Title	Stabilization of Finite Automata with Application to Hybrid Systems Control
Author(s)	Kobayashi, Koichi; Imura, Jun-ichi; Hiraishi, Kunihiko
Citation	Discrete Event Dynamic Systems, 21(4): 519-545
Issue Date	2011-06-17
Type	Journal Article
Text version	author
URL	http://hdl.handle.net/10119/10789
Rights	This is the author-created version of Springer, Koichi Kobayashi, Jun-ichi Imura, and Kunihiko Hiraishi, Discrete Event Dynamic Systems, 21(4), 2011, 519-545. The original publication is available at www.springerlink.com, http://dx.doi.org/10.1007/s10626-011-0110-2
Description	

Discrete Event Dynamic Systems manuscript No.
(will be inserted by the editor)

Stabilization of Finite Automata with Application to Hybrid Systems Control

Koichi Kobayashi · Jun-ichi Imura ·
Kunihiko Hiraishi

Received: date / Accepted: date

Abstract This paper discusses the state feedback stabilization problem of a deterministic finite automaton (DFA), and its application to stabilizing model predictive control (MPC) of hybrid systems. In the modeling of a DFA, a linear state equation representation recently proposed by the authors is used. First, this representation is briefly explained. Next, after the notion of equilibrium points and stabilizability of the DFA are defined, a necessary and sufficient condition for the DFA to be stabilizable is derived. Then a characterization of all stabilizing state feedback controllers is presented. Third, a simple example is given to show how to follow the proposed procedure. Finally, control Lyapunov functions for hybrid systems are introduced based on the above results, and the MPC law is proposed. The effectiveness of this method is shown by a numerical example.

Keywords Finite automata · Hybrid systems · Model predictive control · Stabilization

1 Introduction

To overcome the hardness of analysis/synthesis of complex systems, recently, there have been several works on finite-state approximations such as (bi)simulation re-

---

This work was partially supported by JSPS Grant-in-Aid for Young Scientists (B) 23760387 and Scientific Research (C) 21500009.

Koichi Kobayashi
School of Information Science, Japan Advanced Institute of Science and Technology,
Nomi, Ishikawa, 923-1292, Japan
E-mail: k-kobaya@jaist.ac.jp

Jun-ichi Imura
Graduate School of Information Science and Engineering, Tokyo Institute of Technology,
Oh-okayama, Tokyo, 152-8552, Japan

Kunihiko Hiraishi
School of Information Science, Japan Advanced Institute of Science and Technology,
Nomi, Ishikawa, 923-1292, Japan

lations (e.g., [8,18]) and qualitative models (e.g., [2,7]). In these approaches, dynamical systems are approximated by finite automata (directed graphs), and the analysis/control problems of them are in general reduced to a kind of search problems on graphs. However, more theoretical approaches to the stabilization problem, which is one of the basic and important problems, will be required for further developments of such approaches. On the other hand, the obtained approximate models may be regarded as discrete event systems. Based on a formal language-based approach, theoretical results for discrete event systems have been obtained so far [6]. The stabilization problem of discrete event systems is to find a controller such that the state transits to a given finite set of states in finite step, and has been discussed for general finite automata [4,11,14] and deterministic finite automata (DFA) [16,17]. To our knowledge, a linear characterization (parametrization) of all stabilizing controllers has not been derived yet. A linear form of all stabilizing controllers is useful for not only finite automata but also hybrid systems with discrete dynamics. In the control problem of hybrid systems, little attention to discrete dynamics (finite automata) has been paid so far. Most recently the significance of further works on the discrete dynamics in hybrid systems control has been pointed out, and an efficient method of stabilizing model predictive control (MPC) based on this viewpoint has been proposed in [5]. In this literature, the analytical results on DFAs, e.g., a state feedback gain that can be computed in offline, are not used, but it will be desirable that these are used for reducing the online computation time. In addition, to apply efficient solution methods or solvers to MPC, equality/inequality constraints must be given as a linear form, and it is important to consider a linear characterization.

On the other hand, as for the modeling of finite automata, the authors have recently proposed in [9,10] a new modeling method of representing a finite automaton as *a linear state equation* including binary linear constraints with a relatively small number of free binary variables (called here binary input variables), for reducing the computation time to solve the MPC problem of hybrid systems. It has also been shown that our method is very effective by numerical examples of the optimal control problem of switched/PWA systems. Since the proposed representation is similar to a linear state equation in control theory, it is expected that a new framework of analysis/control synthesis of finite automata will be developed based on this representation.

Motivated by the above backgrounds, this paper addresses the stabilization problem of DFAs, and its application to stabilizing MPC, based on the above linear state equation representation. First, as a natural extension of the stability for continuous systems, the notion of stabilizability of DFAs, in other words, the state equation expressing it, is defined based on the minimality of the state transition trajectory, and a necessary and sufficient condition for a given DFA to be stabilizable is derived. Next, by means of this condition, a linear characterization of all stabilizing controllers for DFAs is proposed. Finally, based on the above result, the stabilizing MPC problem is discussed. In this paper, a control Lyapunov function (CLF) approach [12,15], which is well known in stabilization of nonlinear systems, is applied, and two control Lyapunov functions (CLFs), i.e., a continuous CLF and a discrete CLF, are introduced to derive a stabilizing MPC law according to the result in [5].

This paper is organized as follows. After a state equation representation of DFA is briefly explained in Section 2, equilibrium points and stabilizability of DFA are defined, and a condition for a given DFA to be stabilizable is derived in Section 3. In Section 4, a stabilizing state feedback controller is characterized. In Section 5, a simple example is shown. In Section 6, the obtained result on DFAs is applied to stabilizing MPC of hybrid systems is discussed. In Section 7, we conclude this paper.

Notation: Let $\mathbf{R}$ and $\mathbf{Z}_+$ denote the set of real numbers and the set of semi-positive integers, respectively. Let $\{0,1\}^{m\times n}$ denote the set of $m \times n$ matrices, which consists of elements 0 and 1. Let I_n, $0_{m\times n}$ and e_n denote the $n \times n$ identity matrix, the $m \times n$ zero matrix, and the $n \times 1$ vector whose elements are all 1s, respectively. For simplicity of notation, we sometimes use the symbol 0 instead of $0_{m\times n}$. To denote the i-th element of a vector $a \in \mathbf{R}^n$, we use either $a^{(i)}$ or a_i. A function $\phi(s) : [0,\infty) \to [0,\infty)$ belongs to class $\mathcal{K}_\infty$ if $\phi(s)$ is continuous, strictly increasing, and $\phi(0) = 0$, $\lim_{s\to\infty} \phi(s) = \infty$ hold.

2 State Equation of Deterministic Finite Automata

Consider the following tuple expressing a deterministic finite automaton (DFA):

$$\mathcal{A} = (Q, \varSigma, f) \tag{1}$$

where $Q = \{q_1, q_2, \ldots, q_m\}$ is a finite set of the state, $\varSigma$ is a finite set of the input, and $f : Q \times \varSigma \to Q$ is a transition function. For simplicity of notation, the initial state and the final state are omitted. Moreover, $q_i \in Q$ and $\sigma \in \varSigma$ are called a node and an original input, respectively. As a simple example, consider the DFA of Fig. 1 (a). In this example, Q, $\varSigma$ and f are given by

$$\begin{cases} Q = \{q_1, q_2, q_3, q_4\}, \\ \varSigma = \{0, 1\}, \\ f(q_1, 0) = q_1, \quad f(q_1, 1) = q_2, \\ f(q_2, 0) = q_4, \quad f(q_2, 1) = q_4, \\ f(q_3, 0) = q_1, \quad f(q_3, 1) = q_1, \\ f(q_4, 0) = q_3, \quad f(q_4, 1) = q_4. \end{cases} \tag{2}$$

By means of our approach in [9,10], the DFA of (1) can be expressed by the form of the linear state equation with linear-type inequalities. This derivation procedure is briefly explained for the example of Fig. 1 (a) as follows. See Appendix A for the general case.

Consider a state sequence satisfying the DFA of Fig. 1 (a). For a binary variable δ_{ij}, if the k-th state of the sequence is q_i and the $(k+1)$-th state is q_j, then $\delta_{ij}(k) = 1$, otherwise $\delta_{ij}(k) = 0$. In other words, a binary variable δ_{ij} is assigned to the arc from node q_i to node q_j such as Fig. 1 (b). Note here that k may be regarded as discrete time in control of hybrid systems. Then we focus on the input-output relation of arcs at each node of Fig. 1 (b). For example, from Fig. 1 (b), the input-output relation at node q_1 is represented by the equation

$$\delta_{11}(k+1) + \delta_{12}(k+1) = \delta_{11}(k) + \delta_{31}(k).$$

By expressing the input-output relation at every node in a similar way, the DFA of Fig. 1 (b) can be expressed as the following discrete-time implicit system model with an equality constraint on the initial state:

$$E\xi(k+1) = F\xi(k), \quad e_6^T \xi(0) = 1 \tag{3}$$

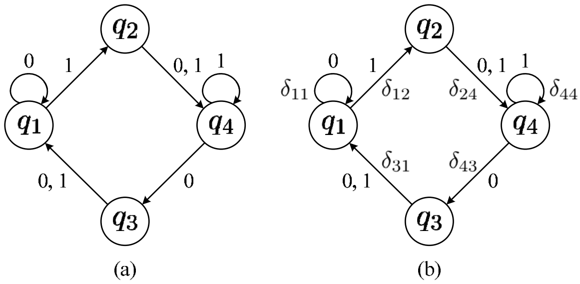


Fig. 1 Simple example of 4-node DFA. (a) Given DFA. (b) Equivalent DFA assigning a binary variable to each arc.

where $\xi = [\ \delta_{11} \quad \delta_{12} \quad \delta_{24} \quad \delta_{44} \quad \delta_{43} \quad \delta_{31}\ ]^T$,

