JAIST Repository
https://dspace.jaist.ac.jp/

Title	Deterministic finite automata representation for model predictive control of hybrid systems
Author(s)	Kobayashi, Koichi; Imura, Jun-ichi
Citation	Journal of Process Control, 22(9): 1670-1680
Issue Date	2012-07-25
Type	Journal Article
Text version	author
URL	http://hdl.handle.net/10119/11473
Rights	NOTICE: This is the author's version of a work accepted for publication by Elsevier. Changes resulting from the publishing process, including peer review, editing, corrections, structural formatting and other quality control mechanisms, may not be reflected in this document. Changes may have been made to this work since it was submitted for publication. A definitive version was subsequently published in Koichi Kobayashi and Jun-ichi Imura, Journal of Process Control, 22(9), 2012, 1670-1680, http://dx.doi.org/10.1016/j.jprocont.2012.07.003
Description	

JAPAN ADVANCED INSTITUTE OF SCIENCE AND TECHNOLOGY

Japan Advanced Institute of Science and Technology

Deterministic finite automata representation for model predictive control of hybrid systems

Koichi Kobayashi [a], Jun-ichi Imura [b],

[a] School of Information Science, Japan Advanced Institute of Science and Technology, Ishikawa 923-1292, Japan

[b] Graduate School of Information Science and Engineering, Tokyo Institute of Technology, Tokyo 152-8552, Japan

Abstract

As is well known, the computational complexity in the mixed integer programming (MIP) problem is one of the main issues in model predictive control (MPC) of hybrid systems such as mixed logical dynamical systems. Thus several efficient MIP solvers such as multi-parametric MIP solvers have been extensively developed to cope with this problem. On the other hand, as an alternative approach to this issue, this paper addresses how a deterministic finite automaton, which is a part of a hybrid system, should be expressed to efficiently solve the MIP problem to which the MPC problem is reduced. More specifically, a modeling method to represent a deterministic finite automaton in the form of a linear state equation with a smaller set of binary input variables and binary linear inequalities is proposed. After a motivating example is described, a derivation procedure of a linear state equation with linear inequalities representing a deterministic finite automaton is proposed as three steps; modeling via an implicit system, coordinate transformation to a linear state equation, and state feedback binarization. Various significant properties on the proposed modeling are also presented throughout the proofs on the derivation procedure.

Key words: hybrid systems, mixed logical dynamical systems, deterministic finite automata, model predictive control, mixed integer programming, binary property of variables.

1 Introduction

The model predictive control (MPC) method based on a mixed logical dynamical (MLD) model [3] will be one of the most useful approaches to control of hybrid systems, and many works on its applications have been reported so far; traction control [8], automotive gearbox control [4], power plant [2,10], DC-DC converter [12] and so on. One of the main issues of this approach is, however, that a mixed integer programming (MIP) problem, to which the MPC problem is reduced, cannot in general be solved in a sufficiently small time. This is also a common problem for most of the other optimal control schemes of hybrid systems [14], and is one of the main reasons to impede progress on applications to a larger class of systems. Thus various improvements on this weakness have been developed. A multi-parametric optimization approach [9,19] has been proposed by focusing on the fact that the MIP problem in MPC depends on the current state. Furthermore, several further improvements on this approach have been provided (e.g., see [13]).

As the other recent approach to reduce the computation time of the MIP problem, a logic-based solution method has been proposed in [6]. This method decomposes the constraints appearing in an MIP problem into three parts, i.e., integer-valued constraints, continuous-valued constraints, and mixed-valued constraints, to exploit a satisfiability (SAT) solver and an LP/QP solver.

⋆ The first author is supported by JSPS Grant-in-Aid for Young Scientists (B) 23760387. The second author is supported by Aihara Project, the FIRST program from JSPS, initiated by CSTP. Corresponding author K. Kobayashi. 1-1 Asahidai, Nomi, Ishikawa 923-1292, Japan. Tel. +81-761-51-1282. Fax +81-761-51-1149.

Email addresses: k-kobaya@jaist.ac.jp (Koichi Kobayashi), imura@mei.titech.ac.jp (Jun-ichi Imura).

The above works concentrate on the improvements of the algorithm for efficiently solving the MIP problem appearing in the control problem. On the other hand, the efficient modeling of hybrid systems is also very important from the viewpoint of computational issues because the worst computation time for solving the MIP problem exponentially grows with the number of binary variables. For example, it will be desirable in general that discrete dynamics such as deterministic finite (for simplicity, DF) automata are represented in terms of equality relations. To our knowledge, however, few results from the above points of view have been obtained in the previous literatures. Indeed the binary-inequality based representation is well known as a method for expressing DF automata [3,23], but it will not be desirable for the branch and bound method, which is one of the standard methods for solving the MIP problem, because the so-called relaxed problem, where some binary variables in the original MIP problem are replaced by real variables in $[0, 1]$, does not effectively work for the binary inequalities expressing DF automata within the framework of the MPC problem (see [15] for more details).

This paper thus proposes a new approach to representing a DF automaton as a linear state equation and linear inequalities with a relatively small number of free binary variables (called here binary input variables), based on the implicit system representation, i.e., the equality-based representation [1,18].

Our modeling has three steps; (i) modeling via an implicit system, (ii) coordinate transformation to a linear state equation with binary linear inequalities, and (iii) state feedback binarization. At the first step, a given DF automaton is expressed in terms of an implicit system with binary variables assigned to arcs, based on the input-output relation of arcs at each node. Then at the next step the implicit system is transformed into a linear constrained system with the binary property of variables preserved. One of the significant issues at these steps is how to find a kind of coordinate transformation on the *real number* field such that the newly defined variables still have the binary property. It is remarked that this problem is closely related to the behavioral approach [22], in particular, the behavioral approach on the finite field, which is applied to the coding theory [20,21]. However, the MIP problem appearing in the MPC problem of hybrid systems, should be described on the *real number* field in order to use the existing MIP solvers. In other words, the DF automaton should be represented in the form of a *linear* state-equation on the *real number* field with *linear* constraints including the binary variables. Hence the existing methods of the behavioral approach on the finite field cannot be directly applied to our problem because, e.g., a linear function on the finite field is in general expressed by a non-linear function on the real number field. Thus this paper will take a direct way to discuss this issue, that is, discuss it on the real number field.

The third step in the proposed modeling is to eliminate the redundant binary input variables if the linear state equation transformed according to the above steps includes them. However, it is not easy to eliminate them with the binary property preserved; thus a kind of state-feedback transformation will be exploited in our modeling.

The organization of the paper is as follows. In Section 2, the standard method and the proposed method to represent a DF automaton are briefly explained, and it is shown by a numerical example that the proposed method is more effective in view of the computation-time for solving an MPC control problem. In Section 3, a modeling method of expressing a given DF automaton as a linear state equation with linear inequalities is proposed. Section 4 provides some discussions on the meaning of binary input variables in the proposed model, and the upper bound of the number of them. In Section 5, some concluding remarks are given.

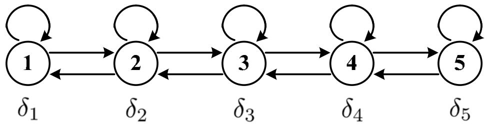
The following notation is used. Let $\{0, 1\}^{m\times n}$ express the set of $m \times n$ matrices, which consists of elements 0 and 1, and also let $\{-1, 0, 1\}^{m\times n}$ express the set of $m \times n$ matrices, which consists of elements -1, 0 and 1. Let $\mathbb{I}_n$, $0_{m\times n}$ and e_n express the $n \times n$ identity matrix, the $m \times n$ zero matrix and the $n \times 1$ vector whose elements are all one, respectively. But for simplicity of notation, we sometimes use the symbol 0 instead of $0_{m\times n}$, and the symbol $\mathbb{I}$ instead of $\mathbb{I}_n$. By $T\{0, 1\}^n$ denote the product of a matrix $T \in \mathbf{R}^{m\times n}$ and a finite set of vectors $\{0, 1\}^n$, i.e., $T\{0, 1\}^n := \{\, Ta \,|\, a \in \{0, 1\}^n \,\}$.

2 Motivating example

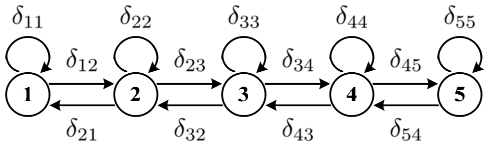
This section presents a motivating example on the modeling issues of deterministic finite (DF) automata.

As an example, we consider a simple motorbike model with five gears [5]. In this model, only the speed v (Km $\cdot$ h^{-1}) and the engine speed ω ($\times 10^2$ rpm) are considered as continuous state variables. In addition, the engine torque u_t (N $\cdot$ m) and the braking force u_b (N) are regarded as continuous control inputs. $I \in \{1, 2, 3, 4, 5\}$ denotes the I-th gear. Then the continuous dynamics are described by the following discrete-time state equation

$$x_c(k + 1) = A_c^{I(k)} x_c(k) + B_c u_c(k) \tag{1}$$



(a) Inequality-based model



(b) State-equation-based model

Fig. 1. Gear shift logic

where $x_c(k) = [\ v(k) \quad \omega(k)\ ]^T \in \mathbf{R}^2$, $u_c(k) = [\ u_t(k) \quad u_b(k)\ ]^T \in \mathbf{R}^2$,

$$A_c^{I(k)} = \begin{bmatrix} 1 - \alpha_{I(k)} & \beta_{I(k)} \\ 0 & 1 \end{bmatrix}, \quad B_c = \begin{bmatrix} 0 & -0.3 \\ 0.2 & -0.4 \end{bmatrix},$$

$\alpha_1 = 0.1$, $\alpha_2 = 0.08$, $\alpha_3 = 0.06$, $\alpha_4 = 0.04$, $\alpha_5 = 0.02$, and $\beta_1 = 0.5$, $\beta_2 = 0.4$, $\beta_3 = 0.3$, $\beta_4 = 0.2$, $\beta_5 = 0.1$. Note that $\alpha_{I(k)}, \beta_{I(k)}$ are constants that depend on the gear. The gear shift logic is given by the DF automaton in Fig. 1. For simplicity of discussion, we assume that gear shifts depend on only automaton, not continuous state variables. The system consisting of (1) and the gear shift logic in Fig. 1 is a discrete-time switched linear system, and can be expressed as a mixed logical dynamical (MLD) system model. To express this system as the MLD model, first, we focus on the modeling of the automaton in Fig. 1.

