

Matrix Approach to Model Matching of Asynchronous Sequential Machines

Xiangru Xu and Yiguang Hong

Abstract—In this note, we propose a matrix-based approach for asynchronous sequential machines. Using semi-tensor product of matrices, we convert an asynchronous machine into a discrete-time bilinear system, and study its dynamics by investigating its structure matrices. We give simple algorithms for cycle detection and reachability analysis, and moreover, provide a control design method for the model matching of two input/state machines.

Index Terms—Asynchronous machines, cycle, model matching.

I. INTRODUCTION

Finite asynchronous sequential machines are machines that operate without clocking [7], [10]. Different from synchronous machines, asynchronous machines are endowed with some features such as the distinction between stable and unstable states and the fundamental mode operation. If the behavior of an asynchronous machine are undesired, it can be modified by connecting to another asynchronous machine that serves as a feedback corrective controller (referring to [8], [12], [14], [16], [17] and references therein).

Model matching is an important problem [2], and the model matching for finite state machines, related to the bisimulation of machines and the supervisory control [1], aims to find a controller for a given machine so that the behavior of the resulting composite system can “match” that of another given machine [3]. For asynchronous machines, the significant results for the input/state and input/output cases were addressed in [8], [12], while the generalized cases such as model matching inclusion and the existence of adversarial inputs were considered in [16], [17].

Matrices provide an elegant tool for finite state machines [7], [9], [13], [18], but a systematic matrix-based approach is still in need. On the other hand, Boolean networks, a special kind of finite machines, have been studied widely and systematically using the semi-tensor product (STP) of matrices in recent years [4], [5], [11]. Inspired by the success in Boolean networks, reachability analysis for finite automata using STP was discussed in [15].

In this note, we propose a matrix approach to the investigation of asynchronous machines and the model matching design problem by avoiding symbolic calculation. With the help of STP, the operation of a machine is first realized as matrix products by expressing the state, input and output as column vectors, and the transition and output functions as structure matrices. In this way, the properties of an asynchronous machine can be investigated by analyzing its corresponding structure matrices, where the cycle detection and the reachability analysis can be easily done. Furthermore, since the dynamics of a machine can be designed by suitable assignments for the structure matrices, a new control design method is proposed for the model matching problem of two input/state machines, where the controlled machine can be either with or without cycles.

Manuscript received April 22, 2012; revised November 23, 2012; accepted April 19, 2013. Date of publication April 24, 2013; date of current version October 21, 2013. Recommended by Associate Editor C. Hadjicostis.

The authors are with Key Laboratory of Systems and Control, Institute of Systems Science, Academy of Mathematics and Systems Science, Chinese Academy of Sciences, Beijing 100190, China (e-mail: xuxiangru@amss.ac.cn, yghong@iss.ac.cn).

Digital Object Identifier 10.1109/TAC.2013.2259957

Notations: I_n is the $n \times n$ identity matrix; $\mathbf{0}_{n \times m}$ is the $n \times m$ matrix with zero entry; $col_i M$ is the i -th column of matrix M ; $\delta_n^k := col_k I_n$ ($1 \leq k \leq n$), and $\Delta_n := \{\delta_n^1, \dots, \delta_n^n\}$; $\delta_n^0 := \mathbf{0}_n$; $\delta_n[i_1, i_2, \dots, i_m]$ is a compact form of matrix $[\delta_n^{i_1}, \delta_n^{i_2}, \dots, \delta_n^{i_m}]$; $M_{(p,q)}$ is the entry in the p -th row and q -th column of matrix M ; $tr(M)$ is the trace of M ; $|\cdot|$ is the cardinality of a set; $\mathcal{P}(X)$ is the set of all subsets of X ; \mathbb{Z}^+ is the set of positive integers; $\mathcal{P}(k) = \{i \in \mathbb{Z}^+ | i < k, k/i \in \mathbb{Z}^+\}$.

II. PRELIMINARIES: STP AND MACHINE

We introduce the semi-tensor product (STP) of matrices, which was first proposed by Cheng [5], [6].

Definition 2.1: For two matrices $M \in \mathbb{R}^{m \times n}$ and $N \in \mathbb{R}^{p \times q}$, their STP, denoted by $M \ltimes N$, is defined as $M \ltimes N := (M \otimes I_{s/n})(N \otimes I_{s/p})$, where s is the least common multiple of n and p and \otimes is the Kronecker product.

STP is consistent with the conventional matrix product. The symbol \ltimes may be omitted when there is no confusion. Given a matrix M , denote $M^1 = M$, $M^{i+1} = M \ltimes M^i$, $i \in \mathbb{Z}^+$.

STP has a so-called pseudo-swap property: for two column vectors $x \in \mathbb{R}^m$ and $y \in \mathbb{R}^n$, there is a unique swap matrix $W_{[m,n]} \in \mathbb{R}^{m \times m}$ depending only on n, m such that $W_{[m,n]}xy = yx$. Other useful properties of STP can be found in [6].

Next we consider an asynchronous (sequential) machine Σ

$$\Sigma = (E, Y, X, x_0, f, h) \quad (1)$$

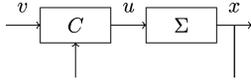
where E is the input alphabet, Y is the output alphabet, X is the set of states, x_0 is the initial state, $f: X \times E \rightarrow X$ (transition function) and $h: X \times E \rightarrow Y$ (output function) are partial functions [10]. x_0 may be omitted when the initial state is irrelevant or can be arbitrary. A pair of state and input $(x, u) \in X \times E$, where f and h are defined, is called a *valid pair*. E^* denotes the set of finite strings on E excluding the empty transition. Starting from x_0 , Σ accepts an input string $u_0 u_1 \dots \in E^*$ and operates according to the following equations:

$$x_{k+1} = f(x_k, u_k), \quad y_k = h(x_k, u_k), \quad k = 0, 1, 2, \dots \quad (2)$$

where state string $x_0 x_1 x_2 \dots$ and output string $y_0 y_1 y_2 \dots$ are generated. Σ is called an *input/state machine* if $Y = X$ and $y_k = x_k$ ($k \geq 0$), and then it can be denoted as (E, X, f, x_0) or simply (E, X, f) . A valid pair (x, u) is a *stable combination* if $f(x, u) = x$; otherwise it is called a *transient combination*. An asynchronous machine stays at a stable combination if the input is kept unchanged, while the transient combination initiates a chain of transitions.

