Free-Variable Analysis of Finite Automata Representations for Hybrid Systems Control

Koichi Kobayashi and Jun-ichi Imura

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 of hybrid systems such as mixed logical dynamical systems. To overcome this issue, the authors have proposed a new method to represent a deterministic finite automaton as a linear state equation with a relatively smaller number of (free) binary input variables, which thus makes the number of binary variables in the resultant MIP problem smaller. This paper continues upon the above approach, and presents theoretical aspects on input binary variables in the linear state equation model such as the upper bound of the number of the binary input variables.**

I. Introduction

The model predictive control (MPC) method based on a mixed logical dynamical (MLD) model [2] will be one of the most useful approaches to control of hybrid systems, and many works on its applications have been reported so far. 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.

To overcome this issue, it is one of effective approaches to express a discrete dynamical system such as a deterministic finite automaton, which is a part of a hybrid system to be controlled, in terms of a relatively smaller number of free binary variables, since the computation time in solving the MIP problem exponentially grows with the number of free binary variables in general. 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 using an adjacency matrix (called here an inequality-based model) is well known as a method for expressing deterministic finite automata [2], [11], but it will not be desirable for the branch and bound method [5].

Thus the authors have proposed in [6] a new modeling method of representing a deterministic finite automaton (for simplicity, finite automaton, hereafter) as *a linear state equation* (referred to as the state-equation-based model) with a relatively small number of free binary variables (called here binary input variables), based on the implicit-system-based model representation [1], [8]. It has also been shown that our method is very effective, by numerical examples of the MPC problem of switched systems. Furthermore, for the proposed model, we have discussed the minimality of the dimension of binary input variables under a class of equivalence transformations [7].

At this step, we will point out the following natural and significant questions on binary input variables of the state-equation-based model: (i) what do binary input variables obtained in the state-equation-based model mean?, (ii) can the number of binary input variables be derived directly from a graph structure of a given finite automaton?, and In this paper, we give solutions to these questions. First, using the relation on variables between the inequality-based model and the state-equation-based model, we present a characterization of binary input variables, which is a solution to (i). Next, as a partial solution to (ii), this paper derives an upper bound of the input dimension in the state-equation-based model in terms of the numbers of nodes/arcs, and shows that the input dimension in the state-equation-based model is necessarily smaller than or equal to that in the inequality-based model.

Notation: Let $\mathbf{R}$ express the set of real numbers. Denote by $\{0,1\}^{m\times n}$ the set of $m \times n$ matrices consisting of elements 0 and 1. Similarly for $\{-1,0,1\}^{m\times n}$. 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 one, respectively. For simplicity of notation, the symbols 0 and I are often used instead of $0_{m\times n}$ and I_n, respectively.

II. Outline of The State-Equation-Based Modeling Method

Consider an example of a finite automaton in Fig. 1, where the number symbol in each node denotes the mode (the discrete state) of the system, to explain the standard method [2] and our proposed method [6], [7].

First, we briefly explain the standard method based on binary inequalities. Suppose that a binary variable $\delta_i^M(k)$ is assigned to each node (i.e., each mode), and that $\delta_i^M(k)=1$ and $\delta_j^M(k)=0$ for all $j\neq i$ hold when the mode at time k is i, where the equality constraint $\sum_{i=1}^4 \delta_i^M(k)=1$ is required. Then the discrete dynamics in Fig. 1 are expressed by the following binary linear inequalities:

$$\left\{\begin{array}{l} \delta_1^M(k)-\delta_1^M(k+1)-\delta_2^M(k+1)\leq 0,\\ \delta_2^M(k)-\delta_4^M(k+1)\leq 0,\\ \delta_3^M(k)-\delta_1^M(k+1)\leq 0,\\ \delta_4^M(k)-\delta_3^M(k+1)-\delta_4^M(k+1)\leq 0,\\ \sum_{i=1}^4 \delta_i^M(k)\leq 1,\quad -\sum_{i=1}^4 \delta_i^M(k)\leq -1. \end{array}\right. \qquad (1)$$