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

Furthermore, by some coordinate transformations, the implicit model (3) can be equivalently transformed into the following state equation:

$$\begin{cases} x(k+1) = \begin{bmatrix} 0\,0\,1\,0 \\ 1\,0\,0\,0 \\ 0\,0\,0\,1 \\ 0\,1\,0\,0 \end{bmatrix} x(k) + \begin{bmatrix} 1 & 0 \\ -1 & 0 \\ 0 & -1 \\ 0 & 1 \end{bmatrix} u(k), \\ -x(k) + \begin{bmatrix} 1\,0 \\ 0\,0 \\ 0\,0 \\ 0\,1 \end{bmatrix} u(k) \leq 0, \\ x(k) \in \mathbf{R}^4, \quad u(k) \in \{0,1\}^2, \\ x(0) \in \left\{ \ \zeta \in \{0,1\}^4 \mid e_4^T \zeta = 1 \ \right\} (:= \mathcal{X}_0). \end{cases} \tag{4}$$

As for the relation between the DFA (2) and the state equation (4), we can show that the k-th state of the DFA is node q_i if and only if $x_i(k) = 1$, $x_j(k) = 0$, $i \neq j$ hold for the state $x(k)$ of (4). Therefore, $x(k)$ represents the k-th state in a state sequence, and one-to-one correspondence between the state of the DFA (2) and the state of (4) holds. The binary input variable $u(k)$ of (4) expresses the k-th input of the fictitious input sequence for uniquely determining the value of the $(k+1)$-th state of the DFA.

For example, consider the transition from node q_1 at 0. Then the initial state $x(0) \in \mathcal{X}_0$ is given by

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

Since $-x_1(0)+u_1(0) \leq 0$ and $-x_4(0)+u_2(0) \leq 0$ hold from the inequality condition in (4), $u_1(0)$ is a free binary variable, and $u_2(0) = 0$ holds. Suppose $u_1(0) = 0$.

Then from the state equation in (4)

$$x(1) = \begin{bmatrix} 0 \\ x_1(0) - u_1(0) \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \end{bmatrix}$$

holds, i.e., the state transits from q_1 to q_2. On the other hand, if $u_1(0) = 1$, then

$$x(1) = \begin{bmatrix} u_1(0) \\ 0 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

holds, i.e., the state stays q_1. In both cases, $x(1) \in \mathcal{X}_0$ holds. Similar and iterative discussion proves $x(k) \in \mathcal{X}_0$ for all k, which implies that one of elements of $x(k)$ is 1, and the others are all 0. Thus, together with the initial state condition $x(0) \in \mathcal{X}_0$ and the binary input condition $u(k) \in \{0,1\}^2$, the inequality condition in (4) is used for guaranteeing that $x(k) \in \mathcal{X}_0$ holds for all k.

In addition, consider the relation between $\sigma(k) \in \Sigma = \{0,1\}$ and $u(k)$. For (2), suppose that the initial state and the original input sequence are given as q_1 and 0111. Then the state transition is derived as

$$q_1 \overset{0}{\to} q_1 \overset{1}{\to} q_2 \overset{1}{\to} q_3 \overset{1}{\to} q_3. \tag{5}$$

In the case that (4) is used, instead of (5), we can derive

$$\begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix} \overset{[1\ 0]^T}{\longrightarrow} \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix} \overset{[0\ 0]^T}{\longrightarrow} \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \end{bmatrix} \overset{[0\ 0]^T}{\longrightarrow} \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \end{bmatrix} \overset{[0\ 1]^T}{\longrightarrow} \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \end{bmatrix}.$$

In this way, we see that (2) and (4) are equivalent in the sense that the state sequence is the same under given initial state and input sequences. Noting that the automaton studied in this paper is deterministic, we see that the implicit system model (3) equivalently expresses a given DFA. See [10] and Appendix A for the equivalence between the implicit system model and the linear state equation. Since the state equation (4) has a form similar to one used in the standard control theory, it can be expected that analysis and synthesis are relatively easy. In fact, in this paper, a stabilizing state feedback controller will be explicitly derived for a general form of a state equation expressing a DFA (1), given as

$$\begin{cases} x(k+1) = Ax(k) + Bu(k), \\ Cx(k) + Du(k) \le G, \\ x(k) \in \mathbf{R}^m, \quad u(k) \in \{0,1\}^{\hat{\alpha}}, \\ x(0) = x_0 \in \mathcal{X}_0 \end{cases} \tag{6}$$

where the input dimension $\hat{\alpha}$ is determined by the derivation procedure in Appendix A, and $\mathcal{X}_0$ is defined newly by replacing the dimension of the space in $\mathcal{X}_0$ of (4) by n.

Remark 1 The model predictive control problem of hybrid systems is in general reduced to a mixed integer quadratic programming (MIQP) problem. In the case that the state equation (6) is used as a model of discrete dynamics, the state sequence of (6) from $k+1$ time to $k+N$ time are often eliminated by replacing it by the current state $x(k)$ and the input sequences in the MIQP problem. So it is necessary to regard $x(k)$ as continuous-valued variables; thus (6) is given so as to implicitly satisfy the condition of $x(k) \in \{0,1\}^m$

3 Stabilizability Problem

First, equilibrium points of a state equation (6) are defined.

Definition 1 Suppose that a state equation (6) expressing a DFA is given. Then if $x_e \in \mathcal{X}_0$, $u_e \in \{0,1\}^{\hat{\alpha}}$ satisfy $x(k+1) - x(k) = 0$, i.e.,

$$(A - I)x_e + Bu_e = 0, \tag{7}$$

then the pair (x_e, u_e) is called an equilibrium point. In addition, the node corresponding to (x_e, u_e) is called an equilibrium node.

For example, in Fig. 1, equilibrium points are derived as

$$(x_e, u_e) = \underbrace{\left(\begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 1 \\ 0 \end{bmatrix} \right)}_{\text{Node } q_1}, \underbrace{\left(\begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}, \begin{bmatrix} 0 \\ 1 \end{bmatrix} \right)}_{\text{Node } q_4}.$$

From Definition 1, equilibrium nodes correspond to nodes that have the self-loop. In the case of Fig. 1, equilibrium nodes are q_1 and q_4. In other words, the DFA without self-loops has no equilibrium nodes.

For fixed (x_e, u_e), by defining $\tilde{x}(k) := x(k) - x_e$ and $\tilde{u}(k) := u(k) - u_e$, (6) is equivalently rewritten as

$$\begin{cases} \tilde{x}(k+1) = A\tilde{x}(k) + B\tilde{u}(k), \\ C\tilde{x}(k) + D\tilde{u}(k) \le \tilde{G}, \\ \tilde{x}(k) \in \mathbf{R}^m, \quad \tilde{u}(k) \in \tilde{\mathcal{U}}, \\ \tilde{x}(0) = \tilde{x}_0 \in \tilde{\mathcal{X}}_0 \end{cases} \tag{8}$$

where $\tilde{G} := G - Cx_e - Du_e$, $\tilde{\mathcal{X}}_0 := \{\ \zeta - x_e,\ \zeta \in \{0,1\}^m \mid e_m^T \zeta = 1\}$ and $\tilde{\mathcal{U}} := \{\ \eta - u_e,\ \eta \in \{0,1\}^{\hat{\alpha}}\}$. In the case of the equilibrium node q_4 of Fig. 1, $\tilde{x}_0$ is given by

$$\tilde{x}_0 \in \tilde{\mathcal{X}}_0 = \left\{ \begin{bmatrix} 1 \\ 0 \\ 0 \\ -1 \end{bmatrix}, \begin{bmatrix} 0 \\ 1 \\ 0 \\ -1 \end{bmatrix}, \begin{bmatrix} 0 \\ 0 \\ 1 \\ -1 \end{bmatrix}, \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \right\}.$$

It is remarked here that since $x(k) \in \mathcal{X}_0$ holds for all k in (6) (see [9,10]), $\tilde{x}(k) \in \tilde{\mathcal{X}}_0$ also holds for all k.

Next, we define the notion of stabilizability of a DFA. Hereafter, a sequence $x(0), x(1), \ldots$ in (6) (or equivalently, a sequence $q_0 \to q_1 \to \cdots$ in (1)) are called a state transition trajectory. The minimality of the state transition trajectory of a DFA is defined in preparation.

Definition 2 For a given DFA, and fixed nodes q_0, q_f, let $\mathcal{T}(q_0, q_f)$ denote the set of all state transition trajectories from node q_0 to node q_f. Then a state transition trajectory from node q_0 to node q_f is called a minimal state transition trajectory if the number of arcs in this trajectory is minimal among all trajectories in $\mathcal{T}(q_0, q_f)$.

For example, in Fig 1, consider the state transition trajectories from node q_3 to q_2. The state transition trajectory $q_3 \to q_1 \to q_2$ is minimal, and the state transition trajectory $q_3 \to q_1 \to q_1 \to q_2$ is not minimal. It is obvious that a minimal state transition trajectory is not always unique.

Under the above preparations, the notion of stabilizability via a state feedback controller $u(k) = Kx(k)$ of a DFA is defined as follows.

Definition 3 Suppose that a state equation (6) expressing a DFA is given. If there exists a state feedback controller $u(k) = Kx(k)$ such that the state transition trajectory from every initial node to the target equilibrium node is minimal, then the DFA is said to be stabilizable at the target equilibrium node. Furthermore, the corresponding controller is called a stabilizing state feedback controller.

The notion of Definition 3 is similar to that of [5]. However, in [5], an explicit form of stabilizing state feedback controller is not derived. Note that the stabilizability property of Definition 3 depends on the target equilibrium node. See Section 5 for examples.

Definition 3 straightforwardly provides the following fact.