To express the automaton in Fig. 1, binary variables are frequently used. In the modeling method in [3], which is called here the inequality-based method, a binary variable δ_i is assigned to node i (mode i), and $\delta_i(k) = 1$ and $\delta_j(k) = 0$ for all $j \neq i$ hold when the mode at time k is i, which implies the equality constraint $\sum_{i=1}^{5} \delta_i(k) = 1$ (see Fig. 1 (a)). In the proposed modeling method, a binary variable δ_{ij} is assigned to the arc (directed edge) from node i to node j, with which the input-output relation at each node can be expressed (see Fig. 1 (b)).

In the inequality-based method, a DF automaton in Fig. 1 is formulated as the propositions $\delta_i(k) \to \bigvee_{j\in\mathcal{I}(i)} \delta_j(k+1)$, $i = 1, 2, \ldots, 5$, where $\to$ and $\vee$ denote implication and OR operators, respectively. In addition, $\mathcal{I}(i)$ is the index set of nodes, which are adjacent to node i, and we have $\mathcal{I}(1) = \{1, 2\}$, $\mathcal{I}(2) = \{1, 2, 3\}$, $\mathcal{I}(3) = \{2, 3, 4\}$, $\mathcal{I}(4) = \{3, 4, 5\}$, and $\mathcal{I}(5) = \{4, 5\}$. Moreover, based on the equivalence relations between propositions and linear inequalities, the above propositions can be expressed by the binary linear inequalities

$$\delta_i(k) \leq \sum_{j\in\mathcal{I}(i)} \delta_j(k+1), \quad i = 1, 2, \ldots, 5. \tag{2}$$

Thus this automaton is represented by a state equation

$$\delta_i(k+1) = u_i(k), \quad i = 1, 2, \ldots, 5 \tag{3}$$

for a new free binary variable $u_i(k)$, together with the binary linear inequalities (2) whose $\delta_i(k+1)$ are replaced by $u_i(k)$. This is called here the inequality-based model.

In the proposed method, we focus on the input-output relation at each node. For example, we consider the input-output relation at node 2. Then binary variables corresponding to input-arcs at node 2 are δ_{12}, δ_{22}, and δ_{32}. In a similar way, binary variables corresponding to output-arcs at node 2 are δ_{21}, δ_{22}, and δ_{23}. Using these binary variables, the input-output relation at node 2 is represented by the equation

$$\delta_{21}(k+1) + \delta_{22}(k+1) + \delta_{23}(k+1) = \delta_{12}(k) + \delta_{22}(k) + \delta_{32}(k).$$

Thus by expressing the input-output relation for every node in a similar way, the automaton in Fig. 1 is represented as the following discrete-time implicit system model with an equality constraint on the initial state:

$$E\xi(k+1) = F\xi(k), \quad e_{13}^T\,\xi(0) = 1 \tag{4}$$

where $\xi = [\ \delta_{11}\ \ \delta_{12}\ \ \delta_{21}\ \ \delta_{22}\ \ \delta_{23}\ \ \delta_{32}\ \ \delta_{33}\ \ \delta_{34}\ \ \delta_{43}\ \ \delta_{44}\ \ \delta_{45}\delta_{54}\ \ \delta_{55}\ ]^T$, and E and F are certain matrices. Then we can show that under a certain linear transformation of coordinates, the implicit system (4) is rewritten as the state equation with linear constraints

$$\begin{cases} x(k+1) = Ax(k) + Bu(k), \\ Cx(k) + Du(k) \leq G, \quad e_5^T x(0) = 1 \end{cases} \tag{5}$$

where $x = E\xi = [\ \delta_{11}+\delta_{12}\ \ \delta_{21}+\delta_{22}+\delta_{23}\ \ \delta_{32}+\delta_{33}+\delta_{34}\ \ \delta_{43}+\delta_{44}+\delta_{45}\ \ \delta_{54}+\delta_{55}\ ]^T \in \mathbf{R}^5$, $u = [\ \delta_{11}+\delta_{21}\ \ \delta_{12}+\delta_{22}+\delta_{32}\ \ \delta_{23}+\delta_{33}+\delta_{43}\ \ \delta_{34}+\delta_{44}+\delta_{54}\ ]^T \in \{0,1\}^4$. See Section 3.5 for the details. In this case, u is a free binary variable with $\dim u = 4$. This model is called here the state-equation-based model for brevity.

In both the inequality-based method and the proposed method, the other steps in the derivation of the MLD model are the same, and the technique described in [3] can be used. Thus a model consisting of the state equation (1), the inequality-based model (2), (3) (or the state-equation-based model (5)), and state/input constraints is represented as an MLD model of

$$\bar{x}(k+1) = \bar{A}\bar{x}(k) + \bar{B}\bar{v}(k), \tag{6}$$
$$\bar{C}\bar{x}(k) + \bar{D}\bar{v}(k) \leq \bar{G}, \tag{7}$$

where $\bar{x}(k) \in \mathbf{R}^{n_c} \times \{0,1\}^{n_d}$ is the state, $\bar{v}(k)$ is given by $\bar{v}(k) = [\ \bar{u}^T(k)\ \ \bar{z}^T(k)\ \ \bar{\delta}^T(k)\ ]^T$, $\bar{u}(k) \in \mathbf{R}^{m_{1c}} \times \{0,1\}^{m_{1d}}$ is the control input, and $\bar{z}(k) \in \mathbf{R}^{m_2}$ and $\bar{\delta}(k) \in \{0,1\}^{m_3}$ are auxiliary continuous and binary variables, respectively. In this example, we have $\bar{x} = [\ x_c^T\ \ x^T\ ]^T$, $\bar{v} = [\ u_c^T\ \ u^T\ \ \bar{z}^T\ ]^T$ ($\bar{z} \in \mathbf{R}^{10}$ is a auxiliary continuous variable), and $n_c = 2$, $n_d = 5$, $m_{1c} = 2$, $m_{1d} = 5$, $m_2 = 10$, and $m_3 = 0$ for the inequality-based model, and $m_{1d} = 4$ for the state-equation-based model; the other variables have the same values as those in the inequality-based model. For this MLD model, the finite-time optimal/model predictive control problem is considered. By a simple calculation, this control problem is reduced to a mixed integer quadratic programming (MIQP) problem with $\bar{v}(k)$, $k = 0, 1, \ldots, N-1$ as decision variables.

Numerical experiments of the finite-time optimal control problem are shown. The cost function is given as $J = \sum_{i=0}^{N-1}\left\{\hat{x}^T(i)Q\hat{x}(i) + \bar{v}^T(i)R\bar{v}(i)\right\} + \hat{x}^T(N)Q_f\hat{x}(N)$, where $\hat{x}(i) := \bar{x}(i) - x_d$, $x_d = [\ 20\ \ 20\ ]^T$ is the offset vector, and $Q = \text{block-diag}(10\mathbb{I}_{n_c}, 0)$, $R = \text{block-diag}(\mathbb{I}_{m_{1c}}, 0)$, $Q_f = \text{block-diag}(10^2\mathbb{I}_{n_c}, 0$. The initial state is given as $\bar{x}_0 = [\ 60\ \ 45 \mid 1\ \ 0\ \ 0\ \ 0\ \ 0\ ]^T$. In this experiment, the prediction horizon length is fixed as $N = 20$, and the numbers of binary variables are given as $5N = 100$ in the inequality-based model and $4N = 80$ in the state-equation-based model, respectively. Then the computation times for solving the MIQP problem are given as

161.04 (sec) for the inequality-based model,
8.48 (sec) for the state-equation-based model

where we used ILOG CPLEX 11.0 [24] as an MIQP solver on the computer with the Intel Core 2 Duo 3.0GHz processor and the 4GB memory. From this result, we see that the computation time in the case of the state-equation-based model is 20 times faster than that in the case of the inequality-based model. So the state-equation-based model is more practical for the case such as Fig. 1.

What we have to remark here is that theoretical evaluation of the computation time is difficult in the general case of MIP problems. The actual computation time depends on the number of binary and continuous variables, the number of constraints, and the structure of the cost function and constraints. Of course, the computation time heavily depends on the kind of an MIQP solver. However, it has been proven that the computational complexity exponentially grows with the number of binary variables. That is, the worst computation time exponentially grows with the number of binary variables. In addition, it will be desirable in general that discrete dynamics such as DF automata are represented in terms of equality relations such as the state equation. This is because the so-called relaxed problem, where some binary variables in the original MIQP problem are replaced by real variables in $[0, 1]$, does not effectively work for the inequality-based model (see also [15]). The relaxed problem is frequently used in MIQP solvers. Therefore, even if the difference in the number of binary variables between two models is small, then the difference in the computation time may appear.

In this paper, instead of the computation cost itself, we focus on the number of binary variables in the MIP problem. Then the state-equation-based model is theoretically proven to be effective in the sense that the number of binary variables is smaller than that in the inequality-based model (see Section 4). Thus in addition to numerical results, the state-equation-based model is also useful from the theoretical viewpoint. In view of this, this paper discusses it in detail.

3 State-equation-based modeling of deterministic finite automata

This section proposes a novel method for systematically deriving the state equation such as (5) for a general class of DF automata appeared in hybrid dynamical systems. The proposed method consists of three steps. First, a given DF automaton will be expressed as the so-called implicit system. Next, the implicit system will be transformed into a state equation via a kind of coordinate transformation. Finally, redundant input (free) variables appeared in the state equation at the second step will be excluded by a kind of input feedback transformation.

3.1 Modeling via implicit systems

In this subsection we derive an implicit system expressing a DF automaton. This paper considers the case of a DF automaton, i.e., the case in which one of nodes in the DF automaton, which expresses the mode (the discrete state) of the corresponding hybrid dynamical system, becomes active at each discrete time according to the discrete dynamics specified by the DF automaton.

Denote by the input-(output-)arc at a node the arc whose arrowhead(rear) is connected to the node. Then the following assumption is made for this automaton.

Assumption 1 *A DF automaton is given as a connected directed graph, where both ends of every arc are connected into some node(s), and every node has at least one input-arc and at least one output-arc.*

By Assumption 1, four cases as shown in Fig. 2 can be excluded. In the cases of Fig. 2 (a),(b) the solution of discrete dynamics cannot be extended after a state transition from the node in question, which implies that it is not well-posed in some sense. On the other hand, in the cases of Fig. 2 (c), (d) the arc whose end point is going into node 1 is regarded as a special arc because a state transition at this arc occurs only if the *initial* state of this dynamics is in this arc. Thus this situation can be exceptionally and easily treated in the optimal control problem of hybrid dynamical systems. Hence these four cases are excluded here, and the DF automaton satisfying Assumption 1 is denoted by $\mathcal{A}$ hereafter.