So defining $\delta^M(k) := \left[\begin{array}{cccc}\delta_1^M(k) & \delta_2^M(k) & \delta_3^M(k) & \delta_4^M(k)\end{array}\right]^T$ and $u(k) := \delta^M(k+1)$, the binary linear inequalities in (1) are expressed in terms of the combination of the state

K. Kobayashi is with the School of Information Science, Japan Advanced Institute of Science and Technology, Ishikawa, Japan k-kobaya@jaist.ac.jp

J. Imura is with the Graduate School of Information Science and Engineering, Tokyo Institute of Technology, Tokyo, Japan imura@mei.titech.ac.jp

ISBN 978-963-311-370-7

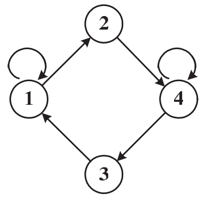

Fig. 1. Example of deterministic finite automaton

equation $\delta^M(k+1) = u(k)$ and the binary linear inequality (1). This model is called here the inequality-based model. Furthermore, assigning the continuous dynamics to each node in the finite automaton of Fig. 1 yields a switched system. Then this system is expressed as the MLD model [2]:

$$\bar{x}(k+1) = \bar{A}\bar{x}(k) + \bar{B}\bar{v}(k), \tag{2}$$

$$\bar{C}\bar{x}(k) + \bar{D}\bar{v}(k) \leq \bar{G} \tag{3}$$

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 $m_{1d} = 4$, $m_3 = 0$ for free binary variables.

Next, let us explain the modeling method proposed in [6], [7] using the above example. See Appendix I for the general case. In our method, a binary variable is assigned to an arc (directed edge), not to a node. So a binary variable δ_{ij} denotes the arc from the node i to the node j, which thus can express the input-output relation at each node (note that the relation between δ_{ij} and a pair (δ_i, δ_j) is given by $\delta_{ij}(k) = \delta_i(k)\delta_j(k+1)$). For example, the input-output relation at node 1 is expressed by the equation $\delta_{11}(k+1)+\delta_{12}(k+1) = \delta_{11}(k)+\delta_{31}(k)$. Thus expressing the input-output relation at every node in a similar way, we express the finite automaton of Fig. 1 as the following discrete-time implicit system model with an equality constraint on the initial state (called here an implicit-system-based model):

$$E\xi(k+1) = F\xi(k), \quad \xi(k) \in \{0,1\}^6, \quad e_6^T\xi(0) = 1 \tag{4}$$

where $\xi = [\ \delta_{11}\ \ \delta_{12}\ \ \delta_{24}\ \ \delta_{43}\ \ \delta_{44}\ \ \delta_{31}\ ]^T$. The details of E and F are omitted. Furthermore, by some transformation of coordinate, the implicit system model (4) can be equivalently transformed into the following state equation with inequality constraints:

$$\begin{cases} x(k+1) = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 \end{bmatrix} x(k) + \begin{bmatrix} 1 & 0 \\ -1 & 0 \\ 0 & -1 \\ 0 & 1 \end{bmatrix} u(k), \\ -x(k) + \begin{bmatrix} 1 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 1 \end{bmatrix} u(k) \leq 0, \\ x(k) \in \mathbf{R}^4, \quad u(k) \in \{0,1\}^2, \\ x(0) \in \mathcal{X}_0 \end{cases} \tag{5}$$

where $\mathcal{X}_0 := \left\{\ \zeta \in \{0,1\}^4 \mid e_4^T\zeta = 1\ \right\}$. This is called here the state-equation-based model. This implies that the solution $\xi(k)$ of the implicit system at time k is uniquely determined by the state $x(k)$ and the input (free variable) $u(k) \in \{0,1\}^2$. Note also that the initial state $x(0)$ has to satisfy the binary property and the equality constraint $e_4^Tx(0) = 1$. In this case, we have $m_{1d} = 2$, $m_3 = 0$ for free binary variables in the MLD model (2), (3). Thus by using the state-equation-based modeling method, the dimension (m_{1d}) of binary variables in the MLD model can be decreased from 4 to 2. Of course, from Fig. 1 and (1), m_{1d} will be also decreased. However, our method enables us to decrease the dimension in a systematic way described in Appendix I.