Theorem 1 *Suppose that a DFA with at least one equilibrium node is given. Then it is stabilizable at the target equilibrium node if and only if every node is connected to the target equilibrium node.*

Proof First, the proof of the necessity is trivial from Definition 3. Next, the sufficiency is proven. Assume that every node is connected to the target equilibrium node. Then there exists at least one minimal trajectory from every initial node to the target equilibrium node. Furthermore, from $x(k) \in \mathcal{X}_0$, each column of a state feedback gain K can be independently determined for each state. Therefore, by appropriately determining K, a given DFA is stabilizable at the target equilibrium node. □

From Theorem 1, we see that the stabilizability condition depends on only the connectivity among nodes of a given DFA.

Then the following stabilizability problem is considered.

Problem 1 For a given target equilibrium node, consider a state equation (6) that expresses a DFA satisfying the stabilizability condition of Theorem 1. Then find all state feedback controllers of the form $u(k) = Kx(k)$ stabilizing (6) at the target equilibrium node, where K is some constant matrix.

In the next section, a stabilizing gain K in Problem 1 will be characterized by the following two steps:

(1) Derivation of a parametrization of all state feedback controllers $u(k) = K_b x(k)$ satisfying $x(k) \in \mathcal{X}_0$ and $u(k) \in \{0,1\}^{\hat{\alpha}}$.
(2) Derivation of all stabilizing state feedback controllers $u(k) = Kx(k)$.

Remark 2 The notions of equilibrium nodes and stabilizability can be directly defined for the DFA (1). However in this paper, based on linear systems theory of continuous systems, we consider deriving the stabilization method of DFAs. In the definition of stabilizability, we suppose stabilization via a state feedback controller $u(k) = Kx(k)$. Also, the definition of equilibrium points corresponds to the standard definition of linear continuous systems. Thus the definitions used in this paper accord with those of linear systems theory.

4 Derivation of All Stabilizing State Feedback Controllers

4.1 Derivation of all state feedback controllers satisfying $x(k) \in \mathcal{X}_0$ and $u(k) \in \{0,1\}^{\hat{\alpha}}$

It is difficult to analytically derive all K_b satisfying $x(k) \in \mathcal{X}_0$ and $u(k) \in \{0,1\}^{\hat{\alpha}}$. So we derive K_b by the following procedure. Note that from $x(k) \in \mathcal{X}_0$ and $u(k) \in \{0,1\}^{\hat{\alpha}}$, we have $K_b \in \{0,1\}^{\hat{\alpha}\times m}$ without loss of generality. Furthermore, we assume without loss of generality that the target equilibrium node is given by q_m, which m denotes the number of nodes.

Derivation procedure of a parametrization of all K_b satisfying $x(k) \in \mathcal{X}_0$ and $u(k) \in \{0,1\}^{\hat{\alpha}}$:
Step 1: Suppose that a state equation (6) expressing a DFA that satisfies the stabilizability condition of Theorem 1, and the target equilibrium node q_m are given. Then set $C = [\ C_1 \ \ C_2 \ \ \cdots \ \ C_m\ ]$ for $C_i \in \mathbf{R}^m$, $i = 1, 2, \ldots, m$, and set $K_b = [\ K_1^b \ \ K_2^b \ \ \cdots \ \ K_m^b\ ]$ for $K_i^b \in \{0,1\}^{\hat{\alpha}}$.
Step 2: Compute each column $C_i + DK_i^b$ of the matrix $C + DK_b$.
Step 3: Set $i = 1$, and find all K_i^b satisfying

$$C_i + DK_i^b \leq E. \tag{9}$$

Repeat this operation until $i = m - 1$.
Step 4: In the case of $i = m$, set $K_m^b = u_e$. Thus we obtain the state feedback gain K_b in question.

In Step 3, it is easy to check whether (9) holds by making a truth table for all combinations of elements of $K_i^b \in \{0,1\}^{\hat{\alpha}}$. For example, suppose in the case of Fig. 1 that q_4 is the target equilibrium node. Then we obtain the following parametrization of K_b:

$$K_b = \begin{bmatrix} k_{11} & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{10}$$

where k_{11} is a free binary parameter. Applying the obtained state feedback $u(k) = K_b x(k)$ to the state equation in (4), we obtain the closed loop system:

$$x(k+1) = \begin{bmatrix} k_{11} & 0 & 1 & 0 \\ 1 - k_{11} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 \end{bmatrix} x(k).$$

From this closed loop system, we see that for $k_{11} = 0$ the state transits from node q_1 to q_2, and for $k_{11} = 1$ the state stays node q_1. Note that if the state reaches the target equilibrium node q_4, then the state stays q_4.

Hereafter, let $\mathcal{K}_b$ express the set of all K_b obtained by the above algorithm. The set $\mathcal{K}_b$ is in general characterized by binary linear equations/inequalities using each element of K_b, $k_{ij} \in \{0, 1\}$. See Section 5 for further details.

4.2 Derivation of all stabilizing state feedback controllers

To characterize a stabilizing state feedback gain, we consider a necessary and sufficient condition for the state transition trajectory to be minimal. First, the distance from some node to the other node is defined.

Definition 4 For a given DFA, the minimal number of arcs from node q_i to q_j is called a distance from node q_i to q_j. Then the distance from the node at k (corresponding to k-th state) to the target equilibrium node is described by

$$V_d(\tilde{x}(k)) := [\ \Phi \quad 0\ ]\,\tilde{x}(k) \in \mathbf{Z}_+, \quad \Phi \in \mathbf{Z}_+^{1\times(m-1)} \tag{11}$$

where the i-th element of Φ denotes the distance from node q_i to the target equilibrium node.

Since the target equilibrium node is given by q_m, it is remarked that the $\tilde{\mathcal{X}}_0$ is given by

$$\tilde{\mathcal{X}}_0 = \left\{ \begin{bmatrix} 1 \\ 0 \\ 0 \\ \vdots \\ 0 \\ 0 \\ -1 \end{bmatrix}, \begin{bmatrix} 0 \\ 1 \\ 0 \\ \vdots \\ 0 \\ 0 \\ -1 \end{bmatrix}, \cdots, \begin{bmatrix} 0 \\ 0 \\ 0 \\ \vdots \\ 0 \\ 1 \\ -1 \end{bmatrix}, \cdots, \begin{bmatrix} 0 \\ 0 \\ 0 \\ \vdots \\ 0 \\ 0 \\ 0 \end{bmatrix} \right\},$$

i.e., the first, second, $\ldots, (m-1)$-th elements of $\tilde{x}(k) \in \tilde{\mathcal{X}}_0$ have either value of "0" or "1" (not "-1"), while m-th element has "-1" or "0". Thus $[\ \Phi \quad 0\ ]\,\tilde{x}(k) \in \mathbf{Z}_+$ holds. Although this distance does not satisfy an axion of a distance, it is called here a distance for simplicity. For example, in Fig. 1, the distance from node q_3 to q_1 is 1, but the distance from node q_1 to q_3 is 3. Furthermore, in Fig. 1, the distance from the node at k to the target equilibrium node is obtained as

$$[\ \Phi \quad 0\ ]\,\tilde{x}(k), \quad \Phi = [\ 2 \quad 1 \quad 3\ ].$$

The vector Φ can be easily derived by means of the adjacency matrix of a given DFA. Note here that the vector Φ depends on the target equilibrium node.

Exploiting (11) gives a necessary and sufficient condition for the state transition trajectory to be minimal.

Lemma 1 *Consider a state equation (6) that expresses a DFA satisfying the stabilizability condition of Theorem 1. Then, the state transition trajectory from every node to the target equilibrium node is minimal if and only if*

$$[\ \Phi \quad 0\ ]\,\tilde{x}(k+1) - [\ \Phi \quad \varepsilon\ ]\,\tilde{x}(k) \le 0, \quad \forall k \in \mathbf{Z}_+, \quad \forall \tilde{x}(k) \in \tilde{\mathcal{X}}_0 \tag{12}$$

holds, where ε is an arbitrary real number satisfying $0 < \varepsilon < 1$.

Proof We consider the nonnegative function $V_d(\tilde{x})$ of (11) in Definition 4. This function satisfies that $V(\tilde{x}) = 0$ if and only if $\tilde{x} = 0$ because $\Phi \in \mathbf{Z}_+^{1\times(m-1)}$ and $\tilde{x} \in \tilde{\mathcal{X}}_0$ hold (note that by definition if $\tilde{x}_i = 0$, $i = 1, 2, \ldots, m-1$ then $\tilde{x}_m = 0$ holds). It is also remarked that the m-th element of $\tilde{x}$ always takes "-1" if $\tilde{x} \neq 0$.

First, the necessity is proven, i.e., it is proven that (12) holds. If the state transition trajectory from every node to the target equilibrium node is minimal, then the distance between the current node to the target equilibrium node is strictly decreasing with respect to k as long as it is positive. Thus if $V_d(\tilde{x}(k)) \neq 0$, then we have

$$V_d(\tilde{x}(k+1)) - V_d(\tilde{x}(k)) = -1, \quad \tilde{x}(k) \in \tilde{\mathcal{X}}_0 \tag{13}$$

which implies that (12) holds because of $-[\ 0\ \ \varepsilon\ ]\tilde{x}(k) = \varepsilon$. On the other hand, if $V_d(\tilde{x}(k)) = 0$, then we have $\tilde{x}(k) = 0$ and $V_d(\tilde{x}(k+1)) = 0$, which implies (12).