Suppose that a transient combination (x, u) initiates a chain of transitions $x_1 = f(x, u), x_2 = f(x_1, u), \dots$, where input u keeps unchanged. If there is an integer $q \geq 1$ such that $x' := f(x_q, u)$ and (x', u) is a stable combination, then state x' is called the *next stable state* of x with the input u . The *stable transition function* s of Σ can be defined as a partial function $s: X \times E \rightarrow X$ for valid pairs (x, u) by $s(x, u) = x'$ if (x, u) has a next stable state x' and undefined otherwise. The stable transition function induces the *stable-state machine* of Σ , denoted as $\Sigma|_s := (E, Y, X, x_0, s, h_s)$.

If a valid pair of Σ has no next stable state, there is an infinite cycle involving a set of distinct states and a corresponding input. To be strict, if Σ has an infinite cycle with l states, say x_{i_1}, \dots, x_{i_l} with input u , then the machine operates on these states as follows: $x_{i_{k+1}} = f(x_{i_k}, u)$, $k = 1, 2, \dots, l-1$, $x_{i_1} = f(x_{i_l}, u)$. Denote the infinite cycle by $\mathcal{C} = \{x_{i_1}, \dots, x_{i_l}; u\}$, and the state set of \mathcal{C} by $\Theta(\mathcal{C})$. Clearly, the stable combination is a special case of infinite cycle with length 1.


 Fig. 1. Composite system Σ_c .

When Σ has no infinite cycles, asynchronous machines are restricted to the *fundamental mode* in most cases, where only one of the input and state may change at one step. When Σ has infinite cycles, the *semi-fundamental mode* is considered (referring to [10], [12], [14] for details).

If an asynchronous machine C , which serves as a feedback controller, is connected to an input/state machine Σ , a composite system denoted by Σ_c can be formed as shown in Fig. 1. The following proposition guarantees the fundamental mode operation of Σ_c [12].

Proposition 2.2: Suppose that Σ has no cycles. Then the system Σ_c operates in the fundamental mode if (i) Starting from a stable combination and in response to a change in the external input variable v of C , the output value of C does not change until C reaches its next stable state. (ii) In response to changes in the output of Σ , C does not commence any state transitions until Σ reaches its next stable state.

Definition 2.3: [7]: Let $\Sigma|_s$ and Σ'_s be the stable-state machines induced by $\Sigma = (E, Y, X, x_0, f, h)$ and $\Sigma' = (E, Y, X', x'_0, f', h')$, respectively. We call the two states x and x' *stably equivalent* if $\Sigma|_s$ starting from x , and Σ'_s starting from x' , admit the same permissible input strings and generate the same output string for each permissible input string. If $x_0 \equiv x'_0$, Σ and Σ' are called *stably equivalent* (denoted as $\Sigma = \Sigma'$).

Definition 2.4 (Model Matching Problem [14]): Let $\Sigma = (E, X, f)$ be an input/state asynchronous machine and $\Sigma' = (E, X, s')$ be a stable-state input/state machine without cycles. Find necessary and sufficient conditions for the existence of a controller C such that $\Sigma_{c|s} = \Sigma'$ for all initial states, and moreover, design the controller C if such C exists.

III. MATRIX EXPRESSION AND ANALYSIS

In this section, we provide a matrix expression for asynchronous machines and related analysis.

Consider Σ defined in (1) with $E = \{e_1, \dots, e_m\}$ and $X = \{x_1, \dots, x_n\}$. Identify x_i and e_j with their respective *vector forms* δ_n^i ($1 \leq i \leq n$) and δ_m^j ($1 \leq j \leq m$) (denoted as $x_i \sim \delta_n^i, e_j \sim \delta_m^j$). Then X and E are identified with Δ_n and Δ_m , respectively. Hence, the valid pair (x_i, e_j) can be expressed as (δ_n^i, δ_m^j) ($1 \leq i \leq n, 1 \leq j \leq m$). In the following, we use both the vector form and the symbolic form without discrimination when there is no confusion.

The transition structure matrix $F_i \in \mathbb{R}^{n \times n}$ associated with input e_i is defined as

$$F_{i(j,k)} = \begin{cases} 1, & \text{if } \delta_n^j = f(\delta_n^k, \delta_m^i); \\ 0, & \text{otherwise.} \end{cases} \quad (3)$$

The *transition structure matrix* (TSM) of Σ is defined as $F = [F_1, F_2, \dots, F_m] \in \mathbb{R}^{n \times mn}$. Moreover, the output structure matrix H corresponding to the output function h and the stable TSM corresponding to the stable transition function s can be defined similar to (3).

Remark 3.1: Clearly, there is an equivalent relationship between the structure matrix F (or H) and the transition function f (or h) of Σ , and we can derive one from the other. Moreover, a valid pair (δ_n^i, δ_m^j) corresponds to the non-zero columns $col_i(F_j)$ and $col_j(H_i)$.

Σ evolves with a given input at each step. The *vector form of input* of Σ at step t ($t \geq 1$) is a column vector $u(t) \in \Delta_m$, and

$u(t) = \delta_m^i$ if input e_i is given at step t . In this way, an input string $e = e_{i_1}e_{i_2}\dots e_{i_t} \in E^*$ can be identified with a column vector $\times_{j=1}^t u(j) := u(1)u(2)\dots u(t) \in \Delta_{tm}$ ($e \sim \times_{j=1}^t u(j)$), where $e_{i_j} \sim u(j)$ ($1 \leq j \leq t$). The *vector form of state* of Σ at step t is a column vector $x(t) = (x^1(t), x^2(t), \dots, x^n(t))^T \in \mathbb{R}^n$, where $x^i(t)$ ($1 \leq i \leq n$) equals the number of the different transitions, by which the initial state $x(1)$ can reach δ_n^i with a given permissible input string of length $t - 1$. The *vector form of output* of Σ can be defined correspondingly. The following result is essential for the matrix expression of asynchronous machines.

Theorem 3.2: Suppose that F and H are the transition and output structure matrices of an asynchronous machine Σ , respectively, and $x(1)$ is its initial state. Then the dynamics of Σ can be described by the following equations:

$$x(t+1) = Fu(t)x(t), \quad y(t) = Hu(t)x(t), \quad t \in \mathbb{Z}^+ \quad (4)$$

where $x(t)$, $u(t)$, and $y(t)$ are the vector forms of the state, input, and output of Σ , respectively.

This theorem can be easily proved by referring to Theorem 1 in [15], so its proof is omitted here. Actually, (4) is the matrix-form interpretation of (2). Moreover, for Σ operating under the (semi-)fundamental mode, we should keep input $u(t+1) = u(t)$ until Σ reaches a stable combination (or infinite cycle), just as we should let $u_{t+1} = u_t$ in (2) [12], [14].