The general form of (5) is given as

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

where the input dimension $\hat{\alpha}$ is determined by the derivation procedure in Step 3 of Appendix I, and $\mathcal{X}_0$ is defined newly by replacing the dimension of the space in $\mathcal{X}_0$ of (5) by m. Thus this model representation is included in the class of the MLD systems. The MPC problem of hybrid systems is in general reduced, through the MLD model, to a mixed integer programming (MIP) problem; thus the number of the free binary variables (i.e., binary decision variables) in the MIP problem is given by $(m_{1d}+m_3)N$ (N: the prediction horizon length). In the above example, we see that the state-equation-based model can give a smaller number of free binary variables in the MIP problem than the inequality-based model does.

We briefly show numerical results on the computation time in solving the MIP problem. As an example, we consider a simple motorbike model with five gears as an example of hybrid systems. This model is based on that in [3]. 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. In addition, $I \in \{1,2,3,4,5\}$ denotes the I-th gear. Then the continuous dynamics are described by the following discrete-time state equation

$$x_c(k+1) = A_c^{I(k)}x_c(k) + B_cu_c(k) \tag{7}$$

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

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

$\alpha_1 = 0.1$, $\alpha_2 = 0.08$, $\alpha_3 = 0.06$, $\alpha_4 = 0.04$, $\alpha_5 = 0.02$, 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 finite automaton in Fig. 2. For simplicity of discussion, we assume that gear shifts depend on only automaton, not continuous state variables. The system consisting of (7) and the gear shift logic in Fig. 2 is a discrete-time switched linear system, and can be expressed as the MLD model.

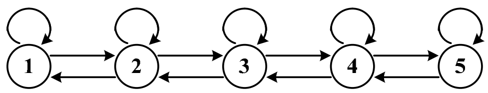

Fig. 2. Gear shift logic

Numerical experiments of the finite-time optimal control problem (i.e., the MIP problem) are shown below. The details of this problem are omitted due to the limited space. In this experiment, N is given as $N = 20$, and as a result, the numbers of binary variables are given as 100 in the inequality-based model and 80 in the state-equation-based model, respectively. Then the computation times for solving the MIP problem are given as

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

where we used ILOG CPLEX 11.0 [12] as an MIP 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. Although the state-equation-based modeling seems to be very powerful for MPC, it is hard to theoretically discuss the effectiveness of the state-equation-based modeling. However, from this experiment, it is clear that the state-equation-based model is useful as one of models expressing finite automata such as Fig. 2. Therefore, it is important to solve several theoretical issues on such free binary variables, e.g., can we specify (the upper bound of) the number of free binary variables directly from the graph structure of a finite automaton. This paper will address them.

III. Finite Automata and Implicit System Models

This section describes assumptions on a finite automaton to be studied here and an implicit-system-based model expressing the finite automaton.

We consider throughout the paper a deterministic finite automaton, i.e., the case in which one of nodes in the finite automaton, which expresses the mode (the discrete state) of the corresponding hybrid system, becomes active at each discrete time according to the discrete dynamics specified by the finite automaton.

The following assumption is made for this finite automaton.

Assumption 1: A finite automaton is given as a connected directed graph, 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, where the input-(output-)arc at a node denotes the arc whose arrowhead(rear) is connected to the node.

By Assumption 1, four cases as shown in Fig. 3 can be excluded. In the cases of Fig. 3 (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. 3 (c), (d) the arc whose end point is going into node 1 is regarded

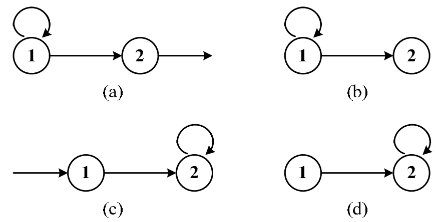

Fig. 3. Exceptional cases of finite automata

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 systems. Hence these four cases are excluded here, and the finite automaton satisfying Assumption 1 is denoted by $\mathcal{A}$ hereafter.