Next, the sufficiency is proven. Condition (12) is rewritten as

$$V_d(\tilde{x}(k+1)) - V_d(\tilde{x}(k)) \leq [\ 0\ \ \varepsilon\ ]\tilde{x}(k), \quad \forall \tilde{x}(k) \in \tilde{\mathcal{X}}_0. \tag{14}$$

Thus if $\tilde{x}(k) \neq 0$, $V_d(\tilde{x}(k+1)) - V_d(\tilde{x}(k)) \leq -\varepsilon < 0$. Otherwise, $V_d(\tilde{x}(k+1)) - V_d(\tilde{x}(k)) \leq 0$ holds, which implies $\tilde{x}(k+1) = 0$, i.e., every transition trajectory in question is minimal. This completes the proof. □

For example, in Fig. 1, if the target equilibrium node is q_4, the node at k is q_1, and the node at $k+1$ is also q_1, then the left-hand side of (12) is obtained as

$$\big[[\ 2\ \ 1\ \ 3\ ]\ 0\big] \begin{bmatrix} 1 \\ 0 \\ 0 \\ -1 \end{bmatrix} - \big[[\ 2\ \ 1\ \ 3\ ]\ \varepsilon\big] \begin{bmatrix} 1 \\ 0 \\ 0 \\ -1 \end{bmatrix} = \varepsilon > 0.$$

So condition (12) is not satisfied. Furthermore, if the node at k is q_1 and the node at $k+1$ is q_2, the left-hand side of (12) is obtained as

$$\big[[\ 2\ \ 1\ \ 3\ ]\ 0\big] \begin{bmatrix} 0 \\ 1 \\ 0 \\ -1 \end{bmatrix} - \big[[\ 2\ \ 1\ \ 3\ ]\ \varepsilon\big] \begin{bmatrix} 1 \\ 0 \\ 0 \\ -1 \end{bmatrix} = -1 + \varepsilon < 0.$$

In this case condition (12) is satisfied.

Based on condition (12), next, we derive a stabilizing state feedback gain K. It is remarked here that $K \in \mathcal{K}_b$ is necessary for deriving stabilizing state feedback gains. The closed loop system via the state feedback $\tilde{u}(k) = K\tilde{x}(k)$, $K \in \mathcal{K}_b$ is given by $\tilde{x}(k+1) = (A + BK)\tilde{x}(k)$. By substituting this into (12), we obtain

$$\Big([\ \Phi\ \ 0\ ](A + BK) - [\ \Phi\ \ \varepsilon\ ]\Big)\tilde{x}(k) \leq 0, \quad \forall k \in \mathbf{Z}_+, \quad \forall \tilde{x}(k) \in \tilde{\mathcal{X}}_0. \tag{15}$$

The equality in (15) holds if and only if $\tilde{x}(k) = 0$ for all k. So (15) is equivalent to

$$\Big([\ \Phi\ \ 0\ ](A+BK) - [\ \Phi\ \ \varepsilon\ ]\Big)\tilde{x}'(k) < 0, \ \ \forall k \in \mathbf{Z}_+, \ \ \forall \tilde{x}'(k) \in \tilde{\mathcal{X}}_0' := \tilde{\mathcal{X}}_0 - \{0\}. \tag{16}$$

The following lemma will be applied to (16).

Lemma 2 *Suppose that* $a = [\ a_1 \ \ a_2\ ]$, $a_1 \in \mathbf{R}^{1\times(m-1)}$, $a_2 \in \mathbf{R}$, *are given. Then the following conditions are equivalent:*

(i) $[\ a_1 \ \ a_2\ ]\,\tilde{x}'(k) < 0, \quad \forall k \in \mathbf{Z}_+, \quad \forall \tilde{x}'(k) \in \tilde{\mathcal{X}}_0' := \tilde{\mathcal{X}}_0 - \{0\}$,

(ii) $[\ a_1 \ \ a_2\ ] \begin{bmatrix} I_{m-1} \\ -e_{m-1}^T \end{bmatrix} < 0_{1\times(m-1)}$.

Proof By definition, $\tilde{\mathcal{X}}_0'$ is given as

$$\tilde{\mathcal{X}}_0' = \left\{ \begin{bmatrix} 1 \\ 0 \\ 0 \\ \vdots \\ 0 \\ 0 \\ -1 \end{bmatrix}, \begin{bmatrix} 0 \\ 1 \\ 0 \\ \vdots \\ 0 \\ 0 \\ -1 \end{bmatrix}, \cdots, \begin{bmatrix} 0 \\ 0 \\ 0 \\ \vdots \\ 0 \\ 1 \\ -1 \end{bmatrix} \right\}. \tag{17}$$

From (17), condition (i) is equivalent to condition (ii). □

For example, in the DFA of Fig. 1, $\tilde{\mathcal{X}}_0'$ is given by

$$\tilde{x}_0' \in \tilde{\mathcal{X}}_0' = \left\{ \begin{bmatrix} 1 \\ 0 \\ 0 \\ -1 \end{bmatrix}, \begin{bmatrix} 0 \\ 1 \\ 0 \\ -1 \end{bmatrix}, \begin{bmatrix} 0 \\ 0 \\ 1 \\ -1 \end{bmatrix} \right\}.$$

Then for $a_1 = [\ a_{11} \ \ a_{12} \ \ a_{13}\ ]$, where a_{11}, a_{12} and a_{13} are appropriate real numbers, condition (i) of Lemma 2 is given by

$$\begin{cases} [\ a_{11} \ \ a_{12} \ \ a_{13} \ \ a_2\ ] \begin{bmatrix} 1 \\ 0 \\ 0 \\ -1 \end{bmatrix} < 0, \\ [\ a_{11} \ \ a_{12} \ \ a_{13} \ \ a_2\ ] \begin{bmatrix} 0 \\ 1 \\ 0 \\ -1 \end{bmatrix} < 0, \\ [\ a_{11} \ \ a_{12} \ \ a_{13} \ \ a_2\ ] \begin{bmatrix} 0 \\ 0 \\ 1 \\ -1 \end{bmatrix} < 0, \end{cases}$$

i.e.,

$$[\ a_{11} \ \ a_{12} \ \ a_{13} \ \ a_2\ ] \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ -1 & -1 & -1 \end{bmatrix} < 0_{1\times 3}.$$

This condition is (ii) of Lemma 2. By applying Lemma 2 to (16), we arrive at the following theorem.

Theorem 2 *Suppose that a state equation (6) that expresses a DFA satisfying the stabilizability condition of Theorem 1, and the set $\mathcal{K}_b$ of all state feedback gains satisfying $x(k) \in \mathcal{X}_0$ and $u(k) \in \{0,1\}^{\hat{\alpha}}$ are given. Then all stabilizing state feedback gains are given as $K \in \mathcal{K}_b$ satisfying the following condition:*

$$\Phi B_1 K \begin{bmatrix} I_{m-1} \\ -e_{m-1}^T \end{bmatrix} < \Phi(I_{m-1} - A_{11}) + (\Phi A_{12} - \varepsilon)e_{m-1}^T \tag{18}$$

where A_{11}, A_{12} and B_1 are given by

$$A = \begin{bmatrix} \underbrace{A_{11}}_{m-1} & \underbrace{A_{12}}_{1} \\ A_{21} & A_{22} \end{bmatrix} \begin{matrix} \} \; m-1 \\ \} \; 1 \end{matrix}, \quad B = \begin{bmatrix} B_1 \\ \underbrace{B_2}_{\hat{\alpha}} \end{bmatrix} \begin{matrix} \} \; m-1 \\ \} \; 1 \end{matrix}.$$

Proof Relation (16) is rewritten as

$$\begin{aligned} &\left[\Phi(A_{11} + B_1K_1 - I_{m-1}) \; \Phi(A_{12} + B_1K_2) - \varepsilon \right] \tilde{x}'(k) < 0, \\ &\forall k \in \mathbf{Z}_+, \;\; \forall \tilde{x}'(k) \in \tilde{\mathcal{X}}_0' := \tilde{\mathcal{X}}_0 - \{0\} \end{aligned}$$

where $K = [\; K_1 \;\; K_2 \;]$, $K_1 \in \{0,1\}^{\hat{\alpha}\times(m-1)}$ and $K_2 \in \{0,1\}^{\hat{\alpha}\times 1}$. Note that $\Phi(A_{12} + B_1K_2) - \varepsilon$ is a scalar. Then by means of Lemma 2, this inequality is equivalently expressed by

$$\left[\Phi B_1 K_1 \; \Phi B_1 K_2 \right] \begin{bmatrix} I_{m-1} \\ -e_{m-1}^T \end{bmatrix} < - \left[\Phi(A_{11} - I_{m-1}) \; \Phi A_{12} - \varepsilon \right] \begin{bmatrix} I_{m-1} \\ -e_{m-1}^T \end{bmatrix}$$

which is equivalent to (18). □

By the derivation procedure of Section 4.1 and Theorem 2, we can derive all stabilizing state feedback controllers. For example, in the DFA of Fig. 1, setting $\varepsilon = 0.1$ in (12), and substituting (10) and the coefficient matrices of (4) into (18), we obtain

$$[\; k_{11} + 3 \;\; 3 \;\; 3 \;] < [\; 3.9 \;\; 3.9 \;\; 3.9 \;].$$

Thus $k_{11} = 0$ holds, and we obtain the stabilizing state feedback gain

$$K = \begin{bmatrix} 0 \; 0 \; 0 \; 0 \\ 0 \; 0 \; 0 \; 1 \end{bmatrix}.$$

Applying the obtained state feedback controller to the DFA of Fig. 1, we obtain the following state transition trajectories:

the state transitions from node q_1: $q_1 \to q_2 \to q_4$,

the state transitions from node q_2: $q_2 \to q_4$,

the state transitions from node q_3: $q_3 \to q_1 \to q_2 \to q_4$

(see Fig. 2).

Comparing Fig. 2 with Fig. 1, we see that state transition trajectories are limited. In general, a state transition trajectory from each node to the target equilibrium node is not uniquely determined (see Section 5).