In Sections III-A–III-C, we further discuss the input/state machine $\Sigma = (E, X, f)$ with $|E| = m$, $|X| = n$.

A. Generalized Stable-State Matrix Expression

Assume that Σ has κ infinite cycles $\mathcal{C}^1, \dots, \mathcal{C}^\kappa$, without loss of generality. Associate \mathcal{C}^i with a new state x_{n+i} ($1 \leq i \leq \kappa$), called a *cycle state*. Correspondingly, the states in X are called *regular states*. Call the new set $\tilde{X} = \{x_1, \dots, x_n, x_{n+1}, \dots, x_{n+\kappa}\}$ the *augmented state set*, and each element in \tilde{X} the *generalized state* [14]. In vector forms, $\tilde{X} = \{\delta_{n+\kappa}^1, \dots, \delta_{n+\kappa}^{n+\kappa}\}$. Identify δ_n^i with $\delta_{n+\kappa}^i$ when there is no confusion, and the set \tilde{X} is $\tilde{X} := \{\delta_{n+\kappa}^1, \dots, \delta_{n+\kappa}^{n+\kappa}\} \subset \tilde{X}$.

$(\delta_{n+\kappa}^i, \delta_m^j)$ is a *generalized valid pair* of Σ if $(\delta_{n+\kappa}^i, \delta_m^j)$ ($1 \leq i \leq n$) is a valid pair of Σ or $(\delta_{n+\kappa}^k, \delta_m^j)$ is a valid pair of Σ for each $\delta_{n+\kappa}^k \in \mathcal{C}^i$ ($n+1 \leq k \leq n+\kappa$). $(\delta_{n+\kappa}^i, \delta_m^j)$ is a *generalized stable combination* of Σ if $(\delta_{n+\kappa}^i, \delta_m^j)$ ($1 \leq i \leq n$) is a stable combination of Σ or δ_m^j is the corresponding input of \mathcal{C}^i ($n+1 \leq i \leq n+\kappa$).

With the augmented state set \tilde{X} , f induces the *generalized transition function* $\tilde{f} : \tilde{X} \times E \rightarrow \mathcal{P}(\tilde{X})$. First, a partial function $f^e : X \times E \rightarrow \tilde{X}$ over all valid pairs $(\delta_{n+\kappa}^i, \delta_m^j) \in X \times E$ of Σ is defined: $f^e(\delta_{n+\kappa}^i, \delta_m^j) = x^{n+l}$ if $f(\delta_{n+\kappa}^i, \delta_m^j) \in \Theta(\mathcal{C}^l)$ ($1 \leq l \leq \kappa$), $f^e(\delta_{n+\kappa}^i, \delta_m^j) = f(\delta_{n+\kappa}^i, \delta_m^j)$ otherwise. Then \tilde{f} is defined as follows:

$$\tilde{f}(\delta_{n+\kappa}^i, \delta_m^j) = \begin{cases} f^e(\delta_{n+\kappa}^i, \delta_m^j), & 1 \leq i \leq n; \\ f^e(\Theta(\mathcal{C}^i), \delta_m^j), & n+1 \leq i \leq n+\kappa, \end{cases} \quad (5)$$

where $f^e(\Theta(\mathcal{C}^i), \delta_m^j) := \{\delta_{n+\kappa}^k | \delta_{n+\kappa}^k = f^e(\delta_{n+\kappa}^i, \delta_m^j), \delta_{n+\kappa}^i \in \Theta(\mathcal{C}^i)\}$.

Then the *generalized stable transition function* $\tilde{s} : \tilde{X} \times E \rightarrow \mathcal{P}(\tilde{X})$ of Σ is defined as $\tilde{s}(\delta_{n+\kappa}^i, \delta_m^j) = \delta_{n+\kappa}^k$ if $(\delta_{n+\kappa}^i, \delta_m^j)$ has a next generalized stable state $\delta_{n+\kappa}^k$, and undefined otherwise. Note that \tilde{s} may be multi-valued for a cycle state $\delta_{n+\kappa}^i$ ($n \leq i \leq n+\kappa$), and \tilde{s} is just the stable transition function s of Σ when Σ has no cycles.

Function \tilde{s} induces the *generalized stable-state machine* $\Sigma|_{\tilde{s}} = (E, \tilde{X}, \tilde{s})$. Similar to (3), the generalized stable TSM \tilde{F} of $\Sigma|_{\tilde{s}}$ is defined as $\tilde{F} = [\tilde{F}_1, \dots, \tilde{F}_m] \in \mathbb{R}^{(n+\kappa) \times (n+\kappa)m}$, where

$$\tilde{F}_{i(j,k)} = \begin{cases} 1, & \text{if } \delta_{n+\kappa}^j \in \tilde{s}(\delta_{n+\kappa}^k, \delta_m^i); \\ 0, & \text{otherwise.} \end{cases} \quad (6)$$

Let $x(t) \in \mathbb{R}^{n+\kappa}$ and $u(t) \in \Delta_m$ be the vector forms of the state and input of $\Sigma_{|\bar{s}}$, respectively. Then similar to (4), the generalized stable-state matrix expression of $\Sigma_{|\bar{s}}$ is

$$x(t+1) = \tilde{F}u(t)x(t), \quad t \in \mathbb{Z}^+. \quad (7)$$

Remark 3.3: Both (4) and (7) characterize the dynamics of Σ with infinite cycles. However, (7) describes the external behavior that can be observed by a user when Σ operates under the semi-fundamental mode, while (4) is the implicit description of the inner dynamics of Σ . Furthermore, since semi-fundamental mode of Σ is equivalent to the fundamental mode operation of $\Sigma_{|\bar{s}}$ [14], (7) will be useful in Sections III-C and IV.

B. Cycle Detection

The method proposed above can help us to detect all the cycles and their specific elements easily given the transition function f . With input $u(t) = e_i$, (4) can be rewritten as

$$x(t+1) = F_i x(t). \quad (8)$$

Lemma 3.4: x is in a cycle of length k with the input δ_m^i if and only if k is the least positive integer to satisfy $x = F_i^k x$.

Proof: “Sufficiency”: With input δ_m^i , the states that Σ undergoes from x are $x, F_i x, F_i^2 x, \dots$. Since k is the least positive integer to make $x = F_i^k x$ hold, $\{x, F_i x, F_i^2 x, \dots, F_i^{k-1} x\}$ are all distinct. Therefore, $\{x, F_i x, F_i^2 x, \dots, F_i^{k-1} x; \delta_m^i\}$ is a cycle of length k . “Necessity”: It is obvious that $x = F_i^k x$ holds. If there exists an integer $k' < k$ and $x = F_i^{k'} x$, then it contradicts the fact that $\{x, F_i x, \dots, F_i^{k-1} x\}$ are all distinct. \square

The following corollary is straightforward.