For the finite automaton $\mathcal{A}$, the following implicit system is given. Let m and n denote the number of the nodes and that of the arcs in the finite automaton, respectively. Suppose that each arc is labeled with the index j, $j = 1, 2, \ldots, n$, whose index set is denoted by $\mathcal{I}_a$, 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

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

where $\xi(k) := [\ \delta_1(k) \quad \delta_2(k) \quad \cdots \quad \delta_n(k)\ ]^T \in \{0,1\}^n$ 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, we can easily prove by construction that if $\xi(0)$ satisfies $e_n^T\xi(0) = 1$, then $e_n^T\xi(k) = 1$ holds for all k, and that the implicit system (8) can express every mode behavior according to the discrete dynamics of a finite automaton $\mathcal{A}$. Then this implicit system has the following relation, which will be used in Lemma 4 of section V.

Lemma 1: For the implicit system (8), the following relation holds:

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

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.

Proof: For an arc $\bar{j} \in \mathcal{I}_I(\bar{i})$ at node $\bar{i}$, we have

$$\left\{ \begin{array}{l} F_{\bar{i},\bar{j}} = 1, \\ F_{i,\bar{j}} = 0, \quad i \neq \bar{i} \end{array} \right. \qquad (10)$$

where $F_{i,j}$ denotes the (i, j)-th element of F. This condition holds for each node $\bar{i} \in \{1, 2, \ldots, m\}$. In a similar way, for an arc $\bar{j} \in \mathcal{I}_O(\bar{i})$ at node $\bar{i}$ we have

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

where $E_{i,j}$ denotes the (i, j)-th element of E. Then the relation (9) follows from (10), (11) and $m \leq n$ (by Assumption 1). This completes the proof. ■

IV. Relation among State/Input Variables in Finite Automata Models

In this section, we give a solution to the first question, i.e., "(i) what do free binary variables (i.e., binary input variables) obtained in the state-equation-based model mean?".

First, 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 $\delta^M := [\ \delta_1^M \ \ \delta_2^M \ \ \cdots \ \ \delta_m^M \ ]^T$. Then the following lemma provides the relation between $\xi(k)$ and $\delta^M(k)$.

Lemma 2: 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, (i) follows from the definition of $\delta_i^M(k)$ straightforwardly.

Next, (ii) is proven. Define the symbol $\eta(k)$ as $\eta(k) := E\xi(k)$. By 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.,

$$\eta_i(k) = \sum_{j \in \mathcal{I}_O(i)} \delta_j(k).$$

Denote by $l_i(j)$ the node to which the arc $j \in \mathcal{I}_O(i)$ is connected. Then noting that $\delta_j(k) = \delta_i^M(k)\delta_{l_i(j)}^M(k+1)$, $j \in \mathcal{I}_O(i)$, we have

$$\eta_i(k) = \delta_i^M(k) \sum_{j \in \mathcal{I}_O(i)} \delta_{l_i(j)}^M(k+1). \quad (12)$$

If $\delta_i^M(k) = 1$ holds, then it follows from the definition of $l_i(j)$ that $\sum_{j \in \mathcal{I}_O(i)} \delta_{l_i(j)}^M(k+1) = 1$ holds, 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 above two relations mean $\eta_i(k) = \delta_i^M(k)$. Hence relation (ii) is proven. Finally, since $\delta^M(k+1) = E\xi(k+1) = F\xi(k)$ holds from (ii), (iii) is proven immediately. This completes the proof. ■

Lemma 2 (ii) shows the relation between the arc variables $\xi(k)$ and the node variables $\delta^M(k)$, which provides a key in discussing the meaning of binary free variables in the state-equation-based model through the relation among three kinds of finite automaton models.

