r/n})(N \otimes I_{r/p}). \tag{1}$$

Where r denotes the least common multiple of n and p, and $\otimes$ is the Kronecker product. $\ltimes$ is the mathematical symbol of STP.

When n equals to p, the STP coincides with the conventional matrix product. The symbol is omitted for convenience except for the special instructions in this paper. Here some properties about STP which play a fundamental role in the following are presented briefly.

(1) Pseudo-commutability: For $x \in \mathbb{R}^m, y \in \mathbb{R}^n$, $y \ltimes x = W_{[m,n]} \ltimes x \ltimes y$, where the matrix $W_{[m,n]}$ is the swap matrix.

(2) Power-reducing matrix: $x \ltimes x = \Phi_n \ltimes x$ for $\forall x \in \Delta_n$, and $\Phi_n := diag\{\delta_n^1, \delta_n^2, \dots, \delta_n^n\}$.

2.1.1. Matrix expression for DFA

The matrix approach to DFA model has been proposed in [19], and this matrix expression provides an effective computational way for analysis of DFA. Some necessary preliminaries are given as follows.

An Moore-type DFA can be defined as an five-tuple $\mathcal{A} = (X, Y, \prod, f, h)$, where X denotes a finite set of states; Y denotes a finite output collection or a set of output events; $\prod$ is a finite set of events called alphabet; f: $X \times \prod \to X$ denotes the transition function which in general is a partial function on its domain,; the output function $h : X \to Y$ is defined for each state.

Now we introduce the matrix expression based STP for Moore-type DFA. Firstly, some basic symbols are listed, $X = \{x_1, x_2, \dots, x_n\}$, $Y = \{y_1, y_2, \dots, y_p\}$, $\prod = \{e_1, e_2, \dots$ $\dots, e_m\}$ and f: $X \times \prod \to 2^X$, where 2^X is the power set of X, $x_i \sim \delta_n^i$, $y_j \sim \delta_p^j$, $e_k \sim \delta_m^k$, and $i \in [1, n]$, $j \in [1, p]$, $k \in [1, m]$.

Remark 2.2. $\mathcal{A}$ is said to be a DFA considered in this paper if for each $x_i \in X$, $e_k \in \prod$, $|f(x_i, e_k)| \leq 1$; When $|f(x_i, e_k)| = 1$, the transition function $f(x_i, e_k) = x_j$ means that there is a transition labeled by input event e_k from x_i to the state x_j, which is called that the $f(x_i, e_k)$ is defined. When $|f(x_i, e_k)| = 0$, the transition function $f(x_i, e_k) = \emptyset$ means the event e_k cause no state transition from x_i, which is called that the $f(x_i, e_k)$ is undefined.

Secondly, the transition structure matrix (TSM) associated with event is defined as:

$$F := [F_1, F_2, \dots, F_m] \in M_{n \times mn}. \tag{2}$$

Where $F_k \in \mathbb{R}^{n \times n}$is defined as follows:

$$F_{k(l,i)} = \begin{cases} 1, \delta_n^l \in f(\delta_n^i, \delta_m^k) \\ 0, \text{otherwise.} \end{cases} \tag{3}$$

Finally, the dynamics of Moore-type automata are described by the algebraic equations in the following:

$$\begin{aligned} x(t+1) &= Fu(t)x(t) \\ y(t) &= Hx(t). \end{aligned} \tag{4}$$

Where $x(t)$ denotes the state which is reached in t steps from $x(0)$, and $x(0)$ is the initial state, and $y(t)$ is the corresponding output vector. $u(t)$ is the input vector. H denotes the output structure matrix.

Example 2.3. Consider DFA $\mathcal{A} = (X, Y, \prod, f, h)$ shown in the following is taken from [2], where $X = \{x_1, x_2, x_3\}$, $\prod = \{e_1, e_2\}$, $Y = \{y_1, y_2\}$. The TSM and the output structure matrix are constructed by the approach proposed in [19] as follows:

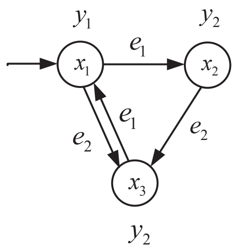

Fig. 1. state transition diagram for Example 2.3.

$$F = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 \end{bmatrix}, \quad H = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 1 \end{bmatrix}.$$

2.1.2. Feasible event matrix of DFA

In this subsection, the concept of feasible event matrix (FEM) is introduced. The feasible event function of each state is defined as $\Psi(x)$ which is an input event set for which $f(x, e)$ is defined, that is, $|f(x_i, e_k)| = 1$. And using the STP theory, the feasible event function can be equivalently represented in the following matrix form:

$$\widehat{u}(t) = Ex(t). \tag{5}$$

Where $\widehat{u}(t) = (\widehat{u}_1(t), \widehat{u}_2(t), \ldots, \widehat{u}_m(t))^T$ represents the corresponding vector of feasible input event of $x(t)$, and E can be defined as:

$$E_{(k,i)} = \begin{cases} 1, & f(x_i, e_k) \text{ is defined} \\ 0, & \text{otherwise.} \end{cases} \tag{6}$$

There is an equivalent formula between the feasible events and the FEM , and the feasible events with respect to the FEM can be calculated by the lemma below:

Lemma 2.4. (Han et al. [11]) Given a DFA $\mathcal{A} = (X, Y, \prod, f, h)$, where $X = \{x_1, x_2, \ldots, x_n\}$, $\prod = \{e_1, e_2, \ldots, e_m\}$, and $X \sim \Delta_n$, $\prod \sim \Delta_m$ respectively. The feasible events $\Psi(x)$ of $\mathcal{A}$ is:

$$\Psi(x_i) = \Psi(\delta_n^i) = \Theta(Col_i(E)), i \in [1, n]. \tag{7}$$

Where $\Theta(x)$ for the nonempty Boolean vector x is defined as $\Theta(x) = \{z \in \Delta_m \,|z \wedge x = z\}$.

The feasible events matrix and the corresponding $\Psi(x)$ in Example 2.3 can be written as:

$$E = \begin{bmatrix} 1 & 0 & 1 \\ 1 & 1 & 0 \end{bmatrix},$$

$$\Psi(x_1) = \{\delta_2^1, \delta_2^2\},\ \Psi(x_2) = \{\delta_2^2\},\ \Psi(x_3) = \{\delta_2^1\}.$$

2.1.3. Prereachability set

In this subsection, we recall the definition of equilibrium point and the definition and some properties of the prereachability set which is to find all permissible feedback input of DFA.

Definition 2.5. Given a DFA $\mathcal{A} = (X, Y, \prod, f, h)$, state $x \in X$ is called an equilibrium point of $\mathcal{A}$ if there exists $e \in \prod$ such that $f(x, e) = x$.

Remark 2.6. In this paper, we assume that the equilibrium point always exists, and the corresponding state is $x_r = \delta_n^r$. If a state is an equilibrium point, there exists a self-loop in state transition diagram, and let Ξ_r denote the set of all events satisfying $f(x_r, e) = x_r$ corresponding to the equilibrium state.