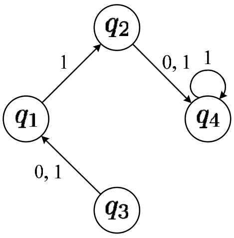


Fig. 2 Controlled 4-node DFA

5 Example

Let us consider the 7-node DFA of Fig. 3, where for simplicity of notation the label of each node expresses the index i of q_i, and the original input on each arc is omitted. The state equation expressing the DFA of Fig. 3 is derived at first. In this case, $m = 7$ holds. From the derivation procedure of Appendix A, $\hat{\alpha} = 6$ is obtained, and each coefficient matrix is obtained as

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

$$C = \begin{bmatrix} 1&0&0&0&0&0&0 \\ 0&1&0&0&0&0&0 \\ 0&0&1&0&0&0&0 \\ 0&0&0&1&0&0&0 \\ -1&-1&-1&-1&0&-1&-1 \\ -1&-1&-1&-1&-1&0&-1 \\ -1&-1&-1&-1&-1&-1&0 \\ 1&1&1&1&1&1&1 \\ -1&-1&-1&-1&-1&-1&-1 \end{bmatrix}, \quad D = \begin{bmatrix} -1&-1&-1&-1&0&0 \\ -1&0&0&-1&-1&0 \\ 0&0&0&-1&0&0 \\ 0&0&0&0&-1&-1 \\ 1&0&1&1&1&1 \\ 1&1&0&1&1&1 \\ 1&1&1&1&1&1 \\ 0&0&0&0&0&0 \\ 0&0&0&0&0&0 \end{bmatrix},$$

$$G = \begin{bmatrix} 0&0&0&0&0&0&0&1&-1 \end{bmatrix}^T.$$

Suppose that the target equilibrium node is given by node q_7. Then this DFA is stabilizable at the target equilibrium node q_7 because every node is connected to the target equilibrium node q_7. Although node q_1 is also an equilibrium node, this DFA is not stabilizable at q_1 because the state cannot transit from q_7 to q_1.

Furthermore, let us derive a stabilizing state feedback gain K. First, by using the proposed procedure of Section 4.1, a state feedback gain K_b satisfying $x(k) \in$

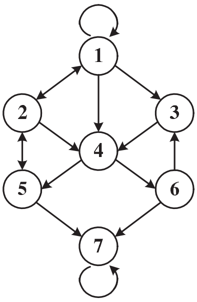


Fig. 3 7-node DFA

$\mathcal{X}_0$ and $u(k) \in \{0,1\}^{\hat{\alpha}}$ is obtained as

$$K_b = \begin{bmatrix} k_{11} & k_{12} & 0 & 0 & 0 & 0 & 0 \\ k_{21} & 0 & 0 & 0 & k_{25} & 0 & 0 \\ k_{31} & 0 & 0 & 0 & 0 & k_{36} & 0 \\ k_{41} & k_{42} & 1 & 0 & 0 & 0 & 0 \\ 0 & k_{52} & 0 & k_{54} & 0 & 0 & 0 \\ 0 & 0 & 0 & k_{64} & 0 & 0 & 0 \end{bmatrix}$$

where k_{11}, k_{21}, k_{31}, k_{41}, k_{12}, k_{42}, k_{52}, k_{54}, k_{64}, k_{25}, $k_{36} \in \{0,1\}$ must satisfy binary linear equalities

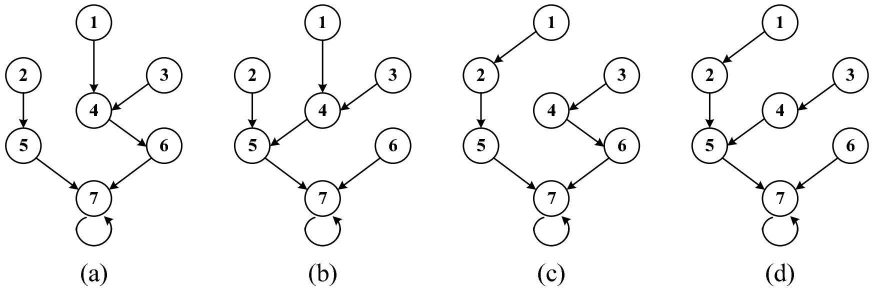
$$\begin{cases} k_{11} + k_{21} + k_{31} + k_{41} = 1, \\ k_{12} + k_{42} + k_{52} = 1, \\ k_{54} + k_{64} = 1 \end{cases} \tag{19}$$

Next, consider the inequality condition (18) of Theorem 2. In this case, (18) is obtained as

$$\begin{bmatrix} 3k_{11} + 2k_{21} + 3k_{31} + 2k_{41} \\ 3k_{12} + 2k_{42} + k_{52} \\ 2 \\ k_{54} + k_{64} \\ 2k_{25} \\ 3k_{36} \end{bmatrix}^T < \begin{bmatrix} 2.9 \\ 1.9 \\ 2.9 \\ 1.9 \\ 0.9 \\ 0.9 \end{bmatrix}^T \tag{20}$$

where $\varepsilon = 0.1$. From (19) and (20), we obtain

$$\begin{cases} k_{11} = k_{31} = 0, \quad k_{21} + k_{41} = 1, \\ k_{12} = k_{42} = 0, \quad k_{52} = 1, \\ k_{54} + k_{64} = 1, \\ k_{25} = 0, \\ k_{36} = 0 \end{cases}$$



(a) (b) (c) (d)

Fig. 4 Controlled 7-node DFA. (a) case 1: $k_{21} = 0$, $k_{54} = 0$. (b) case 2: $k_{21} = 0$, $k_{54} = 1$. (c) case 3: $k_{21} = 1$, $k_{54} = 0$. (d) case 4: $k_{21} = 1$, $k_{54} = 1$.

Thus all stabilizing state feedback gains of this system are obtained as follows:

$$K = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ k_{21} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 - k_{21} & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & k_{54} & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 - k_{54} & 0 & 0 & 0 \end{bmatrix} \tag{21}$$

where $k_{21},\ k_{54} \in \{0, 1\}$ are free parameters. From (21), we obtain four stabilizing state feedback controllers, i.e., case 1: $k_{21} = 0$, $k_{54} = 0$, case 2: $k_{21} = 0$, $k_{54} = 1$, case 3: $k_{21} = 1$, $k_{54} = 0$ and case 4: $k_{21} = 1$, $k_{54} = 1$. Then Fig. 4 shows all controlled DFA. From Fig. 4, we see that each node transits to the target equilibrium node q_7 with the minimal number of arcs.

By using stabilizing state feedback controllers, we can derive all minimal state transition trajectories. Since minimal state transition trajectories are equivalent to shortest paths to the target equilibrium node, they can be of course enumerated by the existing graph search algorithm. However, the proposed method enables us not only to derive such all trajectories in a systematic way, but also to compactly express multiple shortest paths as a linear parametrization of state feedback controllers. Furthermore, the linear state equation of closed-loop systems can be embedded as a part of a mixed logical dynamical (MLD) model [3], which is one of the standard models of hybrid systems. In the next section, the proposed stabilization method of a DFA is applied to stabilizing model predictive control (MPC) of hybrid systems with discrete dynamics.

6 Application to Stabilizing Model Predictive Control of Hybrid Systems

According to the result in [5], we consider stabilizing model predictive control of hybrid systems using the stabilization method of DFAs. First, after the notion of global asymptotic stability is defined, the problem to be studied here is formulated. Next, a construction method of control Lyapunov functions (CLFs) is proposed, and the proposed MPC law is given. Finally, we show the effectiveness of the proposed method by a numerical example.

Hereafter, the symbols x, $\tilde{x}$, u, and m (the dimension of the state in (6)) used in Sections 2–5 are replaced to x_d, $\tilde{x}_d$, u_d, and n_d, respectively.

6.1 Problem Formulation

Consider the following discrete-time piecewise affine (DT-PWA) system

$$\begin{cases} x_c(k+1) = A_{I(k)}x_c(k) + B_{I(k)}u_c(k) + a_{I(k)}, \\ I(k+1) = I_+ \text{ if } x_c(k+1) \in \mathcal{S}_{I_+} \end{cases} \tag{22}$$

where $x_c(k) \in \mathcal{X}_c \subseteq \mathbf{R}^{n_c}$ and $u_c(k) \in \mathcal{U}_c \subseteq \mathbf{R}^{m_c}$ are the state and the input, respectively. Symbols $\mathcal{X}_c$ and $\mathcal{U}_c$ are given as closed and bounded convex sets. Denote by $I(k) \in \mathcal{M} := \{1, 2, \ldots, n_d\}$ the mode of system, and suppose that mode transition constraints are given by a DFA (directed graph) with n_d nodes such as Fig. 1. Furthermore, we assume that $\mathcal{S}_I$ is the bounded convex polyhedron satisfying $\bigcup_{I \in \mathcal{M}} \mathcal{S}_I = \mathcal{X}_c$ and $\mathcal{S}_I \bigcap \mathcal{S}_J = \emptyset$ for all $I \neq J \in \mathcal{M}$. Without loss of generality, we assume that $I_e = n_d$.

To define global asymptotic stability, consider the following autonomous system ($u_c(k) = 0$)

$$\begin{cases} x_c(k+1) = A_{I(k)}x_c(k) + a_{I(k)}, \\ I(k+1) = I_+ \text{ if } x_c(k+1) \in \mathcal{S}_{I_+}, \end{cases} \tag{23}$$

and the equilibrium state $x_e \in \mathcal{S}_{I_e}$. In this paper, we use the following notion of global asymptotic stability defined in [5].

Definition 5 The DT-PWA system (23) is globally asymptotically stable at the equilibrium state $x_e \in I_e$, if for any $x_c(0) \in \mathcal{X}_c$, the following conditions hold:

(i) The state $x_c(k)$ converges to the region $\mathcal{S}_{I_e}$ in finite time,
(ii) The region $\mathcal{S}_{I_e}$ is an invariant set,
(iii) The state of the system $x_c(k+1) = A_{I_e}x_c(k)$ asymptotically converges to x_e in $\mathcal{S}_{I_e}$.