Corollary 3.5: A valid pair (δ_n^j, δ_m^i) is a stable combination of Σ if and only if $F_{i(j,j)} = 1$.

Then let us consider the number of cycles and the specific elements of a given cycle. Denote the number of cycles of length r by N_r . Here is a simple criterion to determine N_r inductively.

Theorem 3.6: N_r can be determined by the following equations: $N_1 = \sum_{i=1}^m \text{tr}(F_i)$, $N_r = \sum_{i=1}^m [\text{tr}(F_i^r) - \sum_{k \in \mathcal{P}(r)} k N_k] / r$, $2 \leq r \leq n$.

Proof: Obviously, $r \leq n$. Corollary 3.5 gives N_1 immediately. If $r \geq 2$, then for any i ($1 \leq i \leq m$), consider the valid pair (δ_n^j, δ_m^i) where δ_n^j is an element of a cycle of length r . Clearly, $F_i^r \delta_n^j = \delta_n^j$, which implies $\text{tr}(F_i^r)$ increases by 1 due to such δ_n^j . Furthermore, for a valid pair (δ_n^q, δ_m^i) , where δ_n^q is an element of a cycle of length $p \in \mathcal{P}(r)$, we have $F_i^r \delta_n^q = \delta_n^q$, which implies $\text{tr}(F_i^r)$ also increases by 1 due to such δ_n^q . We subtract the value contributed by δ_n^q from $\text{tr}(F_i^r)$ for each i . Since i can be arbitrary, we sum up the value and then the conclusion follows. \square

For convenience, we refer to “cycle” to denote an infinite cycle with length greater than 1 in the sequel.

Next, an algorithm with polynomial complexity is given to identify all the states of a cycle.

Algorithm 3.7: Algorithm of Cycle Detection

1. For $s = 1, \dots, n$, $i = 1, \dots, m$, calculate set $\Phi_{is} = \{j | F_{i(j,j)}^s = 1\}$.
 2. For $r = 2, \dots, n$, calculate set $\Psi_{ir} = \Phi_{ir} \cap_{j \in \mathcal{P}(r)} \Phi_{ij}^r$, where $\Phi_{ij}^r = \{1, \dots, n\} \setminus \Phi_{ij}$.
-

Theorem 3.8: δ_n^p is in a cycle of length k ($k \geq 2$) with input δ_m^i if and only if $p \in \Psi_{ik}$.

Proof: Since $F_{i(j,j)}^s = 1$ is equivalent to $F_i^s \delta_n^j = \delta_n^j$, the set Φ_{is} , if nonempty, contains all the states in cycles (with input δ_m^i)

of length equal to either s or an element in $\mathcal{P}(s)$. Clearly, $\Psi_{ik} = \Phi_{ik} \cap (\bigcup_{j \in \mathcal{P}(k)} \Phi_{ij})^c$, and the set $\bigcup_{j \in \mathcal{P}(k)} \Phi_{ij}$ contains all the states in cycles of length j with $j \in \mathcal{P}(k)$. Thus, Ψ_{ik} is the set of states that are in a cycle of length k . Furthermore, by Lemma 3.4, each cycle of length k with input δ_m^i takes the form of $\{x, F_i x, \dots, F_i^{k-1} x; \delta_m^i\}$. \square

As shown above, the number and elements of cycles of Σ can be easily determined by routine calculations of matrices. A Matlab toolbox is also available at <http://lsc.amss.ac.cn/dcheng/> for STP calculations. In [14], several symbolic operators were defined and propositions and lemmas were given for the cycle detection procedure. Compared with these results, the matrix approach is simpler as illustrated in the next example that was considered in [14].

Example 3.9: Given an input/state machine $\Sigma = (E, X, f)$ with $X = \{x_1, x_2, x_3\}$, $E = \{a, b, c\}$, and f shown as follows:

	a	b	c
x_1	x_1	x_3	x_1
x_2	x_2	x_3	x_3
x_3	x_2	x_2	x_3

The TSM of Σ is $F = \delta_3[1, 2, 2, 3, 3, 2, 1, 3, 3]$. By Corollary 3.6, (δ_3^1, δ_3^1) , (δ_3^2, δ_3^1) , (δ_3^1, δ_3^3) and (δ_3^3, δ_3^3) are the stable combinations. Implementing Algorithm 3.7, we have $\Phi_{11} = \Phi_{12} = \Phi_{13} = \{1, 2\}$; $\Phi_{21} = \emptyset$, $\Phi_{22} = \{2, 3\}$, $\Phi_{23} = \emptyset$; $\Phi_{31} = \Phi_{32} = \Phi_{33} = \{1, 3\}$. Therefore, $\Psi_{22} = \Phi_{22} \cap \Phi_{21}^c = \{2, 3\}$ is the only nonempty set, implying $\mathcal{C}^1 = \{\delta_3^2, \delta_3^3; \delta_3^2\}$ is the unique cycle of Σ . This coincides with the result of Example 3 in [14], but our procedure is much simpler.

C. Reachability Analysis

Suppose that Σ has κ cycles. $\Sigma_{|\bar{s}}$ is the induced generalized stable-state machine of Σ . Here we give the reachability analysis for the generalized states of $\Sigma_{|\bar{s}}$.

Definition 3.10: State $x' \in \tilde{X}$ is called *stably reachable* from $x \in \tilde{X}$ if there is an input string $e \in E^*$ such that $x' \in \tilde{s}(x, e)$. The transition induced by e from x is called a *deterministic transition* if $x' = \tilde{s}(x, e)$, and x' is called *deterministically stably reachable* from x .

Remark 3.11: The fact that function \tilde{s} may be multi-valued implies that critical races [10], [12] may arise during the transitions. Deterministic reachability between two states reflects the potential of Σ to move from a state to another state in a deterministic way, masking the non-determinism (critical races) during the transitions.

By the property of STP, (7) is equivalent to

$$x(t+1) = \bar{F}W_{[n+\kappa, m]}x(t)u(t) \quad (9)$$

where $\bar{F} = \tilde{F}W_{[n+\kappa, m]}$ and $W_{[n+\kappa, m]}$ is the swap matrix [5]. Then we give an algorithm.