For the automaton $\mathcal{A}$, the following implicit system is considered. Let m and n denote the number of the nodes and that of the arcs in the automaton, respectively. Suppose that each arc is labeled with the index j, $j = 1, 2, \ldots, n$, and also a binary variable $\delta_j \in \{0, 1\}$, $j = 1, 2, \ldots, n$ is assigned to each arc j. Each node is also labeled with the index i, $i = 1, 2, \ldots, m$. Let $\mathcal{I}_I(i)$ and $\mathcal{I}_O(i)$ denote the index sets of input-arcs and output-arcs at node i, respectively. Then the dynamical relation between input-arcs and output-arcs at node i at each discrete time k can be expressed as

$$\sum_{j \in \mathcal{I}_O(i)} \delta_j(k+1) = \sum_{j \in \mathcal{I}_I(i)} \delta_j(k)$$

together with $\sum_{j=1}^{n} \delta_j(k) = 1$. Thus such a relation for every node yields an implicit system model given by

$$E\xi(k+1) = F\xi(k), \quad \xi(k) \in \{0,1\}^n, \quad e_n^T \xi(k) = 1 \tag{8}$$

where $\xi(k) := [\ \delta_1(k)\ \ \delta_2(k)\ \ \cdots\ \ \delta_n(k)\ ]^T \in \{0,1\}^n$ and $E,\ F \in \{0,1\}^{m \times n}$. Note that, by Assumption 1, the number of nodes is less than that of arcs, i.e., $m \leq n$. Furthermore, this implicit system has the following properties.

Lemma 2 *For the implicit system (8), the following conditions hold:*

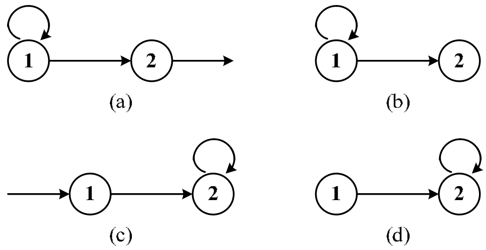


Fig. 2. Exceptional cases of DF automata

(i) *The implicit system (8) expresses all state transitions of the automaton* $\mathcal{A}$.

(ii) *The system (8) has no noncausal behavior.*

(iii) $\max_{z\in\mathbf{C}} \text{rank}\,(F - zE) = m$, *where* $\mathbf{C}$ *denotes the complex number field.*

(iv) $\text{rank}E = \text{rank}F = m$.

(v) *The following relations hold:*

$$\sum_{i=1}^{m} E^{(i)} = e_n^T, \qquad \sum_{i=1}^{m} F^{(i)} = e_n^T \tag{9}$$

where $E^{(i)} \in \{0,1\}^{1\times n}$, $F^{(i)} \in \{0,1\}^{1\times n}$, $i = 1, 2, \ldots, m$, *are the* i*-th row vector of* E *and* F, *respectively.*

(vi) *If* $\xi(0)$ *satisfies* $e_n^T\xi(0) = 1$, *then* $E\xi(k+1) = F\xi(k)$, $\xi(k) \in \{0,1\}^n$ *implies that* $e_n^T\xi(k) = 1$ *holds for all* k.

PROOF. First, (i) follows from the fact that (8) expresses all constraints in the state transition at each node by construction. Next, since the state transitions of $\mathcal{A}$ are causal, the state $\xi(k+1)$ does not depend on $\xi(\bar{k})$, $\bar{k} > k+1$, from which (ii) follows. The proof of (iii) is given as follows. For an arc $\bar{j} \in \mathcal{I}_I(\bar{i})$ at node $\bar{i}$, we have

$$\begin{cases} F_{\bar{i},\bar{j}} = 1, \\ F_{i,\bar{j}} = 0, \quad i \neq \bar{i} \end{cases} \tag{10}$$

where $F_{i,j}$ denotes the (i,j)-th element of F. This condition holds for each node $\bar{i} \in \{1, 2, \ldots, m\}$. Thus from $m \leq n$ (by Assumption 1) it follows that $\text{rank}F = m$, which implies that (iii) holds for $z = 0$ and the second relation in (iv) holds. In a similar way, the first relation in (iv) holds because for an arc $\bar{j} \in \mathcal{I}_O(\bar{i})$ at node $\bar{i}$ we have

$$\begin{cases} E_{\bar{i},\bar{j}} = 1, \\ E_{i,\bar{j}} = 0, \quad i \neq \bar{i} \end{cases} \tag{11}$$

where $E_{i,j}$ denotes the (i,j)-th element of E. Furthermore, (v) follows from (10), (11) and $m \leq n$. Finally, (vi) is proven as follows. Suppose that $e_n^T\xi(0) = 1$ and $\delta_{\bar{j}}(0) = 1$ for an arc $\bar{j} \in \mathcal{I}_I(\bar{i})$ at node $\bar{i}$. From (10), we see that the following relation holds for (8):

$$E\xi(1) = F\xi(0) = [\ 0 \ \cdots \ 0 \ \underbrace{1}_{\bar{i}} \ 0 \ \cdots \ 0\ ]^T.$$

This implies

$$\sum_{j\in\mathcal{I}_O(\bar{i})} \delta_j(1) = 1, \quad \delta_j(1) = 0,\ j \in \{1, 2, \ldots, n\}/\mathcal{I}_O(\bar{i}).$$

Hence $e_n^T\xi(1) = 1$ holds. Repeating the above discussion yields (vi). This completes the proof. □

Since (i) in Lemma 2 holds, we see that the implicit system (8) can express every mode behavior according to the discrete dynamics of an automaton $\mathcal{A}$. Conditions (ii) and (iii) are assumptions usually used for implicit systems. The noncausal behavior in (ii) corresponds to the impulsive behavior in continuous-time systems. Condition (iii) implies that there exist no redundant equations in (8), in other words, (8) has m independent equations. For example, if (8) has two same equations $\xi_1(k+1)+\xi_2(k+1)=\xi_1(k)+\xi_2(k)$, then these are dependent (see [1,18]). Condition (iv) will be used in Section 3.2. Condition (v) will be frequently used. From (vi), we see that the set of ξ satisfying $e_n^T\xi=1$ is positively invariant for the implicit system (8). Thus the implicit system (8) is equivalently rewritten as

$$E\xi(k+1)=F\xi(k),\quad \xi(k)\in\{0,1\}^n,\quad e_n^T\xi(0)=1. \tag{12}$$

Since (12) does not explicitly include the constraint $e_n^T\xi(k)=1$, it may be relatively simpler for discussion.

Next, let $\delta_i^M(k)$ denote a binary variable assigned to mode (node) i at time k, where $\delta_i^M=1$ if mode i is active, otherwise $\delta_i^M=0$, and also $\delta^M:=[\delta_1^M\ \delta_2^M\ \cdots\ \delta_m^M]^T$. Let $l_i(j)$ denote the node to which the arc $j\in\mathcal{I}_O(i)$ is connected. Then the relation between δ_j and a pair $(\delta_i^M,\ \delta_{l_i(j)}^M)$ is defined by

$$\delta_j(k):=\delta_i^M(k)\delta_{l_i(j)}^M(k+1),\quad j\in\mathcal{I}_O(i). \tag{13}$$

The following lemma provides the relation between $\xi(k)$ and $\delta^M(k)$.

Lemma 3 *For the implicit system (8), the following relations hold.*

(i) $e_m^T\delta^M(k)=1$.
(ii) $E\xi(k)=\delta^M(k)$.
(iii) The mode at time $k+1$*, i.e.,* $\delta^M(k+1)$*, is uniquely determined for a given* $\xi(k)$*.*

PROOF. First, from the definition of $\delta_i^M(k)$, (i) holds straightforwardly.

Next, (ii) is proven. Define the symbol $\eta(k)$ as $\eta(k):=E\xi(k)$. From the definition of E, the i-th element $\eta_i(k)$ of $\eta(k)$ corresponds to a sum of binary variables assigned to output-arcs at node i, i.e., $\eta_i(k)=\sum_{j\in\mathcal{I}_O(i)}\delta_j(k)$. From (13), we have

$$\eta_i(k)=\delta_i^M(k)\sum_{j\in\mathcal{I}_O(i)}\delta_{l_i(j)}^M(k+1). \tag{14}$$

If $\delta_i^M(k)=1$ holds, then $\sum_{j\in\mathcal{I}_O(i)}\delta_{l_i(j)}^M(k+1)=1$ holds from the definition of $l_i(j)$, i.e., $\eta_i(k)=1$ holds. Conversely, If $\delta_i^M(k)=0$ holds, then $\sum_{j\in\mathcal{I}_O(i)}\delta_{l_i(j)}^M(k+1)=0$ or 1 holds, i.e., $\eta_i(k)=0$ holds. This is because node $l_i(j)$ may be adjacent to other nodes except for node i. The converse also holds, i.e., if $\eta_i=0$ we have $\delta_i^M(k)=0$. This means $\eta_i(k)=\delta_i^M(k)$. Hence the proof of relation (ii) is completed.

Finally, (iii) is proven immediately from the fact that $\delta^M(k+1)=E\xi(k+1)=F\xi(k)$. This completes the proof. □

From this Lemma, we see that the initial state of this system should be carefully defined. In the optimal control problem of hybrid dynamical systems, in general, an initial mode is given. However, by Lemma 3 (iii), the value of the next mode, $\delta^M(1)$, is uniquely determined if $\xi(0)$ is given in advance. So we suppose that the initial mode $\delta_0^M\in\mathcal{M}_0$ is given as an initial state of this system, where

$$\mathcal{M}_0:=\left\{\eta\in\{0,1\}^m \mid e_m^T\eta=1\right\}.$$

Then applying the set $\Xi_0(\delta_0^M)$ defined as

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

to (12), we hereafter consider the following implicit system:

$$\Sigma_I : \begin{cases} E\xi(k+1) = F\xi(k), \\ \xi(k) \in \{0,1\}^n, \quad \xi(0) \in \Xi_0(\delta_0^M). \end{cases} \tag{15}$$

Thus we will address the problem of transforming (15) into the following form, which is regarded as a part of the MLD model:

$$\Sigma_{\text{target}} : \begin{cases} x(k+1) = Ax(k) + Bu(k), \\ Cx(k) + Du(k) \le G, \\ x(k) \in \mathbf{R}^p, \quad u(k) \in \{0,1\}^r, \\ x(0) = \Gamma\delta_0^M, \quad u(0) \in \{0,1\}^r \end{cases} \tag{16}$$

where $x(k)$ is the state, and $u(k)$ is a free binary variable, called "(binary) input variable" here, p, r denote some dimension of x, u, respectively, and $A \in \mathbf{R}^{p\times p}$, $B \in \mathbf{R}^{p\times r}$, $C \in \mathbf{R}^{s\times p}$, $D \in \mathbf{R}^{s\times r}$, $G \in \mathbf{R}^{s\times 1}$, and $\Gamma \in \mathbf{R}^{p\times m}$ are appropriate matrices. The system Σ_{target} will be often called the linear state equation for brevity, although it includes a kind of linear inequalities.