Furthermore, from Lemma 2 (ii), (iii), we see that the initial state of this system should be carefully defined. In the optimal control problem of hybrid systems, in general, an initial mode is given. However, by Lemma 2 (iii), the value of the next mode, $\delta^M(1)$, is uniquely determined if $\xi(0)$ is given in advance. So we naturally suppose that the initial mode $\delta_0^M \in \mathcal{M}_0$ is given as an initial state of this system, where

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

Then applying the set $\Xi_0(\delta_0^M)$ defined as, based on Lemma 2(ii)

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

to (8), we hereafter consider the following implicit system:

$$\Sigma_I : \begin{cases} E\xi(k+1) = F\xi(k), \\ \xi(k) \in \{0,1\}^n, \quad \xi(0) \in \Xi_0(\delta_0^M). \end{cases} \quad (13)$$

for a given finite automaton $\mathcal{A}$ with $\delta_0^M \in \mathcal{M}_0$. Appendix I shows the procedure of deriving the state-equation-based model from the above implicit-system-based model.

On the other hand, the inequality-based model (the binary-inequality based representation using an adjacency matrix) is defined as follows.

Definition 1: The inequality-based model expressing a given finite automaton $\mathcal{A}$ with $\delta_0^M \in \mathcal{M}_0$ is defined as

$$\begin{cases} \delta^M(k+1) = u^M(k), \quad \delta^M(k) \leq \Phi\delta^M(k+1), \\ u^M(k) \in \mathcal{M}_0, \quad \delta^M(0) = \delta_0^M \end{cases} \quad (14)$$

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

Then we have the following result.

Lemma 3: For the state-equation-based model and the inequality-based model expressing a given finite automaton $\mathcal{A}$ with $\delta_0^M \in \mathcal{M}_0$,

$$x(k) = \delta^M(k) \quad (15)$$

holds.

Proof: From the relation $[\ x^T(k) \ \ \hat{u}^T(k)\ ]^T := \hat{V}\xi(k)$ and $EP = [\ I_m \ \ \tilde{E}\ ]$ in Step 2 of Appendix I, we have $E\xi(k) = x(k)$. Thus Lemma 2 (ii) completes the proof. ■

This lemma implies that the state of the state-equation-based model equivalently corresponds to the mode variables.

Now we are in a position to discuss the meaning of the binary input variables of the state-equation-based model.

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 2, we see that the degree of freedom in $u^M(k)$ depends on the matrix F. Furthermore, since $F\xi(k) = Ax(k) + Bu(k)$ by Steps 2 and 3 of Appendix I, 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 (6) is obtained from the following system using (13) and (14)

$$\begin{cases} \delta^M(k+1) = F\xi(k), \quad \delta^M(k) \leq \Phi\delta^M(k+1), \\ \xi(k) \in \{0,1\}^n, \quad \xi(0) \in \Xi_0(\delta_0^M). \end{cases} \quad (16)$$

Then the inequalities $\delta^M(k) \leq \Phi\delta^M(k+1)$ in (16) can be regarded as the constraints for guaranteeing the binary property of $\delta^M(k+1)$ in (6). In addition, by some simple calculation of (19)-(22) in Appendix I (after (20) and (21) are substituted into (19), (22) and $x(k+1) = \delta^M(k+1)$ are used) $u(k)$ is given as

$$u(k) = [\ I_{\hat{\alpha}} \quad 0_{\hat{\alpha}\times(m-\hat{\alpha})}\ ]P_B^{-1}\delta^M(k+1). \tag{17}$$

Noting that P_B is the permutation matrix obtained by (20) in Appendix I, 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.

V. Upper Bound of Input Dimension

This section discusses question (ii) in section I, i.e., "(ii) can the number of binary input variables be derived directly from a graph structure of a given finite automaton?". As a partial solution to this question, we will show that the upper bound of $\hat{\alpha}$ is derived directly from the numbers m, n of the nodes and the arcs.

First, we discuss what property the matrix $\hat{B}$ in (19) in Appendix I has.

Lemma 4: The matrix $\hat{B} = -\tilde{F}_1\tilde{E} + \tilde{F}_2$ in (19) is the incidence matrix of some directed graph.