Let us denote by $\Omega_t(s)$ the set of all the states that can be driven to $x_r = \delta_n^r$ in t steps by a feasible transition sequence $e = e_{k_1} e_{k_2} \dots e_{k_t} \in \Pi^*$. The prereachability set $\Omega_t(r)$ is defined as follows:

$$\Omega_t(r) := \{x \in \Delta_n \,|\text{there exist } e_{k_1} e_{k_2} \dots e_{k_t} \in \Pi^* \text{ such that } R(x, e, t) = \delta_n^r\}. \qquad (8)$$

Where $R(x, e, t) = \delta_n^r$ denotes a state $x_r = \delta_n^r$ that is reachable from x in t steps for $\mathcal{A}$.

Lemma 2.7. (Han et al. [11]) Given a DFA $\mathcal{A} = (X, Y, \prod, f, h)$ with the matrix form (4) and let $x_r = \delta_n^r \in \Omega_1(r)$. Then $\Omega_t(r) \subseteq \Omega_{t+1}(r)$, $\forall t \geq 1$.

$x_r = \delta_n^r \in \Omega_1(r)$ if and only if $x_r = \delta_n^r$ is equilibrium point, so the lemma holds obviously according to the definition of equilibrium point and the definition of prereachability set.

Lemma 2.8. (Han et al. [11]) $\Omega_t(r)$ has two following properties:

(1) If $\Omega_1(r) = \{\delta_n^r\}$, then $\Omega_t(r) = \{\delta_n^r\}, \forall t \geq 1$.

(2) If $\Omega_i(r) = \Omega_{i+1}(r)$ for some $i \geq 1$, then $\Omega_t(r) = \Omega_i(r), \forall t \geq i$.

3. SOFS FOR DFA

In this section, we investigate the SOFS of DFA based on STP theory. Firstly, single-equilibrium-based SOFS is discussed, and the sufficient and necessary conditions for single equilibrium point are given. Then we try to expand the discussion to the multi-equilibrium-based SOFS and cycle-based SOFS.

3.1. Single-equilibrium-based SOFS

In this subsection, SOFS based on the equilibrium point stability is investigated. At the first, the concept of equilibrium point stability is introduced and then we give the definition of the single-equilibrium-based SOFS. Meanwhile two necessary conditions are presented, and to this end the corresponding sufficient and necessary conditions are given.

Definition 3.1. Given a DFA $\mathcal{A} = (X, Y, \prod, f, h)$ and an equilibrium point $x_r = \delta_n^r$. A state $x \in X$ is said to be stable at $x_r = \delta_n^r$ if all state trajectories starting from x arrive at $x_r = \delta_n^r$ in a finite number of transitions and stay at $x_r = \delta_n^r$ forever. A system is said to be stable at $x_r = \delta_n^r$ if all states are stable at $x_r = \delta_n^r$.

The definition of the SOFS of DFA is introduced in the following form:

Definition 3.2. A DFA is SOF stabilizable to the equilibrium point $x_r = \delta_n^r$ if there exists the output feedback law $u(t) = Ky(t)$, $K \in \mathcal{L}_{m\times p}$ such that the every state transition trajectory arrives at x_r in a finite number of steps and stays at x_r forever.

We observe that SOFS is in essence equilibrium point stability, and under what conditions we can stabilize the DFA to equilibrium point by resorting to an output feedback. Firstly consider system (4), for each integer $j \in [1, p]$, let $O(y_j)$ denote the set of all states that produce the output $y_j = \delta_p^j$, that is

$$O(y_j) = \{\delta_n^i : Col_i(H) = y_j\}. \tag{9}$$

Significantly, $O(y_{j_1}) \cap O(y_{j_2}) = \emptyset$, $\forall j_1 \neq j_2$, and $\bigcup_{j=1}^{j=p} O(y_j) = \Delta_n$. Let $\pi(y_j)$ denote the set of output feedback feasible events (*OFFE*) corresponding to the output y_j, we can obtain that

$$\pi(y_j) = \left\{ \delta_m^k : \bigcap_{\delta_n^i \in O(y_j)} \Theta(Col_i(E)) \right\}. \tag{10}$$

A necessary (but not sufficient) condition for SOFS is given as follows:

Theorem 3.3. Consider the DFA (4). If the system is SOF stabilizable to the equilibrium point $x_r = \delta_n^r$, then $\pi(y_j) \neq \emptyset$, $\forall j \in [1, p]$.

Proof. If the system is said to be SOF stabilizable to the state x_r, there must exist an output feedback matrix $K = \delta_m[j_1, j_2, \ldots, j_p]$ and $u(t) = Ky(t) = KHx(t)$. For each integer $i \in O(y_j)$, the corresponding SOF events must be the same, namely, the set of OFFE $\pi(y_j) \neq \emptyset$. □

Another necessary (but not sufficient) condition for SOFS can also be given as follows:

Theorem 3.4. Consider the DFA (4). If the system is SOF stabilizable to the equilibrium point, then

(1) $\Xi_r \cap \pi(H\delta_n^r) \neq \emptyset$.

(2) There exists an input value $\delta_m^k \in (\pi(H\delta_n^r) \cap \Xi_r)$ such that $x_r = \delta_n^r \in O(H\delta_n^r)$ is the only equilibrium point.

Proof. If the system is SOF stabilizable to the equilibrium point $x_r = \delta_n^r$, the set of OFFE $\pi(H\delta_n^r)$ and the set Ξ_r must be nonempty. For the condition (2), if there exists another equilibrium point $x_i = \delta_n^i, \delta_n^i \in O(H\delta_n^r)$ for an input value $\delta_m^k \in (\pi(H\delta_n^r) \cap \Xi_r)$, then we cannot find a transition trajectory arrive at x_r from x_i, which contradicts the definition of SOFS. □

Before proceeding, an example of the system showed in the following is considered to illustrate the two necessary conditions.

Example 3.5. Consider DFA. Identifying $x_i \sim \delta_6^i (i \in [1,6])$, $y_j \sim \delta_2^j (j \in [1,2])$ and $e_k \sim \delta_3^k (k \in [1,3])$. $x_1 = \delta_6^1$ is the equilibrium point. Suppose that

$$F = \delta_6[1,2,1,0,6,0,3,4,0,0,0,1,2,1,2,6,0,0]$$

$$H = \delta_2[1,1,1,2,2,2].$$

For each integer $j \in [1,2]$,

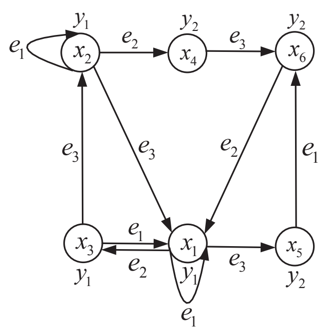

Fig. 2. state transition diagram for Example 3.5.

$$\begin{aligned} \Xi_1 &= \{\delta_3^1\} \\ O(H\delta_6^1) &= O(y_1) = \{\delta_6^1, \delta_6^2, \delta_6^3\} \\ O(y_2) &= \{\delta_6^4, \delta_6^5, \delta_6^6\} \\ \pi(H\delta_6^1) &= \pi(y_1) = \{\delta_3^1, \delta_3^3\} \\ \pi(y_2) &= \emptyset. \end{aligned}$$

And under the input value $\delta_3^1 \in (\pi(H\delta_9^1) \cap \Xi_1)$, $x_2 = \delta_6^2$ is also an equilibrium point belonging to $O(H\delta_6^1)$. The necessary conditions of both Theorem 3.3 and 3.4 are not satisfied, so there no exists an output feedback law such that the system is SOF stabilizable to the equilibrium point $x_1 = \delta_9^1$.