It is remarked here that the condition $x(k) \in \mathbf{R}^p$ is given in (16), instead of $x(k) \in \{0,1\}^p$. If the binary condition $x(k) \in \{0,1\}^p$ is given in (16), then it has to be explicitly added as the constraints of binary variables in the optimal control problem problem of hybrid dynamical systems, i.e., the mixed integer programming (MIP) problem. Thus the number of binary variables in the resultant MIP problem is increased. On the other hand, it can be expected that the binary condition $x(k) \in \{0,1\}^p$ will be indirectly guaranteed by some conditions based on the binary property of input variables and the initial state condition. This means that the binary variables for expressing a DF automaton in the MIP problem may be given by only binary input variables. Note also that the initial state condition does not appear in the MIP problem because the initial state value is given in advance.

3.2 Coordinate transformation to linear state equation

In this subsection, the procedure of transformation from the implicit system Σ_I into the linear state equation Σ_{target} is given.

By Assumption 1 and (iv), (v) of Lemma 2, we can show that for the implicit system Σ_I of (15) there always exists a permutation matrix $P \in \{0,1\}^{n\times n}$ satisfying $EP = [\ \mathbb{I}_m \ \ \tilde{E}\ ]$ for some matrix $\tilde{E} \in \{0,1\}^{m\times(n-m)}$. So for this permutation matrix P, we define

$$\begin{bmatrix} \tilde{F}_a & \tilde{F}_b \end{bmatrix} := FP, \quad \begin{bmatrix} \xi_a \\ \xi_b \end{bmatrix} := P^{-1}\xi$$

where $\tilde{F}_a \in \{0,1\}^{m\times m}$, $\tilde{F}_b \in \{0,1\}^{m\times(n-m)}$. It is remarked that $\xi_a \in \{0,1\}^m$ and $\xi_b \in \{0,1\}^{n-m}$ hold for $\xi \in \{0,1\}^n$, and the converse case holds as well because P^{-1} is also a permutation matrix.

First, a characterization of $\Xi_0(\delta_0^M)$ is given as follows.

Lemma 4 *For a given $\delta_0^M \in \mathcal{M}_0$, $\Xi_0(\delta_0^M)$ is expressed as*

$$\Xi_0(\delta_0^M) = \left\{ P \begin{bmatrix} \mathbb{I}_m \\ 0_{(n-m)\times m} \end{bmatrix} \delta_0^M + P \begin{bmatrix} -\tilde{E} \\ \mathbb{I}_{n-m} \end{bmatrix} z \ \middle|\ z \in \{0,1\}^{n-m},\ \delta_0^M - \tilde{E}z \ge 0 \right\}$$

where P is the permutation matrix satisfying $EP = [\ \mathbb{I}_m \ \ \tilde{E}\ ]$ for some $\tilde{E} \in \{0,1\}^{m\times(n-m)}$.

PROOF. First, it is proven that if $\xi \in \Xi_0(\delta_0^M)$, then there exists $z \in \{0,1\}^{n-m}$ satisfying $\delta_0^M - \tilde{E}z \geq 0$ and

$$\xi = P \begin{bmatrix} \mathbb{I}_m \\ 0_{(n-m)\times m} \end{bmatrix} \delta_0^M + P \begin{bmatrix} -\tilde{E} \\ \mathbb{I}_{n-m} \end{bmatrix} z. \tag{17}$$

From $EP = [\ \mathbb{I}_m \ \ \tilde{E}\ ]$ and $[\ \xi_a^T \ \ \xi_b^T\ ]^T = P^{-1}\xi$, $E\xi = \delta_0^M$ for $\xi \in \{0,1\}^n$ is rewritten as

$$\begin{bmatrix} \mathbb{I}_m & \tilde{E} \end{bmatrix} \begin{bmatrix} \xi_a \\ \xi_b \end{bmatrix} = \delta_0^M \tag{18}$$

where $\xi_a \in \{0,1\}^m$ and $\xi_b \in \{0,1\}^{n-m}$. By solving (18) with respect to $[\ \xi_a^T \ \ \xi_b^T\ ]^T$, we can see that

$$\begin{bmatrix} \xi_a \\ \xi_b \end{bmatrix} = \begin{bmatrix} \mathbb{I}_m \\ 0_{(n-m)\times m} \end{bmatrix} \delta_0^M + \begin{bmatrix} -\tilde{E} \\ \mathbb{I}_{n-m} \end{bmatrix} z$$

holds for any z. From $\xi_b \in \{0,1\}^{n-m}$, z satisfies $z \in \{0,1\}^{n-m}$. Furthermore, from $\xi_a \in \{0,1\}^m$, $\xi_a = \delta_0^M - \tilde{E}z \geq 0$ holds. Thus there exists $z \in \{0,1\}^{n-m}$ satisfying $\delta_0^M - \tilde{E}z \geq 0$ and (17).

Conversely, it is proven that every ξ satisfying (17) with $z \in \{0,1\}^{n-m}$ and $\delta_0^M - \tilde{E}z \geq 0$ is included in $\Xi_0(\delta_0^M)$. First, it follows from (17) that

$$\begin{aligned} E\xi &= EP \begin{bmatrix} \mathbb{I}_m \\ 0_{(n-m)\times m} \end{bmatrix} \delta_0^M + EP \begin{bmatrix} -\tilde{E} \\ \mathbb{I}_{n-m} \end{bmatrix} z \\ &= \begin{bmatrix} \mathbb{I}_m & \tilde{E} \end{bmatrix} \begin{bmatrix} \mathbb{I}_m \\ 0_{(n-m)\times m} \end{bmatrix} \delta_0^M + \begin{bmatrix} \mathbb{I}_m & \tilde{E} \end{bmatrix} \begin{bmatrix} -\tilde{E} \\ \mathbb{I}_{n-m} \end{bmatrix} z \\ &= \delta_0^M, \end{aligned}$$

i.e., $E\xi = \delta_0^M$ holds. Next, it is proven that

$$\delta_0^M - \tilde{E}z \in \{0,1\}^m, \tag{19}$$

$$\begin{bmatrix} e_m^T & e_{n-m}^T \end{bmatrix} \begin{bmatrix} \delta_0^M - \tilde{E}z \\ z \end{bmatrix} = 1. \tag{20}$$

It is remarked that by definition $\tilde{E} \in \{0,1\}^{m\times(n-m)}$ holds. Suppose that the i-th element of δ_0^M is 1. Then the i-th element of $\delta_0^M - \tilde{E}z$ takes the value of 0 or 1 thanks to $\delta_0^M - \tilde{E}z \geq 0$, $\tilde{E} \in \{0,1\}^{m\times(n-m)}$, and $z \in \{0,1\}^{n-m}$. On the other hand, if the i-th element of δ_0^M is 0, then the i-th element of $\delta_0^M - \tilde{E}z$ is also 0. Hence we obtain (19). Furthermore, (20) follows from $e_m^T \tilde{E} = e_{n-m}^T$ (by Lemma 2 (v)) and $e_m^T \delta_0^M = 1$.

Therefore, applying (19) and $z \in \{0,1\}^{n-m}$ into (17) yields $P^{-1}\xi \in \{0,1\}^n$, which implies $\xi \in \{0,1\}^n$, and applying (20) and $e_n^T P = e_n^T$ into (17) yields $e_n^T \xi = 1$. This means $\xi \in \Xi_0(\delta_0^M)$, which completes the proof. □

Lemma 4 provides the main result in this subsection as follows.

Theorem 5 *Consider the implicit system Σ_I of (15) for a given $\delta_0^M \in \mathcal{M}_0$, which expresses an automaton $\mathcal{A}$, and the permutation matrix $P \in \{0,1\}^{n\times n}$ satisfying $EP = [\ \mathbb{I}_m \ \ \tilde{E}\ ]$ for some matrix $\tilde{E} \in \{0,1\}^{m\times(n-m)}$. Then under the coordinate transformation of*

$$\begin{bmatrix} \hat{x} \\ \hat{u} \end{bmatrix} := \hat{V}\xi, \quad \hat{V} := \begin{bmatrix} \mathbb{I}_m & \tilde{E} \\ 0_{(n-m)\times m} & \mathbb{I}_{n-m} \end{bmatrix} P^{-1} \tag{21}$$

the system Σ_I is equivalently expressed as the following state equation:

$$\begin{cases} \hat{x}(k+1) = \hat{A}\hat{x}(k) + \hat{B}\hat{u}(k), \\ \hat{C}\hat{x}(k) + \hat{D}\hat{u}(k) \le \hat{G}, \\ \hat{x}(k) \in \mathbf{R}^m, \quad \hat{u}(k) \in \{0,1\}^{n-m}, \\ \hat{x}(0) = \hat{\Gamma}\delta_0^M, \quad \hat{u}(0) \in \{0,1\}^{n-m} \end{cases} \tag{22}$$

where $\hat{A} := \tilde{F}_a$, $\hat{B} := -\tilde{F}_a\tilde{E} + \tilde{F}_b$, $\hat{C} := -\mathbb{I}_m$, $\hat{D} := \tilde{E}$, $\hat{G} := 0_{m\times 1}$, *and* $\hat{\Gamma} := \mathbb{I}_m$.

PROOF. We will prove that under (21), (a) (22) is obtained by transforming (15) , and conversely (b) (15) is obtained by transforming (22).

Proof of (a): First, noting that

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

we obtain

$$\begin{aligned} E\hat{V}^{-1} &= EP \begin{bmatrix} \mathbb{I}_m & -\tilde{E} \\ 0_{(n-m)\times m} & \mathbb{I}_{n-m} \end{bmatrix} \\ &= \begin{bmatrix} \mathbb{I}_m & \tilde{E} \end{bmatrix} \begin{bmatrix} \mathbb{I}_m & -\tilde{E} \\ 0_{(n-m)\times m} & \mathbb{I}_{n-m} \end{bmatrix} \\ &= \begin{bmatrix} \mathbb{I}_m & 0_{m\times(n-m)} \end{bmatrix} \end{aligned} \tag{23}$$

and

$$\begin{aligned} F\hat{V}^{-1} &= FP \begin{bmatrix} \mathbb{I}_m & -\tilde{E} \\ 0_{(n-m)\times m} & \mathbb{I}_{n-m} \end{bmatrix} \\ &= \begin{bmatrix} \tilde{F}_a & \tilde{F}_b \end{bmatrix} \begin{bmatrix} \mathbb{I}_m & -\tilde{E} \\ 0_{(n-m)\times m} & \mathbb{I}_{n-m} \end{bmatrix} \\ &= \begin{bmatrix} \hat{A} & \hat{B} \end{bmatrix}. \end{aligned} \tag{24}$$

Thus by (21) $E\xi(k+1) = F\xi(k)$ of Σ_I is transformed into

$$\begin{bmatrix} \mathbb{I}_m & 0_{m\times(n-m)} \end{bmatrix} \begin{bmatrix} \hat{x}(k+1) \\ \hat{u}(k+1) \end{bmatrix} = \begin{bmatrix} \hat{A} & \hat{B} \end{bmatrix} \begin{bmatrix} \hat{x}(k) \\ \hat{u}(k) \end{bmatrix}$$

which implies $\hat{x}(k+1) = \hat{A}\hat{x}(k) + \hat{B}\hat{u}(k)$ in (22).