Proof: 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 (ii) of Lemma 1 and the permutation matrix P (see Appendix I), each column vector of $\tilde{F}_1$, $\tilde{F}_2$ and $\tilde{E}$ consists of a '1' and $m-1$ zeros. So each column vector of $\tilde{F}_1\tilde{E}$ also consists of a '1' and $m-1$ zeros. Therefore, each column vector of $\hat{B} = -\tilde{F}_1\tilde{E} + \tilde{F}_2 \in \{-1, 0, +1\}^{m\times(n-m)}$ is a zero vector, or is composed of a '+1', a '−1' and $m-2$ zeros. This implies that $\hat{B} = -\tilde{F}_1\tilde{E} + \tilde{F}_2$ is the incidence matrix of some directed graph. ■

Lemma 4 enables us to derive the upper bound of $\hat{\alpha}$ as follows.

Theorem 1: For the input dimension $\hat{\alpha}$ of the state-equation-based model (6), the following relation holds:

$$\hat{\alpha} \leq \min\{n-m,\ m-1\} \tag{18}$$

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

Proof: Suppose that $m-1 \leq n-m$. From Lemma 4, $\hat{B} \in \{-1, 0, +1\}^{m\times(n-m)}$ is the incident matrix of some directed graph. Then $\hat{\alpha}(=\ \mathrm{rank}\hat{B})$ is given by $m-c$, where $c(\geq 1)$ is the number of connected components of some directed graph [4]. Therefore, (18) holds because the minimum value of c is equal to 1. On the other hand, if $n-m < m-1$ holds, then $\hat{\alpha}(=\mathrm{rank}\hat{B}) \leq n-m$ holds directly. ■

Note that the input dimension of the inequality-based model (14) can be given as $m-1$, and the input dimension of the implicit-system-based model (13) as $n-m$ because m equations, i.e., $E\xi(k+1) = F\xi(k)$, hold. Therefore, from Theorem 1, 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-based model.

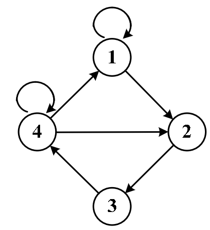

Fig. 4. 4-node finite automaton

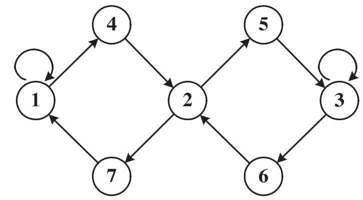

Fig. 5. 7-node finite automaton

Example 1: Consider the finite automaton in Fig. 4. This finite automaton has $4(=m)$ nodes and $7(=n)$ arcs. Then from Theorem 1, $\hat{\alpha} \leq \min\{3,\ 3\} = 3$ holds. In particular, applying the proposed procedure in Appendix I to this finite automaton, we obtain $x(k+1) = \hat{A}x(k) + \hat{B}\hat{u}(k)$ of (19) 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 $\hat{\alpha} = \mathrm{rank}\hat{B} = 2 < 3$ holds.

On the other hand, consider the finite automaton in Fig. 5. This finite 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 (19) as

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

So we see that $\hat{\alpha} = 3$ holds.

From these examples, we see that in Theorem 1, depending on a given finite automaton, $\hat{\alpha} < \min\{n-m,\ m-1\}$ or $\hat{\alpha} = \min\{n-m,\ m-1\}$ holds. ■

VI. Conclusion

In this paper, we have presented three theoretical aspects on the state-equation-based modeling of deterministic finite automata for hybrid systems control, which has been proposed in our previous papers; the meaning of binary input variables and the upper bounds of the number of binary input variables. These results will enhance the further usefulness of the state-equation-based modeling, and be helpful for developing control theory of finite automata in hybrid systems (e.g., see [9], [10]) based on our framework.

APPENDIX I
DERIVATION PROCEDURE OF STATE-EQUATION-BASED MODEL EXPRESSING DETERMINISTIC FINITE AUTOMATA

The procedure deriving a state-equation-based model from a given finite automaton satisfying Assumption 1 is as follows. This is a more sophisticated version of our approach derived in [6], [7].