Algorithm 3.12: Algorithm of Reachability Analysis

1. For $t = 1, \dots, n + \kappa$, calculate \bar{F}^t and split it into $n + \kappa$ equal blocks: $\bar{F}^t = [\bar{F}_1^t, \dots, \bar{F}_{n+\kappa}^t]$.
 2. Calculate the set $\Lambda_{ij}^t = \{k | 1 \leq k \leq m^t, \text{col}_k \bar{F}_i^t = p \delta_{n+\kappa}^j \text{ for some } p, p \in \mathbb{Z}^+\}$.
-

Lemma 3.13: For an asynchronous machine Σ that has κ infinite cycles and operates under the semi-fundamental mode, the generalized state $\delta_{n+\kappa}^j$ is deterministically reachable from $\delta_{n+\kappa}^i$ via an input string $\delta_{m^t}^i$ of length t if and only if $k \in \Lambda_{ij}^t$.

The proof is simple based on (9) and omitted here. Therefore, there is a deterministic transition from x_i to x_j if and only if $\cup_{l=1}^{n_1+n_2} \Lambda_{ij}^l \neq \emptyset$. Note that the corresponding intermediate states and then the transitions can be also got easily by (9).

A numerical matrix called *skeleton matrix* $R(\Sigma) \in \mathbb{R}^{n \times n}$ was defined for Σ without cycles in [12] and extended to Σ with cycles in [14]. In general, the (i, j) th entry of $R(\Sigma)$ is 1 if x_j is deterministically stably reachable from x_i , and it is 0 otherwise. Clearly, $R(\Sigma)$ can be easily constructed by Algorithm 3.12 and Lemma 3.13.

If Σ has no cycles, all the transitions are deterministic given an input string. We simply call x' stably reachable from x if $x' = s(x, e)$ for some $e \in E^*$.

Lemma 3.14: Suppose that Σ has no cycles. For any i ($1 \leq i \leq m$), there is a positive integer τ_i such that $F_i^k = F_i^{\tau_i}$ for any integer $k, k \geq \tau_i$. Furthermore, $F_i = F_i^{\tau_i}$ for any i .

The proof is simple and therefore omitted for space limitations. The following lemma can be proved similar to Theorem 3 in [15] and is omitted here.

Lemma 3.15: Suppose that Σ has no cycles. Then δ_n^q is stably reachable from δ_n^p via an input string of length t if and only if $(\sum_{i=1}^m \tilde{F}_i)^t \binom{p, q}{} > 0$.

Clearly, $R(\Sigma)$ can be obtained by Lemmas 3.14 and 3.15 easily in this case.

Similar to the cycle detection, detailed methods were proposed to construct a skeleton matrix and its generalized forms in [8], [12], [14], with many additional definitions, lemmas, and algorithms. Compared with the existing results, the algorithm here only involves matrix products and is much easier, because the matrix expressions of machines are simple and well-developed matrix analysis tools can be used conveniently.

IV. MODEL MATCHING CONTROL DESIGN

Consider the composite system Σ_c as shown in Fig. 1, where $\Sigma = (E, X, f)$ has no cycles with $|X| = n_1$ and $|E| = m$. The controller C is $C = (E \times X, E, \Xi, \xi_0, f_c, h_c)$, where $E \times X$ and E are the input and output alphabets, respectively; Ξ is the state set with $|\Xi| = n_2$; f_c and h_c are the transition and output functions, respectively. Every valid pair in C has a next stable state. Σ_c is $\Sigma_c = (E, X, \Xi \times X, \phi, \eta)$, where E and X are the same as Σ , $\Xi \times X$ is the state set, transition function $\phi : (\Xi \times X) \times E \rightarrow (\Xi \times X) : \phi((\xi, x), v) = (\xi', x')$ where $\xi' = f_c(\xi, (v, x))$, $x' = f(x, h_c(\xi, (v, x)))$, output function $\eta : (\Xi \times X) \times E \rightarrow X : \eta((\xi, x), v) = x$.

Denote $x_\Sigma(t)$, $u_\Sigma(t)$, and $y_\Sigma(t)$ as the respective vector forms of the state, input, and output for Σ , $x_c(t)$, $u_c(t)$, $y_c(t)$ as those for C , $x(t)$, $u(t)$, $y(t)$ as those for Σ_c , and $v(t)$ as the vector form of the external input. According to Fig. 1, we have $y(t) = x_\Sigma(t)$, $u_\Sigma(t) = y_c(t)$, $u_c(t) = v(t)x_\Sigma(t)$, and $x(t) = x_c(t)x_\Sigma(t)$.

Let \tilde{G} and \tilde{H} be the TSMs of f_c and h_c , respectively. Then the matrix expression of C is $x_c(t+1) = \tilde{G}u_c(t)x_c(t)$ and $y_c(t) = \tilde{H}u_c(t)x_c(t)$. Put $u_c(t) = v(t)x_\Sigma(t)$ into the above equations, and let $G = \tilde{G}(I_m \otimes W_{[n_2, n_1]})$ and $H = \tilde{H}(I_m \otimes W_{[n_2, n_1]})$. Then we have

$$\begin{aligned} C : x_c(t+1) &= Gv(t)x_c(t)x_\Sigma(t), \\ y_c(t) &= Hv(t)x_c(t)x_\Sigma(t). \end{aligned} \quad (10)$$

Take F as the TSM of Σ , and then the matrix expression of Σ is $x_\Sigma(t+1) = Fu_\Sigma(t)x_\Sigma(t)$, $y_\Sigma(t) = x_\Sigma(t)$. Putting $u_\Sigma(t) = Hv(t)x_c(t)x_\Sigma(t)$ into the above equations leads to the matrix expression of Σ

$$\begin{aligned} \Sigma : x_\Sigma(t+1) &= FH(I_{mn_2} \otimes \Phi_{n_1})v(t)x_c(t)x_\Sigma(t), \\ y_\Sigma(t) &= x_\Sigma(t). \end{aligned} \quad (11)$$

Denote K as the TSM of Σ_c . We obtain the matrix expression of Σ_c from (10) and (11) by some calculations

$$\Sigma_c : x(t+1) = K v(t)x(t), \quad y(t) = \underbrace{[I_{n_1}, \dots, I_{n_1}]}_{n_2} x(t) \quad (12)$$

where $n = n_1 n_2$, $K = G(I_{mn} \otimes (FH(I_{mn_2} \otimes \Phi_{n_1})))$.

The next result shows the relations of F , G , H and K .