Next, it will be shown that the remaining relations in (22) always follow from the solution $\xi(k)$ satisfying (15). Recall that $\xi(k)$ satisfies $\xi(k) \in \{0,1\}^n$, $e_n^T\xi(k) = 1$ by Lemma 2 (vi).

Since (21) implies $\hat{x}(k) = E\xi(k)$ for any k, $\hat{x}(k) \in \mathcal{M}_0$ holds from Lemma 3. Furthermore, by applying Lemma 4 to this case, $\xi(k)$ is parametrized as

$$\xi(k) = P \begin{bmatrix} \mathbb{I}_m \\ 0_{(n-m)\times m} \end{bmatrix} \hat{x}(k) + P \begin{bmatrix} -\tilde{E} \\ \mathbb{I}_{n-m} \end{bmatrix} z(k) \tag{25}$$

for $z(k) \in \{0,1\}^{n-m}$ satisfying $\hat{x}(k) - \tilde{E}z(k) \geq 0$. By applying (25) into the transformation (21), we obtain $z = \hat{u}$, which implies $\hat{u}(k) \in \{0,1\}^{n-m}$ and $\hat{x}(k) - \tilde{E}\hat{u}(k) \geq 0$. Furthermore, $\hat{x}(0) = E\xi(0) = \delta_0^M$ holds from $\xi(0) \in \Xi_0(\delta_0^M)$. Hence we have (22).

Proof of (b): We can see that under (21) the matrices E and F in Σ_I are obtained as $E = [\ \mathbb{I}_m \ \ 0_{(n-m)\times m}\ ]\hat{V}$ and $F = [\ \hat{A} \ \ \hat{B}\ ]\hat{V}$. Thus from (22) we obtain $E\xi(k+1) = F\xi(k)$.

Next, we will prove that $\xi(k) \in \{0,1\}^n$ and $\xi(0) \in \Xi_0(\delta_0^M)$ hold from $(\hat{x}(k), \hat{u}(k))$ satisfying (22).

From (21) it follows

$$\xi(0) = P \begin{bmatrix} \mathbb{I}_m \\ 0_{(n-m)\times m} \end{bmatrix} \delta_0^M + P \begin{bmatrix} -\tilde{E} \\ \mathbb{I}_{n-m} \end{bmatrix} \hat{u}(0).$$

Thus since (22) yields $\hat{u}(0) \in \{0,1\}^{n-m}$, $\delta_0^M - \tilde{E}\hat{u}(0) \geq 0$, by Lemma 4 we have $\xi(0) \in \Xi_0(\delta_0^M)$.

In addition, by $\xi(0) \in \Xi_0(\delta_0^M)$ and $F\xi(0) = E\xi(1) = \hat{x}(1)$, Lemma 3 concludes $\hat{x}(1) \in \mathcal{M}_0$. In a similar way to the case of $k = 0$, we have $\xi(1) \in \Xi_0(\hat{x}(1))$ thanks to $\hat{u}(1) \in \{0,1\}^{n-m}$, $\hat{x}(1) - \tilde{E}\hat{u}(1) \geq 0$. Thus we can prove $\xi(k) \in \{0,1\}^n$ for any k.

This completes the proof. □

From Theorem 5, we see that under (22), only $\hat{u}(k)$ can be regarded as a binary decision variable in the MIP problem.

3.3 State feedback binarization

Since the $m \times (n-m)$ matrix $\hat{B}$ in (22) is not a column full-rank matrix in general, there may exist some redundant binary variables among variables of $\hat{u}$. So we denote the rank of $\hat{B}$ by $\hat{\alpha}$, i.e., $\hat{\alpha} := \mathrm{rank}\hat{B} (\leq \min\{m, n-m\})$. The following lemma shows the properties of $\hat{B}$, which will be used in the reduction of binary variables.

Lemma 6 *For the $m \times (n-m)$ matrix $\hat{B}$ in (22) the following statements hold:*

(i) $\hat{\alpha}$ is invariant under the coordinate transformation (21).
(ii) $\hat{B}$ is an incidence matrix of some directed graph.
(iii) There exist a permutation matrix $P_B \in \{0,1\}^{m\times m}$ and a nonsingular matrix $T_B \in \mathbf{R}^{(n-m)\times(n-m)}$ such that

$$\hat{B} = P_B \begin{bmatrix} \mathbb{I}_{\hat{\alpha}} & 0 \\ \tilde{B} & 0 \end{bmatrix} T_B \tag{26}$$

where each element of the matrix $\tilde{B}$ is an integer.

PROOF. First (i) will be proven. From $FP = [\ \tilde{F}_a \ \ \tilde{F}_b\ ]$, $\hat{B} = -\tilde{F}_a\tilde{E} + \tilde{F}_b$ is rewritten as

$$\hat{B} = FP \begin{bmatrix} -\tilde{E} \\ \mathbb{I}_{n-m} \end{bmatrix}.$$

Since permutation matrices are in general nonsingular, $\mathrm{rank}\hat{B}$ are invariant under (21).

For (ii), next, it will be proven that each column vector of $\hat{B}$ is a zero vector, or is composed of a '+1', a '−1', and $m-2$ zeros. From (9) in (v) of Lemma 2 and the permutation matrix P, each column vector of $\tilde{F}_a$, $\tilde{F}_b$ and $\tilde{E}$ consists of a '1' and $m-1$ zeros. So each column vector of $\tilde{F}_a\tilde{E}$ also consists of a '1' and $m-1$ zeros. Therefore, each column vector of $\hat{B} = -\tilde{F}_a\tilde{E} + \tilde{F}_b \in \{-1,0,+1\}^{m\times(n-m)}$ is a zero vector, or is composed of a '+1', a '−1' and $m-2$ zeros. This implies that $\hat{B}$ is the incidence matrix of some directed graph.

Finally, (iii) will be proven. Since $\hat{B} \in \{-1,0,+1\}^{m\times(n-m)}$ holds thanks to (ii), there exist permutation matrices P_1, P_2 such that $\hat{B}_{11} \in \{-1,0,+1\}^{\hat{\alpha}\times\hat{\alpha}}$ is a nonsingular matrix, and $\hat{B}_{22} = \hat{B}_{21}\hat{B}_{11}^{-1}\hat{B}_{12}$, where

$$\begin{bmatrix} \hat{B}_{11} & \hat{B}_{12} \\ \hat{B}_{21} & \hat{B}_{22} \end{bmatrix} := P_1\hat{B}P_2$$

(see e.g., [11]). Then for

$$T_0 := \begin{bmatrix} \hat{B}_{11}^{-1} & -\hat{B}_{11}^{-1}\hat{B}_{12} \\ 0 & \mathbb{I}_{n-m-\hat{\alpha}} \end{bmatrix}$$

we have

$$P_1\hat{B}P_2T_0 = \begin{bmatrix} \mathbb{I}_{\hat{\alpha}} & 0 \\ \hat{B}_{21}\hat{B}_{11}^{-1} & 0 \end{bmatrix}.$$

Thus $P_B = P_1^{-1}$, $T_B = (P_2T_0)^{-1}$, and $\tilde{B} = \hat{B}_{21}\hat{B}_{11}^{-1}$ provide a form as the same as that of (26).

Since $\hat{B}$ is an incidence matrix of some directed graph, it is totally unimodular [16]. So $P_1\hat{B}P_2$ is also totally unimodular. Since every minor determinant of a totally unimodular matrix is −1, 0, or +1, $\hat{B}_{11}^{-1} \in \{-1,0,+1\}^{\hat{\alpha}\times\hat{\alpha}}$ holds. Hence from $\hat{B}_{21} \in \{-1,0,+1\}^{(m-\hat{\alpha})\times\hat{\alpha}}$ it follows that each element of $\tilde{B} = \hat{B}_{21}\hat{B}_{11}^{-1}$ is an integer. □

From Lemma 6 (i), we see that the value of $\hat{\alpha}$ is uniquely determined for the automaton $\mathcal{A}$, independently of the coordinate transformation (21). Lemma 6 will be used in Theorem 9 and Theorem 11.

From (26) we define

$$\begin{bmatrix} \tilde{u}(k) \\ \tilde{u}_e(k) \end{bmatrix} := \begin{bmatrix} T_B^a \\ T_B^b \end{bmatrix} \hat{u}(k), \quad \begin{bmatrix} T_B^a \\ T_B^b \end{bmatrix} := T_B \tag{27}$$

where $\tilde{u}(k) \in T_B^a\{0,1\}^{n-m}$ and $\tilde{u}(k) \in T_B^b\{0,1\}^{n-m}$. If the value of each element of $\tilde{u}(k)$ is still binary, then $n-m-\alpha$ redundant (binary) input variables $\tilde{u}_e(k)$ among variables of $\hat{u}$ should be eliminated, where $P_B[\,\mathbb{I}_{\hat{\alpha}}\ \ \tilde{B}^T\,]^T$ is regarded as an input matrix. However, $\tilde{u}(k)$ is not binary variables in general. Thus $\tilde{u}(k)$ will be transformed into the binary variable by exploiting the state variables without adding new variables as follows.

As a preparation, two lemmas are shown. First, the following lemma provides the relation between the matrices E, F, and $\xi(k)$, $\delta^M(k)$, $\delta^M(k+1)$.

Lemma 7 *Consider an automaton $\mathcal{A}$. Suppose that a binary vector $\xi(k) \in \{0,1\}^n$ assigned to arcs and a binary vector $\delta^M(k) \in \{0,1\}^m$ assigned to modes are given. Then the following relation holds:*

$$E\xi(k) = diag(\delta_1^M(k), \delta_2^M(k), \dots, \delta_m^M(k))EF^T\delta^M(k+1).$$

where $\delta_i^M(k)$, $i = 1,2,\dots,m$ is the i-th element of $\delta^M(k)$.

PROOF. We use the symbol $l_i(j)$ defined for the proof of Lemma 3 again. From the definition of F, $F^T\delta^M(k+1)$ is the n-dimensional vector consisting of $\delta_{l_i(j)}^M(k+1)$, $i = 1,2,\dots,m$, $j \in \mathcal{I}_O(i)$. Then from (14) and the definition of E, we obtain this lemma. □

Next, another lemma is given as follows.