Procedure of deriving a state-equation-based model:
Step 1: For a given deterministic finite automata $\mathcal{A}$ with m nodes and n arcs, let $\mathcal{I}_a = \{i_1,\ i_2,\ \cdots,\ i_n\}$ 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_{i_k} to the arc i_k. Furthermore, set $\xi(k) := [\ \delta_{i_1}(k)\ \ \delta_{i_2}(k)\ \ \cdots\ \ \delta_{i_n}(k)\ ]^T \in \{0,1\}^n$. Then the input-output relation of $\delta_{i_k}(k)$ 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 $E, F \in \{0,1\}^{m\times n}$,

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

and $\delta_0^M \in \{0,1\}^m$ denotes the 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

$$\begin{cases} x(k+1) = \hat{A}x(k) + \hat{B}\hat{u}(k), \\ -x(k) + \tilde{E}\hat{u}(k) \le 0, \\ x(k) \in \mathbf{R}^m, \quad \hat{u}(k) \in \{0,1\}^{n-m}, \\ x(0) = x_0 \in \mathcal{X}_0 := \{\ \zeta \in \{0,1\}^m \mid e_m^T\zeta = 1\ \}. \end{cases} \tag{19}$$

If $\hat{B}$ is full row rank, then (19) 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

$$\hat{B} = P_B \begin{bmatrix} I_{\hat{\alpha}} & 0 \\ \tilde{B} & 0 \end{bmatrix} T_B \tag{20}$$

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

$$\begin{bmatrix} \tilde{u}^T(k) & \tilde{u}_e^T(k) \end{bmatrix}^T := T_B\hat{u}(k) \tag{21}$$

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

$$\tilde{u}(k) = \hat{A}_u x(k) + u(k) \tag{22}$$

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 (19) yields

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

where

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

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

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. By substituting (22) into (19) and replacing the inequality of (19) to the inequality using the adjacency matrix Φ, we obtain the state-equation-based model. The input transformation of (22) in Step 3 guarantees the binary property of the input vectors because $\tilde{u}$ itself does not always take binary values due to some transformation (21).

REFERENCES

[1] J. D. Aplevich, *Implicit Linear Systems. Lecture Notes in Control and Information Sciences* 152, Springer, 1991.
[2] A. Bemporad and M. Morari: Control of systems integrating logic, dynamics, and constraints, *Automatica*, vol. 35, pp. 407–427, 1999.
[3] A. Bemporad and N. Giorgetti, A Logic-Based Hybrid Solver for Optimal Control of Hybrid Systems, *Proc. 42nd IEEE Conf. on Decision and Contr.*, pp. 640–645, 2003.
[4] B. Bollobás, *Modern Graph Theory*. Springer, 1998.
[5] M. Ishikawa, T. Seki, J. Imura, and S. Hara: An efficient algorithm for optimal control of hybrid dynamical systems utilizing mode transition rule, *Proc. 43rd IEEE Conf. on Decision and Control*, pp. 1866–1871, 2004.
[6] K. Kobayashi and J. Imura: Modeling of Discrete Dynamics for Computational Time Reduction of Model Predictive Control, *Proc. of the 17th Int'l Symp. on Mathematical Theory of Networks and Systems*, pp. 628–633, 2006.
[7] K. Kobayashi and J. Imura: Minimality of Finite Automata Representation in Hybrid Systems Control, *Proc. of 10th Int'l Conf. on Hybrid Systems: Computation and Control*, *LNCS* 4416, pp. 343–356, Springer, 2007.
[8] M. Kuijper, *First-order Representations of Linear Systems*. Birkhäuser, 1994.
[9] A. Megretski: Robustness of Finite State Automata, *Multidisciplinary Research in Control: The Mohammed Dahleh Symp.*, pp. 147-160, 2002.
[10] 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.
[11] H. P. Williams, *Model building in mathematical programming*. 4th ed., John Willey & Sons, Ltd, 1999.
[12] http://www.ilog.com/products/cplex/