Lemma 4.1: Let $H = \delta_m[\tau_1, \dots, \tau_{mn}]$ and $F = [F_1, \dots, F_m]$ be the matrices in (10) and (11), respectively. Let matrix $F_0 = \mathbf{0}_{n_1 \times n_1}$. For any p ($1 \leq p \leq nm$) with $p = (i-1)n_1 + j$ ($1 \leq i \leq mn_2, 1 \leq j \leq n_1, i, j \in \mathbb{Z}^+$), we have

$$\text{col}_p K = \text{col}_p G \times \text{col}_j F_{\tau_p}. \quad (13)$$

Proof: Consider (10) and (11). By properties of STP, $\text{col}_p K = \text{col}_p G \times \text{col}_p FH(I_{mn_2} \otimes \Phi_{n_1})$ for any p [5]. Since $FH = [F_{\tau_1}, \dots, F_{\tau_p}]$ and $\Phi_{n_1} = \text{blockdiag}[\delta_{n_1}^1, \delta_{n_1}^2, \dots, \delta_{n_1}^{n_1}]$, we have $FH(I_{mn_2} \otimes \Phi_{n_1}) = [\text{col}_1 F_{\tau_1}, \dots, \text{col}_{n_1} F_{\tau_{n_1}}, \dots, \text{col}_1 F_{\tau_{(p-n_1)}}, \dots, \text{col}_{n_1} F_{\tau_p}]$. For any p with $p = (i-1)n_1 + j$, it is clear that $\text{col}_p FH(I_{mn_2} \otimes \Phi_{n_1}) = \text{col}_j F_{\tau_p}$. Then the conclusion follows. \square

Let matrix K be partitioned as $K = [K_1, \dots, K_m]$ with $K_l \in \mathbb{R}^{n \times n}$ ($1 \leq l \leq m$), and K_l partitioned as $K_l = (K_l^{ij})$ with $K_l^{ij} \in \mathbb{R}^{n_1 \times n_1}$ ($1 \leq i, j \leq n_2$) as the (i, j) th block submatrix. The following corollary is not hard to obtain by (13) and the definition of TSM.

Corollary 4.2: Let ϕ, η be the transition function and output function of Σ_c , respectively. For any i, j, l ($1 \leq i, j \leq n_2, 1 \leq l \leq m$), $\text{col}_\alpha K_l^{ij} = \delta_{n_1}^\beta$ for some α, β ($1 \leq \alpha, \beta \leq n_1$) is equivalent to $\phi(\delta_{n_2}^j, \delta_{n_1}^\alpha, \delta_m^l) = (\delta_{n_2}^i, \delta_{n_1}^\beta)$ and $\eta(\delta_{n_2}^j, \delta_{n_1}^\alpha, \delta_m^l) \in \{\delta_m^s | \text{col}_\alpha F_s = \delta_{n_1}^\beta, 1 \leq s \leq m\}$.

The model matching problem of two input/state machines Σ and Σ' was studied for Σ without cycles in [12] and with cycles in [14], respectively, and the corresponding sufficient and necessary conditions can be summarized as follows:

Proposition 4.3: Given input/state machines $\Sigma = (E, X, f)$ and $\Sigma' = (E, X, s')$, where Σ' is a stable-state machine and has no cycles, there is a controller C such that $\Sigma_{c|s} = \Sigma'$ and Σ_c operates in (semi-)fundamental mode if and only if the skeleton matrix $R(\Sigma) \geq R(\Sigma')$.

Actually, the sufficiency part can be easily proved by definition [12], [14]. For the necessity part, it requires the design of a suitable controller, which was obtained by combining some control modules designed for each specification of model mismatch between Σ and Σ' .

Then the controller can be easily constructed using the matrix approach. Clearly, the design for C is equivalent to finding the suitable G and H in (10). Since F is fixed with a given f , the achievable dynamics of Σ_c via C are equivalent to the possible structure of K in terms of G, H . Conversely, we can derive G, H from a valid K by (13). Furthermore, C should guarantee the fundamental mode of Σ_c as shown in Proposition 2.2, where the constraints for C will be transformed later to the constraints for K .

Let $\tilde{\phi}$ be the stable transition function of Σ_c . If $\Sigma_{c|s} = \Sigma'$, then clearly Σ_c has no cycles and for every valid pair (x, v) of Σ' there is a state $\xi \in \Xi$ such that $\eta(\tilde{\phi}(x, \xi, v), \xi, v) = s'(x, v)$ [14]. Denote $\tilde{F}' = [\tilde{F}'_1, \dots, \tilde{F}'_m]$ as the stable TSM of Σ' and $\tilde{v}(t), \tilde{x}(t)$ as the respective vector forms of its input and state. Then by (12), $\Sigma_{c|s} = \Sigma'$ is equivalent to: for each valid pair $(\tilde{x}(t_0), \tilde{v}(t_0))$ of Σ' , if $(x_\Sigma(t_0), v(t_0))$ is a stable combination of Σ with $\tilde{x}(t_0) = x_\Sigma(t_0)$, $\tilde{v}(t_0) = v(t_0) = \delta_m^l$, then there is an integer τ ($\tau \geq 0$) and $x_c(t_0 + \tau) \in \Delta_{n_2}$ such that

$$\underbrace{[I_{n_1}, \dots, I_{n_1}]}_{n_2} K_l^{\tau+1} x_c(t_0 + \tau)x_\Sigma(t_0) = \tilde{F}'_l \tilde{x}(t_0). \quad (14)$$

Denote \tilde{F} as the stable TSM of Σ . The *mismatch set* V of Σ and Σ' is defined as $V = \{V_1, \dots, V_m\}$, where $V_s = \{v_1^s, \dots, v_{\tau_s}^s\}$ ($1 \leq s \leq m$) with $j \in V_s$ ($1 \leq j \leq n_1$) if $\text{col}_j \tilde{F}_s \neq \text{col}_j \tilde{F}'_s$. Recalling Remark 3.1, V specifies all the valid pairs of Σ' that have different next stable states with Σ . Denote $q = |V|$. Take $i_1 = v_1^1, \dots, i_{\tau_1} = v_{\tau_1}^1, i_{\tau_1+1} = v_1^2, \dots, i_q = v_{\tau_m}^m$, and correspondingly take $j_1 = 1, \dots, j_{\tau_1} = 1, j_{\tau_1+1} = 2, \dots, j_q = m$. Then we have the following result.

Theorem 4.4: Consider two cycle-free input/state machines $\Sigma = (E, X, f)$ and $\Sigma' = (E, X, s')$, where Σ' is a stable-state machine, $|E| = m$, $|X| = n_1$. $\tilde{F} = [\tilde{F}_1, \dots, \tilde{F}_m]$ and $\tilde{F}' = [\tilde{F}'_1, \dots, \tilde{F}'_m]$ are the stable TSMs of Σ and Σ' , respectively. For the mismatch set V of Σ and Σ' , if for any i, j such that $i \in V_j$, $\text{col}_i \tilde{F}'_j$ is stably reachable from $\delta_{n_1}^i$ in Σ , then there exists a controller C for which $\Sigma_{c|s} = \Sigma'$ and Σ_c operates in the fundamental mode.