In the region $\mathcal{X}_c - \mathcal{S}_{I_e}$, the state reaches the region S_{I_e} in finite time, and the state stays S_{I_e}. In only the region $\mathcal{S}_{I_e}$, the asymptotic convergence is required.

Next, the stabilization problem is formulated. For the system, the following assumption is made.

Assumption 1 *For any $x_c \in \mathcal{S}_{I_e}$, there exists $u_c \in \mathcal{U}_c$ satisfying $A_{I_e}x_c + B_{I_e}u_c + a_{I_e} \in \mathcal{S}_{I_e}$, where node I_e is an equilibrium node in the sense of Definition 1.*

Assumption 1 means that $\mathcal{S}_{I_e}$ is an invariant set for the closed-loop system, i.e., Assumption 1 corresponds to condition (ii) in Definition 5. Then consider the following problem.

Problem 2 Suppose that for the DT-PWA system (22) satisfying Assumption 1, the target equilibrium state $x_e \in \mathcal{S}_{I_e}$ is given. Then find $u_c(k) \in \mathcal{U}_c$ such that the system (22) is globally asymptotically stable at $x_e \in \mathcal{S}_{I_e}$.

To solve this problem, we apply the MPC method using a CLF based approach.

6.2 Construction of Control Lyapunov Functions

As a preparation to solve Problem 2, a sufficient condition for the system (22) to be globally asymptotically stable at x_e is introduced based on certain two nonnegative functions [5].

Suppose that mode transition constraints are expressed by the state equation (6) described in Section 2. Then the relation between the discrete state $x_d(k)$ in (6) and the continuous state $x_c(k)$ in (22) is given as

$$[x_d^{(i)}(k) = 1] \leftrightarrow [x_c(k) \in \mathcal{S}_i], \quad i \in \mathcal{M}. \tag{24}$$

This relation can be expressed by a set of linear inequalities. See [3] for further details.

Given an equilibrium state $x_e \in \mathcal{S}_{I_e}$, consider two nonnegative functions

$$V_d(\tilde{x}_d(k)), \quad V_c(x_c(k)) \tag{25}$$

where $V_d(\tilde{x}_d(k))$ is defined by (11), and $V_c(x_c(k))$ is a nonnegative function satisfying

$$\alpha_1(\|x_c(k)\|) \leq V_c(x_c(k)) \leq \alpha_2(\|x_c(k)\|)$$

for some $\alpha_1(\cdot), \alpha_2(\cdot) \in \mathcal{K}_\infty$. Then the following theorem has been obtained in [5].

Theorem 3 *Suppose that for the DT-PWA system (22) satisfying Assumption 1, the target equilibrium state $x_e \in \mathcal{S}_{I_e}$ is given. Then the closed-loop system of (22) is globally asymptotically stable at x_e if there exists $u_c(k)$ satisfying the following conditions*

$$\begin{cases} V_d(\tilde{x}_d(k+1)) - V_d(\tilde{x}_d(k)) < 0, \quad \forall k \in \mathbf{Z}_+ \setminus \mathcal{F}, \quad \textit{if } \tilde{x}_d(k) \neq 0, \\ V_d(\tilde{x}_d(k+1)) = V_d(\tilde{x}_d(k)), \quad \forall k \in \mathcal{F} \subset \mathbf{Z}_+, \quad \textit{if } \tilde{x}_d(k) \neq 0, \\ V_d(\tilde{x}_d(k+1)) = 0, \quad \forall k \in \mathbf{Z}_+, \quad \textit{if } \tilde{x}_d(k) = 0 \end{cases} \tag{26}$$

$$V_c(x_c(k+1)) \leq \rho V_c(x_c(k)) + M_c(1 - x_d^{(I_e)}(k)), \quad \forall k \in \mathbf{Z}_+ \tag{27}$$

where $\mathcal{F} \subset \mathbf{Z}_+$ is some finite set, ρ is some constant satisfying $0 \leq \rho < 1$, and M_c is a sufficiently large scalar.

If there exists $u_c(k)$ satisfying (26) and (27), then $V_d(\tilde{x}_d(k)), V_c(x_c(k))$ in (25) are called a discrete CLF and a continuous CLF, respectively.

6.3 Addition of Self-Loops to Stabilized DFAs

To generate the control input satisfying (26) and (27), the stabilization method of DFAs described in Sections 3–4 is applied. However, this method cannot be directly applied because $V_d(\tilde{x}_d(k+1)) = V_d(\tilde{x}_d(k))$ in (27) hold only for $\tilde{x}_d(k) = 0$ (see (13) and (14)). In other words, self-loops except for the target equilibrium node are eliminated (see Fig. 1 and Fig. 2, or see Fig. 3 and Fig. 4). So it is necessary to add self-loops to stabilized DFAs. Adding self-loops is easy.

As an example, consider the DFA in Fig. 1. The stabilized DFA and its state equation are obtained as Fig. 2 and

$$\begin{cases} x_d(k+1) = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 \end{bmatrix} x_d(k), \\ x_d(k) \in \mathbf{R}^4, \\ x_d(0) = x_{d0} \in \mathcal{X}_d := \left\{ \zeta \in \{0,1\}^4 \mid e_4^T \zeta = 1 \right\}, \end{cases} \tag{28}$$

respectively. Since the self-loop of node 1 is eliminated in Fig. 2, we add this self-loop to Fig. 2. Then the state equation with the self-loop of node 1 is obtained as

$$\begin{cases} x_d(k+1) = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 \end{bmatrix} x_d(k) + \begin{bmatrix} 1 \\ -1 \\ 0 \\ 0 \end{bmatrix} u_d(k), \\ -x_d(k) + \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix} u_d(k) \le 0_{4\times 1}, \\ x_d(k) \in \mathbf{R}^4, \quad u_d(k) \in \{0,1\}^1, \\ x_d(0) = x_{d0} \in \mathcal{X}_d := \left\{ \zeta \in \{0,1\}^4 \mid e_4^T \zeta = 1 \right\}. \end{cases} \tag{29}$$

The inequalities in (29) guarantee $u_d(k) = 0$ except for the case of $x_d^{(1)}(k) = 1$ and $x_d^{(i)}(k) = 0$, $i \neq 1$. In other words, $u_d(k)$ is a free parameter only in the case of $x_d^{(1)}(k) = 1$. If $u_d(k) = 1$, then the mode implies node (mode) 1. If $u_d(k) = 0$, then the mode transits from node 1 to 2. However, in the state equation (29), the mode may remain node 1, which is not the equilibrium.

To avoid such a phenomenon, inequality constraints are imposed. By $\mathcal{L}$ denote a set of nodes that have a self-loop and are not the target equilibrium node n_d (i.e., I_e in Problem 2). That is, $\mathcal{L} \subseteq \mathcal{M} - \{n_d\}$ holds. Then consider the following inequality:

$$\sum_{i=0}^{N} x_d^{(j)}(i) \le p_j, \quad j \in \mathcal{L}, \quad \sum_{j \in \mathcal{L}} p_j = |\mathcal{F}| \tag{30}$$

where $\mathcal{F} \subset \mathbf{Z}_+$ is a finite set (see Theorem 3), and $p_j, j \in \mathcal{L}$ are determined based on continuous dynamics. From this inequality, (26) follows. Finally, a new DFA obtained by adding self-loops to a stabilized DFA is called an S-DFA.

6.4 Model Predictive Control Law

We derive the MPC law to generate the input $u_c(k)$ satisfying (26) and (27). The derived MPC law is an improved version of the MPC law proposed in [5].

First, consider the DT-PWA system (22). Assume that mode transition constraints are given by the S-DFA obtained from a given DFA. Then the DT-PWA system with the S-DFA is represented as an MLD model

$$\begin{cases} x(k+1) = Ax(k) + Bv(k), \\ Cx(k) + Dv(k) \leq G \end{cases} \tag{31}$$

where $x(k) \in \mathbf{R}^{n_c} \times \{0,1\}^{n_d}$ is the state $v(k) = [\ u^T(k) \quad z^T(k) \quad \delta^T(k)\ ]^T$, $u(k) \in \mathbf{R}^{m_c} \times \{0,1\}^{m_d}$ is the input, $z(k) \in \mathbf{R}^{m_1}$ is the auxiliary continuous variable, and $\delta(k) \in \{0,1\}^{m_2}$ is the auxiliary binary variable. See [3] for further details.

Next, we derive an MPC law. For simplicity of discussion, we assume that $x_e = 0$. The cost function is given by the standard quadratic function

$$J(x(t), v(i)) = \sum_{i=t}^{t+N-1} \left\{ x^T(i)Qx(i) + v^T(i)Rv(i) \right\} + x^T(t+N)Q_f x(t+N)$$

where t is current time, and $Q, Q_f, R \geq 0$. Note here that the cost function is not necessarily required in the stabilization problem.

We thus give an MPC law deriving by the following two steps:

[Offline Procedure]
Derive a S-DFA from a given DFA, and determine p_j in (30).

[Online Procedure]
Step 1: Set $t = 0$, and give the initial state $x(t) \in \mathcal{S}_{I(t)}$.
Step 2: If $I(t) \in \mathcal{L}$, then set $p_{I(t)} - 1 \to p_{I(t)}$ in (30).
Step 3: Solve the following finite-time optimal control problem:

$$\begin{aligned} \text{find} \quad & v(k), \quad k = t, t+1, \ldots, t+N-1, \\ \min \quad & J(x(t), v(k)), \\ \text{subject to} \quad & \text{MLD model (31)}, \\ & \text{Inequality constraints (27), (30).} \end{aligned}$$

Step 4: Apply only $u_c(t)$ to the plant.
Step 5: Set $t + 1 \to t$, measure $x(t)$, and return to Step 2.

By simple calculations, the finite-time optimal control problem is rewritten as an MIQP problem. In addition, since the finite-time optimal control problem is solved repeatedly, p_j in (30) must be updated. Then the following theorem is obtained straightforwardly according to the result in [5].