Based on the lemmas and definitions, the sufficient and necessary conditions of SOF stabilization of DFA are presented in the following:

Theorem 3.6. Consider the DFA (4). The system is SOF stabilizable to the equilibrium point $x_r = \delta_n^r$ if and only if there exists an SOF law $u(t) = Ky(t), K \in \mathcal{L}_{m\times p}$ and an integer $\tau \in [1, n-1]$ such that the following conditions are satisfied:

(1) $Col_i(KH) \in \pi(y_j)$, where $\delta_n^i \in O(y_j)$ and $\pi(y_j)$ is the OFFE;

(2) $\Omega_\tau(r) = \Delta_n$;

Proof. (Sufficiency) supposing the conditions (1) and (2) are satisfied, where $\tau \in [1, n-1]$. And namely, there exists an output feedback law $u(t) = Ky(t), K \in \mathcal{L}_{m\times p}$ which is a feasible transition sequence such that $R(x, e, \tau) = \delta_n^r$. We prove that the system is SOF stabilizable at $x_r = \delta_n^r$. Using Lemma 2.7, if $\Omega_\tau(r) = \Delta_n$, then $\Delta_n = \Omega_\tau(r) \subseteq \Omega_{\tau+1}(r) \subseteq \Delta_n$, and then $\Omega_\tau(r) = \Omega_{\tau+1}(r) = \Delta_n$. And using Lemma 2.8, $\Omega_t(r) = \Omega_\tau(r)$ is true for $\forall t \geq \tau$, which implies the sufficient conditions holds according to the definition of the SOF stabilization.

(Necessity) If the system is SOF stabilizable to the equilibrium point, the output feedback control must belong to the set of OFFE. Meanwhile based on the definition of SOFS and the formula (9), the conditions (1) and (2) are significantly established. □

Remark 3.7. We generalize the SOFS conditions of BCN which has been proposed in literature [13] and [1] to the DFA. There are some differences between the result and the proposed conditions about the SOFS of DFA. Because the DFA is the generalization of BCN, we add the condition (1), namely, the SOF matrices must ensure that closed system transition sequence belongs to the set of feasible event.

In fact, the SOF matrices $K \in \mathcal{L}_{m\times p}$ which satisfy the conditions of Theorem 3.6 can be calculated by the following algorithm.

Algorithm a: By resorting the suitable permutations of the state and output components, the equilibrium point is written as $x_1 = \delta_n^1$, and the output structure matrix is presented as $H = diag\{\mathbf{1}_{n_1}^T, \mathbf{1}_{n_2}^T, \dots, \mathbf{1}_{n_p}^T\}$, where $n_1 + n_2 + \cdots + n_p = n$. And the $F \ltimes KH\Phi_n$ can be presented as:

$$[blk_1(F_{v_1}), blk_2(F_{v_2}), \dots blk_p(F_{v_p})] \tag{11}$$

where $|F_{v_j}| = n_j$. Let $v_j^q, q \in [1, n_j]$ denotes the qth indices of F_{v_j}, and the set of OFFE corresponding to the output can be represented as $\pi(y_j) := \bigcap_{i=v_j^1}^{v_j^{n_j}} \Theta(Col_i(E))$. Especially for the equilibrium point $x_1 = \delta_n^1$, $\pi(H\delta_n^1) = \pi(y_1)$.

Step 1: Check whether or not $\pi(y_j) := \bigcap_{i=v_j^1}^{v_j^{n_c}} \Theta(Col_i(E)) \neq \emptyset$, $\forall j \in [1, p]$. If yes, set $\pi(y_j) = \{\delta_m^{c_1^j}, \delta_m^{c_2^j}, \dots, \delta_m^{c_{M_j}^j}\}$, where c_l^j, $l \in [1, M_j]$ denotes the lth index of $\pi(y_j)$, and go to Step 2; otherwise STOP, there no exists an SOF law .

Step 2: Check whether or not $\Xi_1 \cap \pi(H\delta_n^1) \neq \emptyset$ and whether there exists $\delta_m^{c_l^1} \in (\pi(H\delta_n^1) \cap \Xi_1)$ such that x_1 is the only equilibrium point. If yes, find all the events satisfying the conditions, and update the set $\pi(H\delta_n^1) := \{\delta_m^{c_1^1}, \delta_m^{c_2^1}, \ldots, \delta_m^{c_{M_1}^1}\}$. Then set $j = 1$, $v_j := 1$ and go to Step 3; otherwise STOP, there no exists an SOF law.

Step 3: Check whether $F_{v_1^1}$ is δ_n^1 and the principal submatrix with row and column indices $[v_1^2, v_1^{n_1}]$ is nilpotent or not. If yes, set $j = 2$, $v_j := 1$ and go to the Step 4. Otherwise set $v_1 := v_1 + 1$.
Case 1: If $v_1 \leq M_1$, repeat Step 3.
Case 2: If $v_1 > M_1$, then STOP: the SOF law doesn't exist.

Step 4: Case 1: if $j \leq p$ and $v_j \leq M_j$, then check whether the principle submatrix with row and column indices $[v_1^2, \sum_{c=1}^{j} n_c]$ is nilpotent. If yes, set $j := j + 1$, $v_j := 1$, otherwise set $v_j := v_j + 1$. Repeat Step 4.
Case 2: if $j \leq p$ and $v_j > M_j$, set $j := j - 1$ and $v_j := v_j + 1$. If $j = 1$ repeat Step 3, otherwise repeat Step 4.
Case 3: if $j > p$, then STOP; there exists an SOF law and the SOF matrices is $K = \delta_m[c_{v_1}^1, c_{v_2}^2, \ldots, c_{v_p}^p]$.

Remark 3.8. The initialization is imperative and will help us make a better understanding of the algorithm. Step 1 and 2 correspond to the necessary conditions of Theorem 1 and Theorem 2, which can help us reject some infeasible values of the indices and decrease computation amount to enhance the efficiency of algorithm. Step 3 and Step 4 explores all possible SOF matrices and substantially it is to find all nilpotent matrices satisfying the condition.

Remark 3.9. The computational complexity of the proposed Algorithm is at most $o(\prod_{i=1}^{p} M_i)$ which is acceptable.