Proof: We will show that under the given conditions there exists a matrix K that satisfies (14) and guarantees that Σ_c operates under the fundamental mode. Then the controller C exists and can be obtained by Lemma 4.1. Our proof is mainly based on (13) and Corollary 4.2.

For any i ($1 \leq i \leq m$), define a set $I^i = \{j | F_{i(j,j)} = 1\}$. Note that $j \in I^i$ implies $(\delta_{n_1}^j, \delta_m^i)$ is a stable combination of Σ . For any k ($1 \leq k \leq q$), let $e_k = \delta_m^{k_1} \dots \delta_m^{k_{r_k}}$ be the input string of minimal length r_k so that $\text{col}_{i_k} \tilde{F}'_{j_k} = s(\delta_{n_1}^{i_k}, e_k)$, where s is the stable transition function of Σ and the intermediate states are $\delta_{n_1}^{w_{k1}}, \dots, \delta_{n_1}^{w_{kr_k}}$. Denote integers $a_1 = 0, a_2 = r_1, \dots, a_{q+1} = \sum_{j=1}^q r_j$. Then the number of states of C is $n_2 = a_{q+1} + 2 = \sum_{j=1}^q r_j + 2$.

Assignments for K are given as follows:

- Step 1: For any i ($1 \leq i \leq m$), take $\text{col}_j K_i^{11} = \text{col}_j F_i$ if $j \in \{1, \dots, n_1\} \setminus (V \cap I^i)$; take $K_j^{21} = \text{col}_j F_i$ if $j \in V \cap I^i$. Each $j \in V \cap I^i$ indicates that Σ has reached a stable combination $(\delta_{n_1}^j, \delta_m^i)$ and then C can change from state $\delta_{n_2}^1$ to $\delta_{n_2}^2$ as shown by assigning $K_j^{21} = \text{col}_j F_i$ (fundamental mode). This step keeps the output of C at δ_m^i by (13).
- Step 2: We have 3 cases. (2.1): For any i ($1 \leq i \leq m$) and $j = v_k^i \in V_i = \{v_1^i, \dots, v_{\tau_i}^i\}$, take $\text{col}_{v_k^i} K_i^{(a_{\lambda+3})^2} = \text{col}_{v_k^i} F_{i_{\lambda k}}$ with $i_{\lambda} = v_k^i$. For any j in this case, $\delta_{n_1}^j$ is in a stable combination $(\delta_{n_1}^j, \delta_m^i)$ by step 1, and then when Σ_c receives the external input δ_m^i with $j \in V_i$, it changes from state $\delta_{n_2}^2 \delta_{n_1}^j$ to $\delta_{n_2}^{\lambda+2} \delta_{n_1}^j$ with p satisfying $\delta_{n_1}^p = \text{col}_{v_k^i} F_a$ (fundamental mode). Furthermore, for any such j , the output of C is δ_m^i when $j \in I^b$ ($1 \leq b \leq m$).
- (2.2): For any $j \in I^i \setminus V_i$, take $\text{col}_j K_i^{22} = \text{col}_j F_i$. Then Σ_c receives input δ_m^i ($j \in I^i \setminus V_i$) and stays in a stable combination $(\delta_{n_2}^2 \delta_{n_1}^j, \delta_m^i)$ until the external input changes (fundamental mode).
- (2.3): For any $j \in V \setminus (V_i \cup I^i)$, take $\text{col}_j K_i^{12} = \text{col}_j F_i$. Then Σ_c receives the external input δ_m^i with $j \in V \setminus (V_i \cup I^i)$ and it changes to $(\delta_{n_2}^1 \delta_{n_1}^p, \delta_m^i)$ with p satisfying $\delta_{n_1}^p = \text{col}_j F_i$.
- Step 3: To design the q control modules, where the s -th ($1 \leq s \leq q$) basic module is a matrix of $n_1 r_s \times n_1 r_s$ that drives $\delta_{n_1}^{i_s}$ to $\text{col}_{i_s} \tilde{F}'_{j_s}$, we consider the s -th module and suppose $i_s \in V_\gamma$ for some γ ($1 \leq \gamma \leq m$) without loss of generality.
- (3.1): Take $K_\gamma^{-(a_s+3)(a_s+3)} = F_{i_{s1}}$, and then $\text{col}_{w_{s1}} K_\gamma^{-(a_s+3)(a_s+3)} = \delta_{n_1}^0$, $\text{col}_{w_{s1}} K_\gamma^{-(a_s+4)(a_s+3)} = \text{col}_{w_{s1}} F_{i_{s1}}; \dots$; take $K_\gamma^{-(a_s+r_s+1)(a_s+r_s+1)} = F_{i_{s(r_s-1)}}$ and then $\text{col}_{w_{s(r_s-1)}} K_\gamma^{-(a_s+1)(a_s+r_s+1)} = \delta_{n_1}^0$, $\text{col}_{w_{s(r_s-1)}} K_\gamma^{-(a_s+1)(a_s+r_s+1)} = \text{col}_{w_{s(r_s-1)}} F_{i_{s(r_s-1)}}$. In this case, the output of C at $\delta_{n_2}^i$ ($a_s + 3 \leq i \leq a_{s+1} - 1$) is kept a constant $\delta_m^{i_s}$. Since the w_{si} -th column of $K_\gamma^{-(a_s+i)(a_s+i)}$ indicates that Σ has reached a stable combination $(\delta_{n_1}^{w_{si}}, \delta_m^{i_s})$, Σ_c can change from $\delta_{n_2}^{a_s+i} \delta_{n_1}^{w_{si}}$ to $\delta_{n_2}^{a_s+i+1} \delta_{n_1}^{w_{si}}$ with $\delta_{n_1}^p = \text{col}_{w_{s(r_s-1)}} F_{i_{s(r_s-1)}}$

(fundamental mode), as shown by the w_{si} -th assignment for $K_\gamma^{-(a_s+i+1)(a_s+i)}$.

(3.2): Take $K_\gamma^{-a_s+1} = F_{i_{s\tau_s}}$. In this case Σ_c stays in a stable combination $(\delta_{n_2}^{a_s+1+2} \delta_{n_1}^{w_{s\tau_s}}, \delta_m^i)$ when the external input is kept δ_m^i (fundamental mode).