Lemma 8 *For the state variable $\hat{x}(k)$ in (22), $\hat{x}(k) = \delta^M(k)$ holds.*

PROOF. From (21) and $EP = [\ \mathbb{I}_m \ \ \tilde{E}\ ]$, we have $E\xi(k) = \hat{x}(k)$. Thus Lemma 3 (ii) completes the proof. □

Then we are in a position to complete the state-equation-based modeling.

Theorem 9 *The state equation (22) is equivalently expressed as*

$$\begin{cases} x(k+1) = Ax(k) + Bu(k), \\ Cx(k) + Du(k) \leq G, \\ x(k) \in \mathbf{R}^m, \quad u(k) \in \{0,1\}^{\hat{\alpha}}, \\ x(0) = \Gamma\delta_0^M, \quad u(0) \in \{0,1\}^{\hat{\alpha}} \end{cases} \tag{28}$$

where

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

$$C := \begin{bmatrix} \mathbb{I}_m - EF^TA \\ e_m^TA \\ -e_m^TA \end{bmatrix}, \quad D := \begin{bmatrix} -EF^TB \\ e_m^TB \\ -e_m^TB \end{bmatrix}, \quad G := \begin{bmatrix} 0_{m\times 1} \\ 1 \\ -1 \end{bmatrix}, \quad \Gamma := \mathbb{I}_m.$$

PROOF. First, we will derive the state equation, which is equivalent to (22). From Theorem 5, (15) and (22) are equivalent. So if (22) holds, then $\xi(k) \in \{0,1\}^n$, $[\ \xi_a^T(k) \ \ \xi_b^T(k)\ ]^T = P^{-1}\xi(k)$ hold, and $\xi_a(k) \in \{0,1\}^m$ is obtained. Furthermore, $\xi_a(k) = \hat{x}(k) - \tilde{E}\hat{u}(k) \in \{0,1\}^m$ holds from (25). Thus (22) is transformed into

$$\begin{cases} \hat{x}(k+1) = \hat{A}\hat{x}(k) + \hat{B}\hat{u}(k), \\ -\hat{x}(k) + \tilde{E}\hat{u}(k) = -\xi_a(k), \\ \hat{x}(k) \in \mathbf{R}^m, \quad \hat{u}(k) \in \{0,1\}^{n-m}, \quad \xi_a(k) \in \{0,1\}^m, \\ \hat{x}(0) = \hat{\Gamma}\delta_0^M, \quad \hat{u}(0) \in \{0,1\}^{n-m}. \end{cases} \tag{29}$$

Obviously (29) is transformed into (22). Hereafter we will prove that (a) (28) follows from (29), and conversely (b) (29) from (28).

Proof of (a): Suppose that (29) is given. First, we will obtain $x(k+1) = Ax(k) + Bu(k)$ in (28). For $\hat{x}(k+1) = \hat{A}\hat{x}(k) + \hat{B}\hat{u}(k)$ in (29), consider the following linear transformation of coordinates:

$$\bar{x}(k) = T\hat{x}(k), \quad T := \begin{bmatrix} \mathbb{I}_{\hat{\alpha}} & 0 \\ -\tilde{B} & \mathbb{I}_{m-\hat{\alpha}} \end{bmatrix} P_B^{-1}. \tag{30}$$

So from (26) and (27) we obtain

$$\bar{x}(k+1) = \begin{bmatrix} \bar{A}_a \\ \bar{A}_b \end{bmatrix} \bar{x}(k) + \begin{bmatrix} \mathbb{I}_{\hat{\alpha}} \\ 0 \end{bmatrix} \tilde{u}(k) \tag{31}$$

where $[\ \bar{A}_a^T \ \ \bar{A}_b^T\ ]^T := T\hat{A}T^{-1}$. Letting

$$u(k) := \tilde{u}(k) + \bar{A}_a\bar{x}(k) \tag{32}$$

(31) can be expressed as

$$\bar{x}(k+1) = \begin{bmatrix} 0 \\ \bar{A}_b \end{bmatrix} \bar{x}(k) + \begin{bmatrix} \mathbb{I}_{\hat{\alpha}} \\ 0 \end{bmatrix} u(k). \tag{33}$$

Since Theorem 5 proves that $x(k) \in \mathcal{M}_0$ holds for all k in (29), (30) implies

$$u(k) = \begin{bmatrix} \mathbb{I}_{\hat{\alpha}} & 0_{\hat{\alpha}\times(m-\hat{\alpha})} \end{bmatrix} \bar{x}(k+1) \in \{0,1\}^{\hat{\alpha}}. \tag{34}$$

Finally, transforming $\bar{x}(k)$ into $\hat{x}(k)$ via the converse transformation of (30) in (33) yields $\hat{x}(k+1) = A\hat{x}(k) + Bu(k)$. Thus the state equation in (28) together with $x(k) \in \mathbf{R}^m$, $u \in \{0,1\}^{\hat{\alpha}}$, $x(0) \in \Gamma\delta_0^M$ is obtained by using a new symbol x instead of $\hat{x}$.

Next, we will obtain the inequality in (28). Note that we have (21) and $\xi_b = \hat{u}$ by definition. So $-\hat{x}(k) + \tilde{E}\hat{u}(k) = -\xi_a(k)$ in (29) implies $\hat{x}(k) = E\xi(k)$. Thus from Lemma 7 and Lemma 8 it follows that

$$\hat{x}(k) = \text{diag}(\hat{x}_1(k), \ldots, \hat{x}_m(k))EF^T\hat{x}(k+1)$$

holds, where $\hat{x}_i$ is the i-th element of $\hat{x}$. This implies

$$\hat{x}(k) \leq EF^T\hat{x}(k+1) \tag{35}$$

thanks to $\hat{x}_i(k) \in \{0,1\}$. Consider the relation (35) with $\hat{x}$ replaced by a new symbol x. So substituting $Ax(k)+Bu(k)$ into $x(k+1)$ there yields $(\mathbb{I}_m - EF^TA)x(k) - EF^TBu(k) \leq 0$ in (28). Finally, since $e_m^T x(k+1) = 1$ holds from $x(k) \in \mathcal{M}_0$, the second inequalities in (28) are obtained. Hence we have (28).

Proof of (b): Suppose that (28) is given. First, we will prove that $x(k) \in \mathcal{M}_0$ holds in (28). Consider the following linear transformation of coordinate using the permutation matrix P_B:

$$\begin{bmatrix} \tilde{x}_a(k) \\ \tilde{x}_b(k) \end{bmatrix} = \tilde{x}(k) = P_B^{-1}x(k)$$

where $\tilde{x}_a(k) \in \mathbf{R}^{\hat{\alpha}}$, $\tilde{x}_b(k) \in \mathbf{R}^{m-\hat{\alpha}}$. Then from $x(k+1) = Ax(k) + Bu(k)$ in (28), we obtain

$$\begin{aligned} \tilde{x}_a(1) &= u(0), \\ \tilde{x}_b(1) &= \begin{bmatrix} -\tilde{B} & \mathbb{I}_{m-\hat{\alpha}} \end{bmatrix} P_B^{-1}\hat{A}P_B\tilde{x}(0) + \tilde{B}u(0). \end{aligned}$$

Noting here that $\tilde{x}(0) \leq P_B^{-1}EF^TP_B\tilde{x}(1)$, $P_B^{-1}EF^TP_B \in \{0,1\}^{m\times m}$ hold from the inequality in (28), we obtain $\tilde{x}(1) \geq 0$ from $\tilde{x}(0) = P_B^{-1}\delta_0^M \in \mathcal{M}_0$. Therefore, since $e_m^Tx(1) = e_m^T\tilde{x}(1) = 1$ holds from the inequality in (28), we should consider two cases in $u(0)$, i.e., $e_{\hat{\alpha}}^Tu(0) = 1$ and $u(0) = 0$. First, consider the case of $e_{\hat{\alpha}}^Tu(0) = 1$. Then since $\tilde{x}_b(1) = 0$ is obtained from $\tilde{x}(1) \geq 0$ and $e_m^T\tilde{x}(1) = 1$, $\tilde{x}(1) \in \mathcal{M}_0$ holds. Next, consider the case of $u(0) = 0$. First, $P_B^{-1}\hat{A}P_B\tilde{x}(0) = P_B^{-1}\hat{A}\delta_0^M \in \mathcal{M}_0$ holds from $\hat{A} = \tilde{F}_a$, $[\ \tilde{F}_a \ \ \tilde{F}_b\ ] = FP$ and Lemma 2 (v). Next, noting that each element of $-\tilde{B}$ is an integer from Lemma 6(iii) and $P_B^{-1}\hat{A}\delta_0^M \in \mathcal{M}_0$ holds, each element of $\tilde{x}_b(1)$ is given as an integer. Furthermore, from $\tilde{x}(1) \geq 0$ and $e_m^T\tilde{x}(1) = 1$, $\tilde{x}_b(1)$ is a binary vector satisfying $e_{m-\hat{\alpha}}\tilde{x}_b(1) = 1$, that is, $\tilde{x}(1) \in \mathcal{M}_0$ holds. By repeating recursively, we can prove $x(k) \in \mathcal{M}_0$ for any k.

Next, consider the linear transformation of input variables $u(k) = \bar{A}_a\bar{x}(k) + \tilde{u}(k)$ ($\bar{x} = Tx$). Since $x(k) \in \mathcal{M}_0$ holds for any k, we can set $\tilde{u}(k) \in T_B^a\{0,1\}^{n-m}$ in a similar way to the proof of (a). Furthermore, adding redundant binary input variables $\tilde{u}_e(k) \in T_B^b\{0,1\}^{n-m}$, and using a new symbol $\hat{x}$ instead of x, the state equation in (28) is transformed into

$$\begin{aligned} \hat{x}(k+1) &= \hat{A}\hat{x}(k) + P_B \begin{bmatrix} \mathbb{I}_{\hat{\alpha}} & 0 \\ \tilde{B} & 0 \end{bmatrix} \begin{bmatrix} \tilde{u}(k) \\ \tilde{u}_e(k) \end{bmatrix} \\ &= \hat{A}\hat{x}(k) + \hat{B}\hat{u}(k) \end{aligned}$$

in (29).