3.2. Multi-equilibrium-based SOFS

In this subsection, SOFS based on the multi-equilibrium is investigated. Suppose a set of equilibriums are given for the DFA, and we want to stabilize the DFA to the set of equilibriums by the SOF controller. At the first, the definition of the multi-equilibrium-based SOFS is introduced based on the literature [20]

Definition 3.10. A DFA is SOF stabilizable to a set of s equilibriums $\{x_1 = \delta_n^{r_1}, \ldots, x_s = \delta_n^{r_s}\}$ denoted by X_s and $X_s \subseteq X$ if there exists the output feedback law $u(t) = Ky(t)$, $K \in \mathcal{L}_{m\times p}$ such that the every state transition trajectory arrives at X_s in a finite number of steps and stays at X_s forever.

Based on Theorem 3.3 and Theorem 3.4, two necessary conditions are presented here to check the multi-equilibrium-based SOFS.

Theorem 3.11. Consider the DFA (4)and a set of s equilibriums $X_s = \{x_1 = \delta_n^{r_1}, \ldots, x_s = \delta_n^{r_s}\}$. If the system is SOF stabilizable to one of X_s, then $\pi(y_j) \neq \emptyset$, $\forall j \in [1, p]$.

This theorem is proved in a similar way of Theorem 3.3, and the proof procedure is omitted here.

Theorem 3.12. Consider the DFA (4) and a set of s equilibriums X_s. If the system is SOF stabilizable to one of X_s, then $\forall r \in [r_1, r_2, \ldots, r_s]$, $\Xi_{r_i} \cap \pi(H\delta_n^{r_i}) \neq \emptyset$.

This necessary condition holds significantly and the proof procedure is omitted here. Compared with the Theorem 3.4, the necessary condition (2) is reduced in Theorem 3.12. Because if there exists another equilibrium point $x_i = \delta_n^i, i \in O(H\delta_n^r)$ for an input value $\delta_m^k \in (\pi(H\delta_n^r) \cap \Xi_r)$, $x_i = \delta_n^i$ can be defined as one of the set X_s.

Theorem 3.13. Consider the DFA (4). The system is SOF stabilizable to one of $X_s = \{x_1 = \delta_n^{r_1}, \dots x_s = \delta_n^{r_s}\}$ if and only if there exists an SOF law $u(t) = Ky(t), K \in \mathcal{L}_{m\times p}$ and an integer $\tau \in [1, n-s]$ such that the following conditions are satisfied:

(1) $Col_i(KH) \in \pi(y_j)$, where $\delta_n^i \in O(y_i)$ and $\pi(y_j)$ is the OFFE;

(2) $Col_r(KH) \in (\pi(H\delta_n^r) \cap \Xi_r), \forall r \in [r_1, r_2, \dots, r_s]$;

(3) $\Omega_\tau(r) = \Delta_n$;

Proof. (Sufficiency) Supposing the condition (1) and (2) is satisfied. The proof is a simple extension of the proof Theorem 3.6, namely, by Using Lemma 2.7 and Lemma 2.8 we prove that the system is multi-equilibrium-based SOFS. One needs to consider the additional condition $\tau \in [1, n-s]$ and $Col_r(KH) \in (\pi(H\delta_n^r) \cap \Xi_r)$, $\forall r \in [r_1, r_2, \dots, r_s]$, which is straightforward on the strength of Definition 3.10.

(Necessity) Based on the Theorem 3.12 and Theorem 3.6, the conditions (1) and (2) are significantly established □

We aim at designing the controller $u(t) = Ky(t)$ to satisfy the conditions of Theorem 3.13. The corresponding SOF matrices $K \in \mathcal{L}_{m\times p}$ can be calculated by the following algorithm.

Remark 3.14. Here if there exist equilibriums belonging to $O(y_j)$, it is supposed to be only. And next the assumption will be extended to more general case.

Algorithm b: By resorting the suitable permutations of the state and output components, the output structure matrix is presented as $H = diag\{\mathbf{1}_{n_1}^T, \mathbf{1}_{n_2}^T, \dots, \mathbf{1}_{n_p}^T\}$, where $n_1 + n_2 + \cdots + n_p = n$. and $\mathcal{C}_s$ is written as $\{x_{N_1} = \delta_n^{N_1}, x_{N_2} = \delta_n^{N_2}, \dots x_{N_s} = \delta_n^{N_s}\}$, where

$$N_i = \begin{cases} 1, & i = 1 \\ \sum_{j=1}^{i-1} n_j + 1, & i > 1. \end{cases} \tag{12}$$

The matrix $F \ltimes KH\Phi_n$ can be presented as:

$$[blk_1(F_{v_1}), blk_2(F_{v_2}), \dots blk_p(F_{v_p})]$$

where $|F_{v_j}| = n_j$. Let $v_j^q, q \in [1, n_j]$ denotes the qth indices of F_{v_j}, and the set of OFFE can be represented as $\pi(y_j) := \bigcap_{i=v_j^1}^{v_j^{n_j}} \Theta(Col_i(E))$.

Step 1: Check whether or not $\pi(y_j) := \bigcap_{i=v_j^1}^{v_j^{n_c}} \Theta(Col_i(E)) \neq \emptyset$, $\forall j \in [1, p]$. If yes, set $\pi(y_j) = \{\delta_m^{c_1^j}, \delta_m^{c_2^j}, \dots, \delta_m^{c_{M_j}^j}\}$, where c_l^j, $l \in [1, M_j]$ denotes the lth index of $\pi(y_j)$, and go to Step 2; otherwise STOP, there no exists a solution.

Step 2: Check whether or not $\Xi_{N_i} \cap \pi(H\delta_n^{N_i}) \neq \emptyset$ for $i \in [1, s]$. If yes, update the set $\pi(H\delta_n^{N_i}) := \pi(H\delta_n^{N_i}) \cap \Xi_{N_i} = \{\delta_m^{c_1^i}, \delta_m^{c_2^i}, \ldots, \delta_m^{c_{M_i}^i}\}$. Then set $j = 1$, $v_j := 1$ and go to Step 3; otherwise STOP, there no exists an SOF law.

Step 3: Check whether $F_{v_j^{N_j}}$ is $\delta_n^{N_j}$ and the principal submatrix with row and column indices $[v_j^{N_j+1}, v_j^{n_j}]$ is nilpotent or not. If yes, set $j = s+1$, $v_j := 1$, repeat Step 4. Otherwise set $v_j := v_j + 1$.
Case 1: If $v_j \leq M_i$, repeat Step 3;
Case 2: If $v_j \geq M_i$, set $j := j+1$, if $j \leq s$, repeat Step 3; Otherwise STOP: the SOF law doesn't exist.

Step 4: Case 1: if $j \leq p$ and $v_j \leq M_j$, then check whether the principle submatrix with row and column indices $[v_1^2, \sum_{c=1}^{j} n_c]$ except for $v_j^{N_j}$, $j \in [1, s]$ is nilpotent. If yes, set $j := j+1$, $v_j := 1$, otherwise set $v_j := v_j + 1$. Repeat Step 4.
Case 2: if $j \leq p$ and $v_j > M_j$, set $j := j - 1$ and $v_j := v_j + 1$. If $j \leq s$ repeat Step 3, otherwise repeat Step 4.
Case 3: if $j > p$, then STOP; there exists an SOF law such that the DFA is SOF stabilizable to X_s and the SOF matrices is $K = \delta_m[c_{v_1}^1, c_{v_2}^2, \ldots, c_{v_p}^p]$.