Theorem 4 *The closed-loop system of (22) is globally asymptotically stable at* $x_e = 0$ *if for any* $x_c(0) \in \mathcal{X}_c$*, the finite-time optimal control problem at each time is feasible.*

Feasibility in the state set $\mathcal{X}_c$ can be checked by using, e.g., the bisimulation technique [1].

Remark 3 The MPC law described in [5] consists of only online procedure. In other words, an S-DFA is computed online there. Our proposed method is more suitable than this existing method from the computational viewpoint, since an S-DFA can be computed offline.

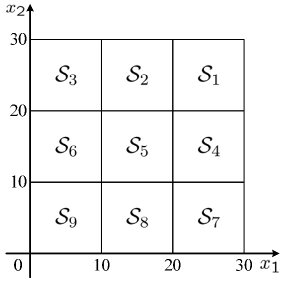


Fig. 5 State partition

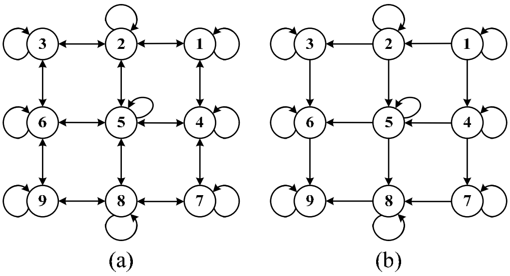


Fig. 6 Mode transition constraints. (a) Given DFA, (b) S-DFA.

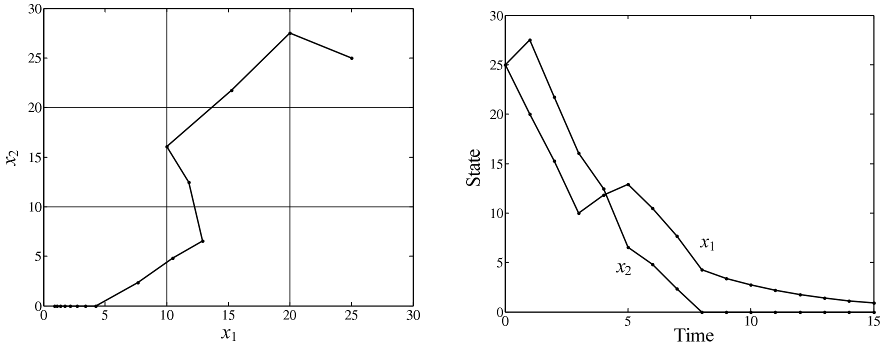
6.5 Numerical Example

As a numerical example, consider the 9-mode and 2nd-order DT-PWA system, where

$$A_1 = \begin{bmatrix} 0.3 & 0.4 \\ 0.3 & 0.7 \end{bmatrix}, \quad A_2 = \begin{bmatrix} 1.2 & -0.3 \\ -0.4 & 1.1 \end{bmatrix}, \quad A_3 = \begin{bmatrix} -0.7 & -0.2 \\ 0 & 1.2 \end{bmatrix},$$
$$A_4 = \begin{bmatrix} 1.0 & 0.1 \\ 0 & 1.0 \end{bmatrix}, \quad A_5 = \begin{bmatrix} 1.2 & 0.3 \\ -0.5 & 1.4 \end{bmatrix}, \quad A_6 = \begin{bmatrix} -0.2 & -0.4 \\ 0 & 0.1 \end{bmatrix},$$
$$A_7 = \begin{bmatrix} 1.0 & 0.3 \\ 0 & 1.0 \end{bmatrix}, \quad A_8 = \begin{bmatrix} 1.3 & -0.2 \\ 0.2 & 1.1 \end{bmatrix}, \quad A_9 = \begin{bmatrix} 1.0 & 0.2 \\ 0.2 & 1.0 \end{bmatrix},$$
$$B_i = \begin{bmatrix} 1 \\ 1 \end{bmatrix}, \quad a_i = \begin{bmatrix} 0 \\ 0 \end{bmatrix},$$

and $\mathcal{X}_c = [\, [\, 0 \;\; 0\,]^T, [\, 30 \;\; 30\,]^T \,], \mathcal{U}_c = [-5, +5]$. Fig. 5 and Fig. 6 (a) show the state partition and the mode transition constraints, respectively, where $x_c = [\, x_1 \;\; x_2\,]^T$. Furthermore, the target equilibrium state is given by $x_e = 0 \in \mathcal{S}_9$ ($I_e = n_d = 9$).

First, consider mode transition constraints. Fig. 6 (b) shows the obtained S-DFA. Then $\mathcal{L}$ in (30) is given by $\mathcal{L} = \{1, 2, 3, 4, 5, 6, 7, 8\}$. In addition, p_j is given as $p_j = 2$ (i.e., $|\mathcal{F}| = 16$). Next, consider the finite-time optimal control problem in the online procedure. Then $N = 5$, $Q = Q_f =$ block-diag$(10I_2, 0_{9\times 9})$, and $R =$ block-diag$(1, 0_{(m_d+m_1+m_2)\times(m_d+m_1+m_2)})$ are given. Furthermore, $V_c(x_c(k))$ in (25) is given by $V_c(x_c(k)) = \|x_c(k)\|_1 = |x_c^{(1)}(k)| + |x_c^{(2)}(k)|$, and ρ, M_c are



(a) (b)

Fig. 7 Obtained trajectories. (a) State space trajectory, (b) State trajectory.

given as $\rho = 0.8$, $M_c = 100$, respectively. In addition, the initial state is given as $x_c(0) = [\ 25 \ \ 25\ ]^T$.

In Fig. 7 (a) and (b) the state trajectory of the closed system is shown. From Fig. 7 (a), we see that the state $x_c(k)$ converges to the region $\mathcal{S}_9$ in finite time. Furthermore, from Fig. 7 (b), it turns out that the state $x_c(k)$ asymptotically converges to $x_e = 0$ in the region $\mathcal{S}_9$.

Finally, we discuss the computation time to solve the MPC problem. Thus we solved the MIQP problem at each time of the time period $[0, 14]$. Then the worst-case computation time was 0.2069[sec], and the mean computation time was 0.0449[sec], where we used ILOG CPLEX 11.0 [19] as an MIQP solver on the computer with the Intel Core 2 Duo 3.0GHz processor and the 4GB memory. On the other hand, the stabilization method using the terminal constraint [3] is well known as one of the existing methods. In order to compare the proposed method with the existing method, we consider to impose the terminal constraint $x_c(t+N) = 0$. Note here that mode transition constraints are given as Fig. 6 (a) in the existing method. If the main purpose is to stabilize a given system, then it is desirable to set N to be small. However, the MIQP problem with $x_c(t+N) = 0$ is infeasible in $N = 5$. If $N \geq 8$, then this problem is feasible. As previously shown in [5] by numerical examples, a longer N is needed for feasibility, with respect to the case with terminal constraint. This is the case also for the example discussed here. Also, since a longer N is needed in the existing method, the computation time in the existing method is longer than that in the proposed method. Furthermore, even if N is the same in the proposed and the existing methods, the computation time is different. This is because mode sequences are limited in the proposed method. For example, in the case of $N = 10$, the worst-case computation time was 1.0015[sec] (the proposed method), 1.9716[sec] (the existing method), and the mean computation time was 0.1668[sec] (the proposed method), 0.3946[sec] (the existing method). In the case of $N = 20$, the worst-case computation time was 4.3926[sec] (the proposed method), 18.8641[sec] (the existing method), and the mean computation time was 0.6695[sec] (the proposed method), 6.4353[sec] (the existing method). Thus reducing the computation time is achieved by applying the proposed method.

7 Conclusion

In this paper, based on our previously proposed state equation representation, the design method of a stabilizing state feedback controller has been proposed for a deterministic finite automaton (DFA). In particular, we have derived a necessary and sufficient condition for stabilizability, and given a characterization of all the stabilizing state feedback gains. Furthermore, this method has been applied to the stabilization problem of hybrid systems with discrete dynamics, and the stabilizing model predictive control (MPC) law has been proposed. From the computational viewpoint, the proposed MPC law is effective.

Future works are as follows. The stabilization problem of DFAs is closely related to the shortest path problem, although the former will consider the problem of finding a shortest path for every initial state. For example, in Fig. 4, the obtained state trajectories express all shortest paths. Thus, it may be interesting to evaluate the computational amount of the proposed method comparing with the existing shortest path algorithms. Furthermore, it will be one of the challenging topics to develop the robust control theory of DFA [13] based on our framework. In control of hybrid systems, it will be significant to consider how to efficiently check feasibility of the finite-time optimal control problem.

A Derivation Procedure of State Equation Expressing Deterministic Finite Automata

The procedure deriving a state equation from a given DFA is as follows. This is a more sophisticated version of our approach derived in [10].