Finally, using a new symbol $\hat{x}$ instead of x, consider $(\mathbb{I}_m - EF^TA)\hat{x}(k) - EF^TBu(k) \leq 0$ in (28). Obviously this inequality is equivalent to $\hat{x}(k) \leq EF^T\hat{x}(k+1)$. Noting that $\hat{x}(k), \hat{x}(k+1) \in \mathcal{M}_0$, $EF^T \in \{0,1\}^{m\times m}$ hold, if the i-th element $\hat{x}_i(k)$ of $\hat{x}(k)$ is equal to 1, then $\hat{x}_i(k) = 1 = (EF^T)_i\hat{x}(k+1)$, i.e., $\hat{x}_i(k) = \hat{x}_i(k)(EF^T)_i\hat{x}(k+1)$ holds, where $(EF^T)_i$ is the i-th row of EF^T. If $\hat{x}_i(k) = 0$ holds, then $\hat{x}_i(k) = 0 \leq (EF^T)_i\hat{x}(k+1)$, i.e, $\hat{x}_i(k) = \hat{x}_i(k)(EF^T)_i\hat{x}(k+1)$ holds. Thus if $\hat{x}(k) \leq EF^T\hat{x}(k+1)$ holds, then $\hat{x}(k) = \mathrm{diag}(\hat{x}_1(k), \ldots, \hat{x}_m(k))EF^T\hat{x}(k+1)$ holds. Furthermore, since $\hat{x}(k), \hat{x}(k+1) \in \mathcal{M}_0$, $\hat{x}(0) = \delta_0^M$, $\hat{x}(k) \leq EF^T\hat{x}(k+1)$ hold, $\hat{x}(k)$ is regarded as $\delta^M(k)$ (see Definition 10). So from Lemma 7, we obtain $-\hat{x}(k) + E\xi(k) = 0$, i.e., $-\hat{x}(k) + \tilde{E}\hat{u}(k) = -\xi_a(k)$ in (29). Hence we have (29).

This completes the proof. □

It is remarked here that (28) is a state-equation-based model with a smaller number of binary input variables, which can be directly used in the MLD model.

3.4 Procedure for deriving a state-equation-based model

The above discussions conclude the procedure for deriving a state-equation-based model as follows.

Procedure for deriving a state-equation based model (Summary):

Step 1: For an automaton satisfying Assumption 1, derive an implicit system (15).

Step 2: In terms of $\hat{V}$ of (21), and (23), (24), transform the implicit system (15) into a linear state equation (22). If $\hat{B}$ is full row rank, then (22) with $x(k) := \hat{x}(k)$ and $u(k) := \hat{u}(k)$ is the linear state equation to be found. Otherwise, go to Step 3.

Step 3: Represent the matrix $\hat{B}$ in (22) by (26), and obtain (28) from the state feedback binarization (32).

3.5 Example

This section shows by the simple example in Fig. 3, which satisfies Assumption 1, how the derivation procedure in Section 3.4 is applied.

- **Step 1: Derivation of the implicit system Σ_I of (15)**

First, from the input-output relation at each node, the implicit system $E\xi(k+1) = F\xi(k)$ in (15) (or (4)) is derived, where $n = 7$, $m = 3$, and

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

- **Step 2: Transformation of the implicit system Σ_I of (15) into the state equation (22)**

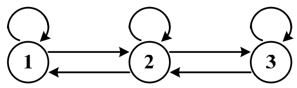


Fig. 3. Simple 3-node DF automaton

As P and $\tilde{E}$ satisfying $EP = [\ \mathbb{I}_m \quad \tilde{E}\ ]$, we consider

$$P = \begin{bmatrix} 1&0&0&0&0&0&0 \\ 0&0&0&1&0&0&0 \\ 0&0&0&0&1&0&0 \\ 0&1&0&0&0&0&0 \\ 0&0&0&0&0&1&0 \\ 0&0&0&0&0&0&1 \\ 0&0&1&0&0&0&0 \end{bmatrix}, \quad \tilde{E} = \begin{bmatrix} 1&0&0&0 \\ 0&1&1&0 \\ 0&0&0&1 \end{bmatrix}.$$

It is remarked here that P is not uniquely determined. From the above P and $\tilde{E}$, the matrix $\hat{V}$ of (21) is obtained as follows:

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

We also have

$$FP = \left[\begin{array}{ccc|cccc} 1&0&0&0&1&0&0 \\ 0&1&0&1&0&0&1 \\ 0&0&1&0&0&1&0 \end{array}\right] = \left[\ \tilde{F}_a \quad \tilde{F}_b\ \right].$$

Then applying $\hat{V}$ to (23) and (24) yields the state equation (22), where $\hat{x}(k) = [\ \delta_{11}(k) + \delta_{12}(k) \quad \delta_{21}(k) + \delta_{22}(k) + \delta_{23}(k) \quad \delta_{32}(k) + \delta_{33}(k)\ ]^T$, $\hat{u}(k) = [\ \delta_{12}(k) \quad \delta_{21}(k) \quad \delta_{23}(k)\ \delta_{32}(k)\ ]^T$, and

$$\hat{A} = \mathbb{I}_3, \quad \hat{B} = \begin{bmatrix} -1&1&0&0 \\ 1&-1&-1&1 \\ 0&0&1&-1 \end{bmatrix}.$$

Since $\hat{B}$ is not full row rank, go to Step 3.

• Step 3: Reduction of the input matrix $\hat{B}$ and state feedback binarization

From rank$\hat{B} = 2(\leq 4)$ we see that $\hat{u}(k)$ includes two redundant variables. So these redundant variables will be eliminated as follows. First, the matrix $\hat{B}$ is rewritten as (26); for example,

$$\hat{B} = \underbrace{\begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix}}_{P_B} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ -1 & -1 & 0 & 0 \end{bmatrix} \underbrace{\begin{bmatrix} -1 & 1 & 0 & 0 \\ 0 & 0 & 1 & -1 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}}_{T_B}.$$

Note also that P_B is not uniquely determined. From the matrix T_B, we have

$$\begin{bmatrix} \tilde{u}(k) \\ \tilde{u}_e(k) \end{bmatrix} = T_B \hat{u}(k) = \begin{bmatrix} -\delta_{12}(k) + \delta_{21}(k) \\ +\delta_{23}(k) - \delta_{32}(k) \\ \hline \delta_{12}(k) \\ \delta_{32}(k) \end{bmatrix}$$

where $\tilde{u}(k)$ is an independent variable, and $\tilde{u}_e(k)$ is a dependent variable. Thus $\tilde{u}_e(k)$ can be eliminated. However, we see that $\tilde{u}(k)$ is not a binary variable because its elements can take the value of -1, 0, 1. Hence a further transformation of $\tilde{u}(k)$ to a binary variable is required without the dimension of $\tilde{u}(k)$ increased.

So consider a state feedback binarization (32). Then from $\tilde{B} = [\ -1 \quad -1\ ]$, we obtain a state feedback binarization $\tilde{u}(k) = -\bar{A}_a \bar{x}(k) + u(k)$ as

$$\tilde{u}(k) = -\underbrace{\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}}_{\bar{A}_a} \underbrace{\begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 1 & 1 & 1 \end{bmatrix} x(k)}_{\bar{x}(k)} + u(k).$$

Note that $x(k)$ is defined as $\hat{x}(k)$. From Theorem 9 $u(k)$ is treated as a free binary variable. Thus we obtain the following state equation representation:

$$\begin{cases} x(k+1) = \begin{bmatrix} 0 & 0 & 0 \\ 1 & 1 & 1 \\ 0 & 0 & 0 \end{bmatrix} x(k) + \begin{bmatrix} 1 & 0 \\ -1 & -1 \\ 0 & 1 \end{bmatrix} u(k), \\ \begin{bmatrix} 0 & -1 & -1 \\ -1 & 0 & -1 \\ -1 & -1 & 0 \\ 1 & 1 & 1 \\ -1 & -1 & -1 \end{bmatrix} x(k) + \begin{bmatrix} 0 & 1 \\ 0 & 0 \\ 1 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix} u(k) \le \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \\ -1 \end{bmatrix}, \\ x(k) \in \mathbf{R}^3, \quad u(k) \in \{0,1\}^2, \\ x(0) = \delta_0^M \in \mathcal{M}_0, \quad u(0) = u_0 \in \{0,1\}^2. \end{cases}$$

The obtained state equation can be directly expressed in the form of the MLD model.

Finally, we discuss the computation cost of the proposed procedure. In the proposed procedure, permutation matrices P, P_B are not uniquely determined. However, from Lemma 6 (i), the dimension $\hat{\alpha}$ of the binary input variables does not depend on the selection of P, P_B. So we may consider to find one P, P_B. Therefore, since the procedure includes no calculation of iterations, the computation cost is small. For example, consider 100 randomly directed graphs with 20 nodes. Then the worst computation time of the proposed procedure is 0.0396 (sec), where we used the MATLAB.

4 Further discussions on state-equation-based modeling

This section provides further results on our modeling method. First, we clarify the meaning of the binary input variables in the state-equation-based model. Next, the upper bound of the dimension $\hat{\alpha}$ of the input vector in (28) is directly derived from the numbers of modes and arcs. From the obtained result, it will be shown that the input dimension in the state-equation-based model is necessarily smaller than or equal to that in the inequality-based model and the implicit system.

4.1 Relation among state/input variables in deterministic finite automata representations

In this subsection, after showing the relation on variables between the inequality-based model and the state-equation-based model (or the implicit system), we discuss what are binary input variables appeared in the state-equation-based model.

The relation between the state-equation-based model (28) and the implicit system (15) has been already shown in the previous subsections. Thus we focus on the inequality-based model such as (3) and (2), which is formally defined as follows.

Definition 10 *The inequality-based model expressing a given automaton $\mathcal{A}$ with $\delta_0^M \in \mathcal{M}_0$ is defined as*

$$\begin{cases} \delta^M(k+1) = u^M(k), \quad \delta^M(k) \leq \Phi\delta^M(k+1), \\ u^M(k) \in \mathcal{M}_0, \quad \delta^M(0) = \delta_0^M \end{cases} \tag{36}$$

where $\Phi \in \{0,1\}^{m\times m}$ is the adjacency matrix of the automaton $\mathcal{A}$.

Then since $x(k) = \delta^M(k)$ holds from Lemma 8, we see that the state $x(k)$ of the state-equation-based model (28) equivalently corresponds to the mode variables $\delta^M(k)$. Also in the inequality-based model, $\delta^M(k)$ is the state variable, and $u^M(k) = \delta^M(k+1)$ is the input variable. Furthermore, by $e_m^T\eta = 1$ in $\mathcal{M}_0$, $m-1$ elements of $u^M(k)$ are regarded as free binary variables. However, since $F\xi(k) = u^M(k)$ follows from the proof of (iii) in Lemma 3, we see that the degree of freedom in $u^M(k)$ depends on the matrix F. Furthermore, since $F\xi(k) = \hat{A}x(k) + \hat{B}\hat{u}(k) = Ax(k) + Bu(k)$ holds by the proofs of Theorems 5 and 9, we have $u^M(k) = A\delta^M(k) + Bu(k)$. Thus if $\dim u(k) < m-1$ holds, then the inequality-based model includes redundant binary variables.