Remark 3.15. If there exist multi-equilibriums belonging to $O(y_j)$, simply adjust the formula (12) of the Algorithm. For example, there exist two equilibriums belonging to $O(y_1)$, the parameters in the Algorithm are adjusted as following: N_2 is equal to 2, and $N_i = \sum_{j=1}^{i-2} n_j + 1$ for $i > 2$.

3.3. Cycle-based SOFS

In this subsection, we investigate the problem of SOFS based on the cycle which is a more common case. Firstly suppose that there exist a cycle or multi-cycles in DFA, and we want to stabilize the DFA to the cycle by the SOF controller. And then the definition of cycle-based SOFS is given and some theorems and the corresponding algorithm based on the results are presented.

Definition 3.16. (Han et al. [11]) Given a DFA, an ordered set consisting of finite distinct states $\mathcal{C}_s = \{\delta_n^{r_1}, \delta_n^{r_2}, \ldots \delta_n^{r_s}\} \subseteq X$ is called a limit cycle with length s if there exists a transition sequence $\delta_m^{k_1}\delta_m^{k_2}\ldots\delta_m^{k_s} \in \Sigma^*$, where $[r_1, r_2, \ldots, r_s]$ and $[k_1, k_2, \ldots, k_s]$ represent the sequence of state indices and event indices respectively such that $\delta_n^{r_i} = f(\delta_n^{r_{i-1}}, \delta_m^{k_i})$, $i = 2, 3, \ldots, s$, and $\delta_n^{r_1} = f(\delta_n^{r_s}, \delta_m^{j_s})$

Limit cycle can be viewed as the generalization of equilibrium, namely, equilibrium is a special cycle of length 1, and limit cycle is also a elementary cycle. Here we give the definition of cycle-based SOFS.

Definition 3.17. A DFA is SOF stabilizable to a cycle of length s denoted by $\mathcal{C}_s$ if there exists the output feedback law $u(t) = Ky(t)$, $K \in \mathcal{L}_{m\times p}$ such that the every state transition trajectory arrives at $\mathcal{C}_s$ in a finite number of steps and stay at $\mathcal{C}_s$ forever.

The detection and stabilization of limit cycle for DFA has been investigated in [11], and the approach of calculating the limit cycles of different lengths has been given. However, do there exist the difference between the cycle in [11] and the cycle this paper? The answer is yes, this leads to the following definition and theorem.

Definition 3.18. Given a DFA and a cycle of length s denoted by $\mathcal{C}_s$, namely, $\delta_n^{r_1} \overset{\delta_m^{k_1}}{\rightarrow} \delta_n^{r_2} \overset{\delta_m^{k_2}}{\rightarrow} \cdots \overset{\delta_m^{k_{s-1}}}{\rightarrow} \delta_n^{r_s} \overset{\delta_m^{k_s}}{\rightarrow} \delta_n^{r_1}$, if $H\delta_n^{r_i} = H\delta_n^{r_j} \Rightarrow \delta_m^{k_i} = \delta_m^{k_j}$ for $\forall i, j \in [1, s]$, the cycle $\mathcal{C}_s$ is SOF-compatible.

Now based on definition above, a necessary condition is presented here to check the cycle-based SOFS.

Theorem 3.19. Given a DFA and a cycle of length s denoted by $\mathcal{C}_s$, If the system is SOF stabilizable to the cycle $\mathcal{C}_s$, then the cycle $\mathcal{C}_s$ must be SOF-compatible.

Another necessary (but not sufficient) condition for cycle-based SOFS can also be given as follows:

Theorem 3.20. Consider the DFA (4) and a cycle denoted by $\mathcal{C}_s = \{\delta_n^{r_1}, \delta_n^{r_2}, \ldots \delta_n^{r_s}\}$ and the corresponding transition sequence is $\delta_m^{k_1}\delta_m^{k_2}\ldots, \delta_m^{k_s}$. If the system is SOF stabilizable to $\mathcal{C}_s$, then

(1) $\forall j \in [1, p],\ \pi(y_j) \neq \emptyset$;

(2) $\forall i \in [1, 2, \ldots s],\ \delta_m^{k_i} \in \pi(H\delta_n^{r_i})$.

Proof. The necessary condition (1) holds significantly like the Theorem 3.3 and Theorem 3.11. If the cycle has already been determined, the corresponding transition sequence is also fixed, and according to the Theorem 3.4 and Theorem 3.12, the condition (2) is proved. □

Theorem 3.21. Consider the DFA (4)and a cycle denoted by $\mathcal{C}_s = \{\delta_n^{r_1}, \delta_n^{r_2}, \ldots, \delta_n^{r_s}\}$ and the corresponding transition sequence $\delta_m^{k_1}\delta_m^{k_2}\ldots, \delta_m^{k_s}$. The system is SOF stabilizable to $\mathcal{C}_s$ if and only if there exists an SOF law $u(t) = Ky(t), K \in \mathcal{L}_{m\times p}$ and an integer $\tau \in [1, n-s]$ such that the following conditions are satisfied:

(1) $Col_i(KH) \in \pi(y_j)$, where $\delta_n^i \in O(y_i)$ and $\pi(y_j)$ is the OFFE;

(2) $Col_{r_i}(KH) = \delta_m^{k_i},\ \forall i \in [1, s]$;

(2) $\Omega_\tau(r) = \Delta_n$.

This theorem is proved in a similar way of Theorem 3.13, and the proof procedure is omitted here.

Remark 3.22. Supposed that the DFA has already exist a cycle of length s which is SOF-compatible. The algorithm to seek the SOF-compatible cycle is not discussed here.

The controller $u(t) = Ky(t)$ is designed to satisfy the conditions of Theorem 3.21 where the corresponding SOF matrices $K \in \mathcal{L}_{m\times p}$ can be calculated by the following algorithm.

Remark 3.23. For convenience and clear, supposed that the states in the cycle have $s-1$ different outputs which means that there exist two states of cycle with the same output.

Algorithm c: By resorting the suitable permutations of the state and output components, the output structure matrix is presented as $H = diag\{\mathbf{1}_{n_1}^T, \mathbf{1}_{n_2}^T, \dots, \mathbf{1}_{n_p}^T\}$, where $n_1 + n_2 + \dots + n_p = n$. and X_s is written as $\{x_{N_1} = \delta_n^{N_1}, x_{N_2} = \delta_n^{N_2}, \dots x_{N_s} = \delta_n^{N_s}\}$, where

$$N_i = \begin{cases} i, & i = 1,2 \\ \sum_{j=1}^{i-2} n_j + 1, & i > 2. \end{cases} \tag{13}$$

The corresponding transition sequence of the cycle $\mathcal{C}_s$ is $\delta_m^{k_1}\delta_m^{k_2}\dots\delta_m^{k_s}$. The matrix $F \ltimes KH\Phi_n$ can be presented as:

$$[blk_1(F_{v_1}), blk_2(F_{v_2}), \dots blk_p(F_{v_p})]$$

where $|F_{v_j}| = n_j$. Let $v_j^q, q \in [1, n_j]$ denotes the qth indices of F_{v_j}, and the set of OFFE can be represented as $\pi(y_j) := \bigcap_{i=v_j^1}^{v_j^{n_j}} \Theta(Col_i(E))$.