Procedure of deriving a state equation:
Step 1: For a given deterministic finite automata $\mathcal{A}$ with m nodes and $n(\geq m)$ arcs, let $\mathcal{I}_a$ denote the set of combinations of (i,j) such that the arc from node i to node j exists, and assign a binary variable δ_l to the arc l. Furthermore, set $\xi(k) := [\ \delta_1(k)\ \ \delta_2(k)\ \ \cdots\ \ \delta_n(k)\ ]^T \in \{0,1\}^n$. Then the input-output relation of $\delta_l(k)$ on each node gives the implicit system of

$$\Sigma_I : \begin{cases} E\xi(k+1) = F\xi(k), \\ \xi(k) \in \{0,1\}^n, \quad \xi(0) \in \Xi_0(\delta_0^M). \end{cases}$$

where $E, F \in \{0,1\}^{m\times n}$,

$$\Xi_0(\delta_0^M) := \left\{\ \eta \in \{0,1\}^n \mid e_n^T\eta = 1,\ E\eta = \delta_0^M\ \right\}$$

and $\delta_0^M \in \{0,1\}^m$ denotes a given initial mode satisfying $e_m^T\delta_0^M = 1$.
Step 2: Derive a permutation matrix P satisfying $EP = [\ I_m\ \ \tilde{E}\ ]$, where $\tilde{E} \in \{0,1\}^{m\times(n-m)}$ is some matrix. Then by using

$$\hat{V} = \begin{bmatrix} I_m & \tilde{E} \\ 0_{(n-m)\times m} & I_{n-m} \end{bmatrix} P^{-1},$$

compute $E\hat{V}^{-1} = [\ I_m\ \ 0_{m\times(n-m)}\ ]$ and

$$F\hat{V}^{-1} = \begin{bmatrix} \tilde{F}_1 & -\tilde{F}_1\tilde{E} + \tilde{F}_2 \end{bmatrix} =: \begin{bmatrix} \hat{A} & \hat{B} \end{bmatrix}$$

where $[\ \tilde{F}_1\ \ \tilde{F}_2\ ] := FP$. Thus letting $[\ x^T(k)\ \ \hat{u}^T(k)\ ]^T := \hat{V}\xi(k)$, the state equation with inequality constraints is obtained as

$$\begin{cases} x(k+1) = \hat{A}x(k) + \hat{B}\hat{u}(k), \\ -x(k) + \tilde{E}\hat{u}(k) \leq 0, \\ x(k) \in \mathbf{R}^m, \quad \hat{u}(k) \in \{0,1\}^{n-m}, \\ x(0) = x_0 \in \mathcal{X}_0 := \{\ \zeta \in \{0,1\}^m \mid e_m^T\zeta = 1\ \}. \end{cases} \tag{32}$$

If $\hat{B}$ is full row rank, then (32) with $u(k) := \hat{u}(k)$ is the state-equation-based model to be found. Otherwise, go to Step 3.
Step 3: Reduce the matrix $\hat{B}$ to

$$\hat{B} = P_B \begin{bmatrix} I_{\hat{\alpha}} & 0 \\ \tilde{B} & 0 \end{bmatrix} T_B \tag{33}$$

where $\hat{\alpha} := \mathrm{rank}\hat{B}$, P_B is a permutation matrix, T_B is a nonsingular matrix, and $\tilde{B}$ is some matrix. Next, define

$$\begin{bmatrix} \tilde{u}(k) \\ \tilde{u}_e(k) \end{bmatrix} := T_B \hat{u}(k) \tag{34}$$

where $\tilde{u}_e(k)$ denotes redundant input variables. Then applying the input transformation

$$\tilde{u}(k) = \hat{A}_u x(k) + u(k) \tag{35}$$

where $\hat{A}_u := -[\ I_{\hat{\alpha}} \ \ 0_{\hat{\alpha}\times(m-\hat{\alpha})}\ ]P_B^{-1}\tilde{F}_1$ and $u(k) \in \{0,1\}^{\hat{\alpha}}$ is the binary input vector, to (32) yields

$$\begin{cases} x(k+1) = Ax(k) + Bu(k), \\ Cx(k) + Du(k) \leq G, \\ x(k) \in \mathbf{R}^m, \quad u(k) \in \{0,1\}^{\hat{\alpha}}, \\ x(0) = x_0 \in \mathcal{X}_0 \end{cases} \tag{36}$$

where

$$A := P_B \begin{bmatrix} 0 & 0 \\ -\tilde{B} & I_{m-\hat{\alpha}} \end{bmatrix} P_B^{-1}\hat{A}, \quad B := P_B \begin{bmatrix} I_{\hat{\alpha}} \\ \tilde{B} \end{bmatrix},$$

$$C := \begin{bmatrix} I_m - \Phi A \\ e_m^T A \\ -e_m^T A \end{bmatrix}, \quad D := \begin{bmatrix} -\Phi B \\ e_m^T B \\ -e_m^T B \end{bmatrix}, \quad G := \begin{bmatrix} 0_{\hat{\alpha}\times 1} \\ 1 \\ -1 \end{bmatrix},$$

and Φ is the adjacency matrix of a given finite automaton.

In Step 1, noting that $n \geq m$ holds, for a given δ_0^M, $\xi(0)$ is not uniquely determined. So we consider the set $\Xi_0(\delta_0^M)$.

In Step 2, the state equation (32) includes the inequality $-x(k) + \tilde{E}\hat{u}(k) \leq 0$. To explain this inequality, we show a very simple example. Consider the linear scalar system $x(k+1) = x(k) - u_1(k) - u_2(k)$, where $x(k+1), x(k), u_1(k), u_2(k) \in \{0,1\}$. To satisfy the binary property of $x(k+1)$, the constraint must be considered for $u_1(k), u_2(k)$. If $x(k) = 0$, then $u_1(k) = u_2(k) = 0$ must hold. If $x(k) = 1$, then we must consider only two cases: (i) $u_1(k) = 1$, $u_2(k) = 0$ and (ii) $u_1(k) = 0$, $u_2(k) = 1$. From the above discussion, the inequality constraint $-x(k)+u_1(k)+u_2(k) \leq 0$ must be imposed. This inequality corresponds to $-x(k)+\tilde{E}\hat{u}(k) \leq 0$ in (32). Furthermore, the state x implies a dependent variable, which can be determined from m equations in Σ_I. The input $\hat{u}$ implies an independent (free) variable.

In Step 3, by substituting (35) into (32) and replacing the inequality of (32) to the inequality using the adjacency matrix Φ, we obtain (36). The input transformation of (35) guarantees the binary property of the input vectors because $\tilde{u}$ itself does not always take binary values due to some transformation (34).

In addition, although the matrices P, P_B, T_B in the above procedure are not unique, P, P_B, T_B satisfying the conditions can be derived by elementary transformations of matrices, which can be easily implemented by a suitable software such as MATLAB. Note here that the dimension of u does not depend on selection of P, P_B, T_B. Note here that the computation cost of the above procedure is very small, since there does not exist iteration in all steps of the proposed procedure.

References

1. R. Alur, T. A. Henzinger, G. Lafferriere, and G. J. Pappas, Discrete Abstraction of Hybrid Systems, Proceedings of the IEEE, vol. 88, no. 7, pp. 971–984, 2000.

2. G. Batt, D. Ropers, H. de Jong, J. Geiselmann, R. Mateescu1, M. Page, and D. Schneider, Validation of qualitative models of genetic regulatory networks by model checking: analysis of the nutritional stress response in Escherichia coli Bioinformatics, vol. 21, no. 1, pp. 19–28, 2005.
3. A. Bemporad and M. Morari, Control of systems integrating logic, dynamics, and constraints, Automatica, vol. 35, pp. 407–427, 1999.
4. Y. Brave and M. Heymann, On Stabilization of Discrete-Event Processes, Proc. 28th IEEE Conf. on Decision and Control, pp. 2737–2742, 1989.
5. S. Di Cairano, M. Lazar, A. Bemporad, and W.P.M.H. Heemels, A Control Lyapunov Approach to Predictive Control of Hybrid Systems, Proc. 11th Int'l Conf. on Hybrid Systems: Computation and Control, LNCS 4981, Springer, 2008.
6. C. G. Cassandras and S. Lafortune, Introduction to Discrete Event Systems, 2nd Edition, Springer. 2008.
7. M. Chaves, T. Eissing, and F. Allgöwer, Regulation of Apoptosis via the NFκB Pathway: Modeling and Analysis, In "Dynamics on and of complex networks: applications to biology, computer science and the social sciences", N. Ganguly, A. Deutsch and A. Mukherjee (Eds.), Birkhauser Boston, 2009, pp. 19-34.
8. A. Girard, A. A. Julius, and G. J. Pappas, Approximate Simulation Relations for Hybrid Systems, Discrete Event Dynamic Systems, vol. 18, no. 2, pp. 163–179, 2008.
9. K. Kobayashi and J. Imura, Modeling of Discrete Dynamics for Computational Time Reduction of Model Predictive Control, Proc. 17th Int'l Symp. on Mathematical Theory of Networks and Systems, pp. 628–633, 2006.
10. K. Kobayashi and J. Imura, Minimality of Finite Automata Representation in Hybrid Systems Control, Proc. 10th Int'l Conf. on Hybrid Systems: Computation and Control, *LNCS* 4416, pp. 343–356, Springer, 2007.
11. R. Kumar, V. K. Garg, and S. I. Marcus, Language Stability and Stabilizability of Discrete Event Dynamical Systems, SIAM Journal on Control and Optimization, vol. 31, no. 5, pp. 1294–1320, 1993.
12. M. Lazar, Flexible Control Lyapunov Functions, 2009 American Control Conference, pp. 102–107, 2009.
13. A. Megretski, Robustness of Finite State Automata, Multidisciplinary Research in Control: The Mohammed Dahleh Symp., pp. 147-160, 2002.
14. C. M. Ozveren, A. S. Willsky, and P. J. Antsaklis, Stability and Stabilizability of Discrete Event Dynamic Systems, Journal of the Association for Computer Machinery, vol. 38, no. 3, pp. 730–752, 1991.
15. E. D. Sontag, A Lyapunov-like characterization of asymptotic controllability, SIAM Journal of Control and Optimization, vol. 21, no. 3, pp. 462–471, 1983.
16. D. C. Tarraf, M. A. Dahleh, and A. Megretski, Stability of Deterministic Finite State Machines, Proc. American Control Conf., pp. 3932–3936, 2005.
17. D. C. Tarraf, A. Megretski, and M. A. Dahleh, A Framework for Robust Stability of Systems Over Finite Alphabets, IEEE Trans. on Automatic Control, vol. 54, no. 5, pp. 1133–1146, 2008.
18. Y. Tazaki and J. Imura, Bisimilar Finite Abstractions of Interconnected Systems, Proc. 11th Int'l Conf. on Hybrid Systems: Computation and Control, LNCS 4981, Springer, 2008.
19. http://www.ilog.com/cplex/