Therefore, from the above discussion, we see that the state-equation-based model (28) is obtained from the following system using (15) and (36)

$$\begin{cases} \delta^M(k+1) = F\xi(k), \quad \delta^M(k) \leq \Phi\delta^M(k+1), \\ \xi(k) \in \{0,1\}^n, \quad \xi(0) \in \Xi_0(\delta_0^M). \end{cases} \tag{37}$$

Then the inequalities $\delta^M(k) \leq \Phi\delta^M(k+1)$ in (37) can be regarded as the constraints for guaranteeing the binary property of $\delta^M(k+1)$ in (28).

Finally, we discuss the meaning of the binary input variables of the state-equation-based model. From (34) and $x(k+1) = \delta^M(k+1)$, $u(k)$ is given as

$$u(k) = [\ \mathbb{I}_{\hat{\alpha}} \ \ 0_{\hat{\alpha}\times(m-\hat{\alpha})}\ ]P_B^{-1}\delta^M(k+1).$$

Noting that P_B is the permutation matrix obtained by (26), we see that the input variable vector $u(k)$ consists of $\hat{\alpha}$ elements in $\delta^M(k+1)$, that is, $u(k)$ at least implies the next mode itself which the system can be transited to, although the meaning in the graph structure of $\hat{\alpha}$ cannot be exactly captured by the above analysis.

4.2 Upper bound of input dimension

The dimension $\hat{\alpha}$ of the input vector in (28) corresponds to the number of the input binary variables in the MLD model. So it is important to determine the value of $\hat{\alpha}$ in (28) directly from the graph structure of a given automaton

$\mathcal{A}$, because the worst computation time of the MIP problem increases exponentially with the dimension of the binary variables $(= (m_{1d} + m_3)N)$. However this issue will be very hard in general. Thus we will find an upper bound of $\hat{\alpha}$, derived directly from the numbers m, n of the nodes and the arcs. Then Lemma 6 in Section 3.3 enables us to derive the upper bound of $\hat{\alpha}$ as follows.

Theorem 11 *For the input dimension $\hat{\alpha}$ of the state-equation-based model (28), the following relation holds:*

$$\hat{\alpha} \leq \min\{n - m,\ m - 1\} \tag{38}$$

where m, n are the number of the nodes and the arcs in the automaton $\mathcal{A}$.

PROOF. Suppose that $m - 1 \leq n - m$. From Lemma 6(ii), $\hat{B} \in \{-1, 0, +1\}^{m\times(n-m)}$ is the incidence matrix of some directed graph. Then $\hat{\alpha}(= \text{rank}\hat{B})$ is given by $m - c$, where $c(\geq 1)$ is the number of connected components of some directed graph [7]. Therefore, (38) holds because the minimum value of c is equal to 1. On the other hand, if $n - m < m - 1$ holds, then $\hat{\alpha}(= \text{rank}\hat{B}) \leq n - m$ holds directly. □

Note that the free input dimension of the inequality-based model can be given as $m-1$, and the free input dimension of the implicit system (15) as $n - m$ because m equations, i.e., $E\xi(k + 1) = F\xi(k)$, hold. Therefore, from Theorem 11, we see that the number of the free binary variables of the state-equation-based model is necessarily smaller than or equal to that of the inequality-based model and the implicit system. In addition, if a given directed graph has sparse structure, then $n - m$ becomes small. So the effectiveness of the state-equation-based model is big for sparse directed graphs.

We show two examples. First, consider the DF automaton in Fig. 4 (a). This automaton has $4(= m)$ nodes and $7(= n)$ arcs. Then from Theorem 11, $\hat{\alpha} \leq \min\{3,\ 3\} = 3$ holds. In particular, applying the proposed procedure to this automaton, we obtain $x(k + 1) = \hat{A}x(k) + \hat{B}\hat{u}(k)$ of (22) as

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

So we see that $\hat{\alpha} = \text{rank}\hat{B} = 2 < 3$ holds.

Next, consider the DF automaton in Fig. 4 (b). This automaton has $7(= m)$ nodes and $10(= n)$ arcs. Then $\hat{\alpha} \leq \min\{3,\ 6\} = 3$ holds. Applying the proposed procedure yields $\hat{A}$ and $\hat{B}$ of (22) as

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

Then we have $\hat{\alpha} = 3$.

From these examples, we see that in Theorem 11, both cases of $\hat{\alpha} < \min\{n-m,\ m-1\}$ and $\hat{\alpha} = \min\{n-m,\ m-1\}$ will be able to occur.

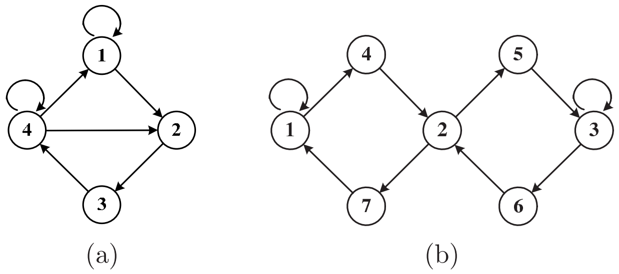


Fig. 4. Examples of DF automata

5 Conclusion

In this paper, we have proposed a state-equation-based modeling method of a DF automaton as a method for reducing the computation time spent to solve the MIP problem, i.e., the MPC problem of a class of hybrid systems. Furthermore, we have discussed the meaning of binary input variables in the proposed model, and derived the upper bound of the binary input dimension.

To clarify the relation between the binary input dimension and the graph structure is still one of the most interesting future topics. In addition, it will be one of significant topics to exploit our model in solving various kinds of control problems. Based on our model, the stabilization problem of a hybrid system /DF automaton has been addressed in [17].

Acknowledgment

The authors would like to thank Prof. Tomomi Matsui, Chuo University, and Prof. Ryuhei Miyashiro, Tokyo University of Agriculture and Technology for fruitful discussion.

References

[1] J. D. Aplevich, *Implicit Linear Systems. Lecture Notes in Control and Information Sciences 152*, Springer, 1991.

[2] A. G. Beccuti, T. Geyer, and M. Morari, "A hybrid system approach to power systems voltage control," in *Proc. 44th IEEE Conf. on Decision and Contr. and 2005 European Contr. Conf.*, pp. 6774–6779, 2005.

[3] A. Bemporad and M. Morari, "Control of systems integrating logic, dynamics, and constraints," *Automatica*, vol. 35, pp. 407–427, 1999.

[4] A. Bemporad, P. Borodani, and M. Morari, "Hybrid control of an automotive robotized gearbox for reduction of consumptions and emissions," in *Proc. 6th Int'l Workshop on Hybrid Systems: Computation and Control*, O. Maler, A. Pnueli Eds., *Lecture Notes in Computer Science 2623*, pp. 81–96, Springer, 2003.

[5] A. Bemporad and N. Giorgetti, "A logic-based hybrid solver for optimal control of hybrid systems," in *Proc. 42nd IEEE Conf. on Decision and Contr.*, pp. 640–645, 2003.

[6] A. Bemporad and N. Giorgetti, "Logic-based solution methods for optimal control of hybrid systems," *IEEE Trans. on Automat. Contr.*, vol. 51, no. 6, pp. 963–976, 2006.

[7] B. Bollobás, *Modern Graph Theory*. Springer, 1998.

[8] F. Borrelli, A. Bemporad, M. Fodor, and D. Hrovat, "A hybrid approach to traction control," in *Proc. 4th Int'l Workshop on Hybrid Systems: Computation and Control*, M.D. Di Benedetto, A. Sangiovaani-Vincentelli Eds., *Lecture Notes in Computer Science 2034*, pp. 162–174, Springer, 2001.

[9] F. Borrelli, *Constrained Optimal Control of Linear and Hybrid Systems*, *Lecture Notes in Control and Information Sciences 290*, Springer, 2003.

[10] G. Ferrari-Trecate, E. Gallestey, P. Letizia, M. Spedicate, M. Morari, and M. Antoine, "Modeling and control of co-generation power plants: A hybrid system approach," in *Proc. 5th Int'l Workshop on Hybrid Systems: Computation and Control*, C. J. Tomlim, M. R. Greenstreet Eds., *Lecture Notes in Computer Science 2289*, pp. 209–224, Springer, 2002.

[11] F. R. Gantmacher, *The Theory of Matrices*. vol. 1, 2nd ed., Chelsea Pub. Co., 1990.

[12] T. Geyer, G. Papafotiou, and M. Morari, "On the optimal control of switch-mode DC-DC converters," in *Proc. 7th Int'l Workshop on Hybrid Systems: Computation and Control*, R. Alur, G. J. Pappas Eds., *Lecture Notes in Computer Science 2993*, pp. 342–356, Springer, 2004.

[13] P. Grieder, M. Kvasnica, M. Baotić, and M. Morari, "Stabilizing low complexity feedback control of constrained piecewise affine systems," *Automatica*, vol. 41, no. 10, pp. 1683–1694, 2005.

[14] J. Imura, "Optimal control of sampled-data piecewise affine systems," *Automatica*, vol. 40, no. 4, pp. 661–669, 2004.

[15] M. Ishikawa, T. Seki, J. Imura, and S. Hara, "An efficient algorithm for optimal control of hybrid dynamical systems utilizing mode transition rule," in *Proc. 43rd IEEE Conf. on Decision and Contr.*, pp. 1866–1871, 2004.

[16] D. Jungnickel, *Graphs, Networks and Algorithms.* Springer, 1998.

[17] K. Kobayashi, J. Imura, and K. Hiraishi, "Stabilization of finite automata with application to hybrid systems control," *Discrete Event Dynamic Systems: Theory and Applications*, vol. 21, no. 4, pp. 519-545, 2011.

[18] M. Kuijper, *First-order Representations of Linear Systems.* Birkhäuser, 1994.

[19] M. Kvasnica, P. Grieder, and M. Baotić, "Multi-parametric toolbox (MPT)," http://control.ee.ethz.ch/˜mpt/, 2004.

[20] J. Rosenthal, J. M. Schumacher, and E. V. York, "On behaviors and convolutional codes," *IEEE Trans. Inform. Theory*, vol. 42, no. 6, pp. 1881–1891, 1996.

[21] J. Rosenthal and E. V. York, "BCH convolutional codes," *IEEE Trans. Inform. Theory*, vol. 45, no. 6, pp. 1833–1844, 1999.

[22] J. C. Willems, "Paradigms and puzzles in the theory of dynamical systems," *IEEE Trans. Automat. Contr.*, vol. 36, no. 3, pp. 259–294, 1991.

[23] H. P. Williams, *Model Building in Mathematical Programming*, 4th Eds., John Willey & Sons, Ltd, 1999.

[24] http://www.ilog.com/products/cplex/