Step 1: Check whether or not the cycle $\mathcal{C}_s$ $\{x_{N_1} = \delta_n^{N_1}, x_{N_2} = \delta_n^{N_2}, \dots x_{N_s} = \delta_n^{N_s}\}$ is SOF-compatible. If yes, go to Step 2; otherwise STOP, there no exists an SOF law such that the system is SOF stabilizable to cycle $\mathcal{C}_s$.

Step 2: Check whether or not $\pi(y_j) := \bigcap_{i=v_j^1}^{v_j^{n_c}} \Theta(Col_i(E)) \neq \emptyset$, $\forall j \in [1, p]$. If yes, set $\pi(y_j) = \{\delta_m^{c_1^j}, \delta_m^{c_2^j}, \dots, \delta_m^{c_{M_j}^j}\}$, where c_l^j, $l \in [1, M_j]$ denotes the lth index of $\pi(y_j)$, and go to Step 3; otherwise STOP, there no exists an SOF law.

Step 3: Check whether or not $\delta_m^{k_i} \subseteq \pi(H\delta_n^{N_i})$ for $i \in [1, s]$. If yes, update the set $\pi(H\delta_n^{N_i}) := \{\delta_m^{k_i}\}$. And go to Step 4; otherwise STOP, there no exists an SOF law.

Step 4: Check whether the principal submatrix with row and column indices $[v_j^{N_j+1}, v_j^{n_j}]$ except for $v_j^{N_j}$, $j \in [1, s-1]$ is nilpotent or not. If yes, set $j = s$, $v_j := 1$, and go to the Step 5. Otherwise STOP: the SOF law doesn't exist.

Step 5: Case 1: if $j \leq p$ and $v_j \leq M_j$, then check whether the principle submatrix with row and column indices $[v_1^2, \sum_{c=1}^{j} n_c]$ except for $v_j^{N_j}$, $j \in [1, s]$ is nilpotent. If yes, set $j := j+1$, $v_j := 1$, otherwise set $v_j := v_j + 1$. Repeat Step 5.

Case 2: if $j \leq p$ and $v_j > M_j$, set $j := j-1$ and $v_j := v_j + 1$, repeat Step 5. If $j \leq s-1$ then STOP: the SOF law doesn't exist.
Case 3: if $j > p$, then STOP; there exists an SOF law such that the DFA is SOF stabilizable to $\mathcal{C}_s$ and the SOF matrices is $K = \delta_m[k_1, k_3, \dots, k_s, c_{v_{s+1}}^{s+1}, \dots, c_{v_p}^p]$.

Remark 3.24. Now we extend the assumption to more general case. If the states of the cycle $\mathcal{C}_s$ have s different outputs or other number different outputs, simply adjust the formula (13) of the **Algorithm c** above. For example, if there exists $s-3$ different outputs including three equal to y_1 and two equal to y_2, the parameters are adjusted as following:

$$N_i = \begin{cases} i, & i = 1,2,3 \\ n_1 + i - 3, & i = 4,5 \\ \sum_{j=1}^{i-5} n_j + 1, & i > 5. \end{cases} \tag{14}$$

4. ILLUSTRATIVE EXAMPLES

In this section, two examples are illustrated to validate the results and analyzed in details from many different prospective.

Example 4.1. Consider DFA. $X = \{x_1, x_2, x_3, x_4, x_5\}$, $\prod = \{e_1, e_2, e_3, e_4\}$, $Y = \{y_1, y_2, y_3, y_4\}$. Identifying $x_i \sim \delta_5^i (i \in [1,5])$, $y_j \sim \delta_4^j (j \in [1,4])$ and $e_k \sim \delta_4^k (k \in [1,4])$.

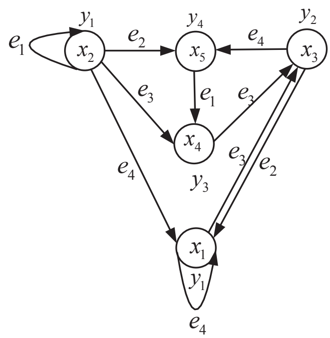

Fig. 3. state transition diagram for Example 4.1.

And by resorting to the suitable permutations of the state and output components, we get the dynamics of Moore-type automata as follows:

$$\begin{aligned} x(t+1) &= Fu(t)x(t) \\ y(t) &= Hx(t). \end{aligned} \tag{15}$$

Where the transition structure matrix F, the output structure matrix H and the feasible events matrix E are

$$\begin{aligned} F &= \delta_5[0,2,0,0,4,0,5,1,0,0,3,4,0,3,0,1,1,5,0,0], \\ H &= \delta_4[1,1,2,3,4], \\ E &= \begin{bmatrix} 0 & 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 \\ 1 & 1 & 1 & 0 & 0 \end{bmatrix}. \end{aligned}$$

A cycle exists in Example 4.1, so the considered automata are not stable to the equilibrium point $x_1 = \delta_5^1$. Now we investigate the SOFS of the considered DFA at equilibrium point. If we can find the SOF law $u(t) = Ky(t), K \in \mathcal{L}_{m\times p}$ satisfying the conditions of Theorem 3.6, the system will be SOFS to the equilibrium point.

According to the (9) and (10), we can obtain that for each integer $j \in [1, 4]$,

$$\begin{aligned}
\Xi_1 &= \{\delta_4^4\},\\
O(H\delta_5^1) &= O(y_1) = \{\delta_5^1, \delta_5^2\},\\
O(y_2) &= \{\delta_5^3\},\\
O(y_3) &= \{\delta_5^4\},\\
O(y_4) &= \{\delta_5^5\},\\
\pi(H\delta_5^1) &= \pi(y_1) = \{\delta_4^3, \delta_4^4\},\\
\pi(y_2) &= \{\delta_4^2, \delta_4^4\},\\
\pi(y_3) &= \{\delta_4^3\},\\
\pi(y_4) &= \{\delta_4^4\}.
\end{aligned}$$

And the necessary conditions of theorem 3.3 and theorem 3.4 are satisfied. By resorting to **Algorithm a**, we can calculate that the optimal SOF matrix is $K = \delta_4[4, 2, 3, 1]$. All state transition trajectories arrive at the equilibrium point $x_1 = \delta_5^1$ for $\forall t \geq 3$, and the corresponding SOF paths with event string and SOF controller are: $x_5 \underset{Ky_4}{\overset{e_1}{\to}} x_4 \underset{Ky_3}{\overset{e_3}{\to}} x_3 \underset{Ky_2}{\overset{e_2}{\to}} x_1$ and $x_2 \underset{Ky_1}{\overset{e_4}{\to}} x_1$.

Remark 4.2. The considered DFA is also a multi-equilibrium system, and $X_s = \{x_1, x_2\}$. We can find that $\pi(H\delta_5^2) = \pi(y_1) = \{\delta_4^4\}$ is not belonging to the set $\Xi_2 = \{\delta_4^1\}$, so the necessary condition of theorem 3.12 is not satisfied, and then the system cannot be SOF stabilizable to the set of multi-equilibrium X_s.

The considered DFA has a cycle $\mathcal{C}_3 = \{\delta_5^3, \delta_5^5, \delta_5^4\}$ and the corresponding transition sequence is $\delta_4^4\delta_4^1\delta_4^3$. Now we investigate that whether the system is cycle-based SOFS. Firstly, the states of cycle $x_3 \underset{y_2}{\overset{e_4}{\to}} x_5 \underset{y_4}{\overset{e_1}{\to}} x_4 \underset{y_3}{\overset{e_3}{\to}} x_3$ have different outputs, which mean that the cycle is SOF-compatible. By resorting to the **Algorithm c**, we can calculate that the optimal SOF matrix is $K = \delta_4[3, 4, 3, 1]$.

Example 4.3. Consider a traffic system. $X = \{x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_8, x_9\}$ denotes different locations in the map, $\prod = \{e_1, e_2, e_3, e_4\}$ denotes four different types of traffic signal indicators, $Y = \{y_1, y_2, y_3\}$ represents the road names where the locations are. The traffic system can be represented as the DFA transition diagram in the following. The case that people in different locations get to x_1 can be represented as the single-equilibrium based stabilization.

And by resorting to the suitable permutations of the state and output components, identifying $x_i \sim \delta_9^i (i \in [1, 9])$, $y_j \sim \delta_3^j (j \in [1, 3])$ and $e_k \sim \delta_4^k (k \in [1, 4])$, and the transition structure matrix F, the output structure matrix H and the feasible events matrix E are

$$F = \delta_9[1, 5, 9, 3, 0, 0, 0, 0, 0, 7, 4, 5, 9, 4, 8, 9, 0, 0, 6, 8, 0, 0, 5, 2, 1, 1, 3, 0, 0, 0, 0, 0, 0, 0, 2, 1],$$

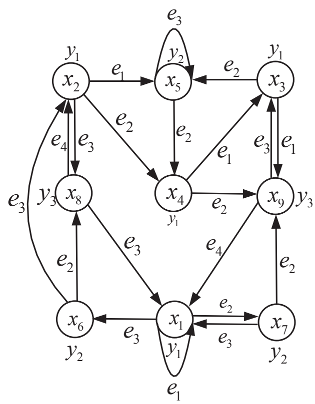

Fig. 4. state transition diagram for Example 4.3.

$$H = \delta_3[1,1,1,1,2,2,2,3,3],$$

$$E = \begin{bmatrix} 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \end{bmatrix}.$$

A cycle exists in the considered DFA, so the DFA is not stable to the equilibrium point $x_1 = \delta_9^1$. That is, people may be trapped in the cycle, and can not get to the destination x_1. Now we investigate the SOFS of the considered DFA at equilibrium point. According to the (9) and (10), we can obtain that for each integer $j \in [1,3]$,

$$\begin{aligned} \Xi_1 &= \{\delta_4^2\}, \\ O(H\delta_9^1) &= O(y_1) = \{\delta_9^1, \delta_9^2, \delta_9^3, \delta_9^4\}, \\ O(y_2) &= \{\delta_9^5, \delta_9^6, \delta_9^7\}, \\ O(y_3) &= \{\delta_9^8, \delta_9^9\}, \\ \pi(H\delta_5^1) &= \pi(y_1) = \{\delta_4^1\}, \\ \pi(y_2) &= \{\delta_4^2, \delta_4^3\}, \\ \pi(y_3) &= \{\delta_4^3, \delta_4^4\}. \end{aligned}$$

And the necessary conditions of theorem 3.3 and theorem 3.4 are satisfied. By resorting to **Algorithm a**, we can calculate that the optimal SOF matrix is $K = \delta_4[1,2,4]$. People can get to the location $x_1 = \delta_9^1$ for $\forall t \geq 3$, and the corresponding SOF paths with event string and SOF controller are: $x_6 \underset{Ky_2}{\overset{e_3}{\longrightarrow}} x_2$, $x_7 \underset{Ky_3}{\overset{e_2}{\longrightarrow}} x_9$ and $x_8 \underset{Ky_3}{\overset{e_4}{\longrightarrow}} x_2 \underset{Ky_1}{\overset{e_1}{\longrightarrow}} x_5 \underset{Ky_2}{\overset{e_2}{\longrightarrow}} x_4 \underset{Ky_1}{\overset{e_1}{\longrightarrow}} x_3 \underset{Ky_1}{\overset{e_1}{\longrightarrow}} x_9 \underset{Ky_3}{\overset{e_4}{\longrightarrow}} x_1$.

Remark 4.4. The SOF path with event string and SOF controller can be represented as: people in the road y_1 including the locations x_1, x_2, x_3, x_4 go along with the first type of traffic signal indicator, and people in the road y_2 including the locations x_5, x_6, x_7 go along with the second type of traffic signal indicator, and people in the road y_3 including the locations x_8, x_9 go along with the forth type of traffic signal indicator.

If we want that people in different locations get to the set $X_s = \{x_1, x_5\}$, the case can be regarded as that the considered DFA is multi-equilibrium based stabilizable to the $X_s = \{x_1, x_5\}$. We can find that the necessary conditions of theorem 3.11 and Theorem 3.12 are satisfied, and by resorting to **Algorithm b**, the system can also be SOF stabilizable to the set of multi-equilibrium X_s by the SOF matrix $K = \delta_4[1, 3, 4]$. The corresponding SOF paths with event string and SOF controller are: $x_6 \underset{Ky_2}{\overset{e_3}{\rightarrow}} x_2 \underset{Ky_1}{\overset{e_1}{\rightarrow}} x_5$, $x_8 \underset{Ky_3}{\overset{e_4}{\rightarrow}} x_2 \underset{Ky_1}{\overset{e_1}{\rightarrow}} x_5$ and $x_7 \underset{Ky_2}{\overset{e_3}{\rightarrow}} x_1$, $x_4 \underset{Ky_1}{\overset{e_1}{\rightarrow}} x_3 \underset{Ky_1}{\overset{e_1}{\rightarrow}} x_9 \underset{Ky_3}{\overset{e_4}{\rightarrow}} x_1$.

If we want that people in different locations get to the cycle $\mathcal{C}_3 = \{\delta_9^3, \delta_9^5, \delta_9^4, \delta_9^9\}$, the case can be regarded as that the considered DFA is cycle-based stabilizable to the cycle. The considered DFA has a cycle $\mathcal{C}_3 = \{\delta_9^3, \delta_9^5, \delta_9^4, \delta_9^9\}$ and the corresponding transition sequence is $\delta_4^2\delta_4^1\delta_4^2\delta_4^2$. Now we investigate that whether the system is SOF stabilizable to $\mathcal{C}_s$. Firstly, the states of cycle $x_3 \underset{y_1}{\overset{e_2}{\rightarrow}} x_5 \underset{y_2}{\overset{e_2}{\rightarrow}} x_4 \underset{y_1}{\overset{e_2}{\rightarrow}} x_9 \underset{y_3}{\overset{e_3}{\rightarrow}} x_3$ have 3 different outputs and we can find that the cycle is SOF-compatible. The suitable permutations of the state and output components can be found out and by resorting to the **Algorithm c**, we can calculate that the optimal SOF matrix is $K = \delta_4[2, 2, 4]$.

Remark 4.5. The parameters in the **Algorithm c** are adjusted as following:

$$N_i = \begin{cases} i, & i = 1, 2 \\ \sum_{j=1}^{i-2} n_j + 1, & i > 2. \end{cases} \tag{16}$$

4.1. Conclusion

In this paper, we investigate the equilibrium-based SOFS and cycle-based SOFS of DFA by using STP of matrices. The sufficient and necessary algebraic conditions for the existence of the SOFS are given, and then efficient algorithms to seek the SOF controller are provided respectively. We generalize the SOFS results of BCN to the DFA via the approach of STP of matrices, which helps the reader to the extended version of SOFS problem of DFA and makes a better understanding of feedback stabilization. Two examples are presented here to illustrate the effectiveness of the results.

A complete solution for the SOFS problem of DFA and how to reduce the computational complexity effectively of algorithms is still a challenging open problem. The powerful ability of finite automata and its widely use deserve our continue exploration. Future work will concentrate on the following directions:

(1) We try to find better result and the corresponding algorithm, and then generalize the research to the set-based SOFS of DFA.

(2) Automata theory is a synthesis tool for modeling and analyzing DESs, next we will investigate the SOFS problem of DESs.

ACKNOWLEDGEMENT

This work is supported by National Natural Science Foundation (NNSF) of China under Grant Nos. 61573199 and 61573200.

(Received May 20, 2017)

REFERENCES

[1] N. Bof, E. Fornasini, and M. Valcher: Output feedback stabilization of Boolean control networks. Automatica *57* (2015), 21–28. DOI:10.1016/j.automatica.2015.03.032

[2] C. Cassandras and S. Lafortune: Introduction to Discrete Event System. Second edition. Springer Science and Business Media, New York 2008. DOI:10.1007/978-0-387-68612-7

[3] D. Cheng: Disturbance decoupling of Boolean control networks. IEEE Trans. Automat. Control *56* (2011), 2–10. DOI:10.1109/tac.2010.2050161

[4] D. Cheng, F. He, H. Qi, and T. Xu: Modeling, analysis and control of networked evolutionary games. IEEE Trans. Automat. Control *60* (2015), 2402–2415. DOI:10.1109/tac.2015.2404471

[5] D. Cheng and H. Qi: Controllability and observability of Boolean control networks. Automatica *45* (2009), 1659–1667. DOI:10.1016/j.automatica.2009.03.006

[6] D. Cheng and H. Qi: A linear representation of dynamics of Boolean networks. IEEE Trans. Automat. Control *55* (2010), 2251–2258. DOI:10.1109/tac.2010.2043294

[7] R. Daniel and L. Markus: Automata with modulo counters and nondeterministic counter bounds. Kybernetika *50* (2014), 66–94. DOI:10.14736/kyb-2014-1-0066

[8] E. Fornasini and M. E Valcher: On the periodic trajectories of Boolean control networks. Automatica *49* (2013), 1506–1509. DOI:10.1016/j.automatica.2013.02.027

[9] J. Holub: The finite automata approaches in stringology. Kybernetika *48* (2012), 386–401.

[10] X. Han, Z. Chen, Z. Liu, et al.: Calculation of siphons and minimal siphons in Petri nets based on semi-tensor product of matrices. IEEE Trans. Systems, Man Cybernet.: Systems *47* (2017), 531–536. DOI:10.1109/tsmc.2015.2507162

[11] X. Han, Z Chen, Z. Liu, et al.: The detection and stabilisation of limit cycle for deterministic finite automata. Int. Control *91* (2017), 4, 874–886. DOI:10.1080/00207179.2017.1295319

[12] A. Kobetski and M Fabian: Time-optimal coordination of flexible manufacturing systems using deterministic finite automata and mixed integer linear programming. Discrete Event Dynamic Systems *19* (2009), 287–315. DOI:10.1007/s10626-009-0064-9

[13] H. Li and Y. Wang: Output feedback stabilization control design for BCNs. Automatica *49* (2013), 3641–3645. DOI:10.1016/j.automatica.2013.09.023

[14] Z. Li, Y. Qiao, H. Qi, and D. Cheng: Stability of switched polynomial systems. J. Systems Science Complexity *21* (2008), 362–377. DOI:10.1007/s11424-008-9119-5

[15] C. Ozveren and A. Willsky: Output stabilizability of discrete event dynamic systems. IEEE Trans. Automat. Control *19* (1991), 925–935. DOI:10.1109/9.133186

[16] K. Passino, A. Michel, and P. Antsaklis: Lyapunov stability of a class of discrete event systems. IEEE Trans. Automat. Control *39* (1994), 269–279. DOI:10.1109/9.272323

[17] V. Syrmos, C. Abdallah and P. Dorato, et al.: Static output feedback – A survey. Automatica *33* (1997), 125–137. DOI:10.1016/s0005-1098(96)00141-0

[18] S. P. Tiwari and A. K. Srivastava: On a decomposition of fuzzy automata. Fuzzy Sets Systems *151* (2005), 503–511. DOI:10.1016/j.fss.2004.06.014

[19] X. Xu and Y. Hong: Matrix expression and reachability analysis of finite automata. J. Control Theory Appl. *10* (2012), 210–215. DOI:10.1007/s11768-012-1178-4

[20] X. Xu, Y. Zhang, and Y. Hong: Matrix approach to stabilization of deterministic finite automata. In: Proc. American Control Conference, Washington 2013, pp. 3242–3247. DOI:10.1109/acc.2013.6580331

[21] Y. Yan, Z. Chen, and Z. Liu: Solving type-2 fuzzy relation equations via semi-tensor product of matrices. Control Theory Technol. *12* (2014), 173–186. DOI:10.1007/s11768-014-0137-7

Zhipeng Zhang, College of Computer and Control Engineering, Nankai University, Tianjin 300350. P. R. China.
e-mail: yueyueniao20091406@163.com

Zengqiang Chen, Corresponding author. College of Computer and Control Engineering, Nankai University, Tianjin 300350, College of Science, Civil Aviation University of China, Tianjin 300300. P. R. China.
e-mail: chenzq@nankai.edu.cn

Xiaoguang Han, College of Computer and Control Engineering, Nankai University, Tianjin 300350. P. R. China.
e-mail: hxg-allen@163.com

Zhongxin Liu, College of Computer and Control Engineering, Nankai University, Tianjin 300350. P. R. China.
e-mail: Lzhx@nankai.edu.cn