(3.3): Take $K_j^{1a_s+1} = F_j, \forall j \in \{1, \dots, m\} \setminus \gamma$. In this case C changes from state $\delta_{n_2}^{a_s+1+2}$ to $\delta_{n_2}^1$ in response to the external input changing from δ_m^i to δ_m^j .

In all the above situations, the assigned K guarantees that C satisfies the conditions of Proposition 2.2, implying that Σ_c operates in the fundamental mode. Finally, it is found that (14) holds for each valid pair of Σ' , where $x_c(t_0 + \tau)$ in (14) was obtained in (2.2) and (3.2). Thus, the controller C makes $\Sigma_{c|s} = \Sigma'$ hold and the fundamental mode satisfied. By (13), G, H can be obtained from K and the above statements, which give functions f_c, h_c , respectively. \square

If $R(\Sigma) \geq R(\Sigma')$, conditions of Theorem 4.4 are satisfied and therefore, the necessity part of Proposition 4.3 for the controller construction is proved. Different controllers can be given easily by different assignments for K if we choose different stable transition paths (may not be minimal length) between the states of Σ . The number n_2 of the states of C here is the same as that given in [12] and it can be further minimized by standard methods [10].

Example 4.5: Consider input/state machines $\Sigma = (E, X, f)$ and $\Sigma' = (E, X, f')$ with $E = \{a, b, c\}$, $X = \{x_1, x_2, x_3\}$, and f and f' shown as follows:

f	a	b	c	f'	a	b	c
x_1	x_1	x_2	x_1	x_1	x_1	x_3	x_1
x_2	x_3	x_2	x_2	x_2	x_1	x_2	x_2
x_3	x_3	x_3	x_1	x_3	x_3	x_3	x_1

The TSMs of Σ, Σ' are $F = \delta_3[1, 3, 3, 2, 2, 3, 1, 2, 1]$ and $F' = \delta_3[1, 1, 3, 3, 2, 3, 1, 2, 1]$, respectively. Σ, Σ' have no cycles, and $R(\Sigma) \geq R(\Sigma')$ obtained by Lemma 3.15 implies the existence of controller C . With $V = \{2, 1\}$ and $n_2 = 6$, K is given as $K = \delta_{18}[4, 3, 3, 4, 8, 6, 7, 9, 12, 10, 11, 10, 0, 0, 1, 3, 3, 2, 5, 3, 14, 5, 6, 0, 0, 0, 2, 2, 3, 14, 17, 15, 16, 18, 18, 4, 5, 1, 4, 5, 0, 0, 0, 0, 1, 2, 1, 0, 0, 0, 1, 2, 1]$ by Theorem 4.4. Then, G, H (correspondingly, f_c, h_c) can be derived from K by (13).

Before the end of this section, we briefly discuss the case when Σ has cycles. Let $\Sigma_{|\tilde{f}} = (E, \tilde{X}, \tilde{f})$ be a generalized machine induced by a generalized transition function \tilde{f} in (5). Then the matrix expression of $\Sigma_{|\tilde{f}}$ is $x_\Sigma(t+1) = \tilde{F} x_\Sigma(t)$ where \tilde{F} is the generalized TSM of Σ . Replacing $x_\Sigma(t+1) = F x_\Sigma(t)$ by the above equation, Lemma 4.1 still holds by repeating the preceding arguments. Moreover, replacing “stably reachable” by “deterministically stably reachable” in Theorem 4.4, the model matching result can be extended to the case for Σ with cycles. The procedure to design C in this case is basically the same as that of Theorem 4.4, though the fact that some columns of \tilde{F} may have more than one “1” makes it a little more complicated. The detailed proof and design procedure are omitted for space limitations.

REFERENCES

- [1] G. Barrett and S. Lafortune, “Bisimulation, the supervisory control problem and strong model matching for finite state machines,” *Discrete Event Dyn. Syst.: Theory Appl.*, vol. 8, no. 4, pp. 377–429, 1998.
- [2] M. Di Benedetto and J. Grizzle, “Asymptotic model matching for nonlinear systems,” *IEEE Trans. Autom. Control*, vol. 39, no. 8, pp. 1539–1550, Aug. 1994.
- [3] M. Di Benedetto, A. Sangiovanni-Vincentelli, and T. Villa, “Model matching for finite state machines,” *IEEE Trans. Autom. Control*, vol. 46, no. 11, pp. 1726–1743, Nov. 2001.

- [4] D. Cheng and H. Qi, "A linear representation of dynamics of Boolean networks," *IEEE Trans. Autom. Control*, vol. 55, no. 10, pp. 2251–2258, Oct. 2010.
- [5] D. Cheng, H. Qi, and Z. Li, *Analysis and Control of Boolean Networks: A Semi-Tensor Product Approach*. London, U.K.: Springer, 2010.
- [6] D. Cheng and H. Qi, *An Introduction to Semi-Tensor Product of Matrices and its Applications*. Singapore: World Scientific, 2012.
- [7] S. Eilenberg, *Automata, Languages, and Machines*. New York: Academic Press, 1976.
- [8] X. Geng and J. Hammer, "Input/output control of asynchronous sequential machines," *IEEE Trans. Autom. Control*, vol. 50, no. 12, pp. 1956–1970, Dec. 2005.
- [9] R. Grosu, "Finite automata as time-invariant linear systems observability, reachability and more," in *Hybrid Systems: Computation and Control, Lecture Notes in Computer Science*, R. Majumdar and P. Tabuada, Eds. Berlin: Springer-Verlag, 2009, vol. 5469, pp. 194–208.
- [10] Z. Kohavi, *Switching and Finite Automata Theory*. New York: McGraw-Hill, 1970.
- [11] H. Li and Y. Wang, "On reachability and controllability of switched Boolean control networks," *Automatica*, vol. 48, no. 11, pp. 2917–2922, 2012.
- [12] T. Murphy, X. Geng, and J. Hammer, "On the control of asynchronous machines with races," *IEEE Trans. Autom. Control*, vol. 48, no. 6, pp. 1073–1081, 2003.
- [13] S. Seshu, R. Miller, and G. Metzger, "Transition matrices of sequential machines," *IRE Trans. Circuit Theory*, vol. 6, no. 1, pp. 5–12, 1959.
- [14] N. Venkatraman and J. Hammer, "On the control of asynchronous sequential machines with infinite cycles," *Int. J. Control*, vol. 79, no. 7, pp. 764–785, 2006.
- [15] X. Xu and Y. Hong, "Matrix expression and reachability analysis of finite automata," *J. Control Theory Appl.*, vol. 10, no. 2, pp. 210–215, 2012.
- [16] J. M. Yang, "Corrective control of input/output asynchronous sequential machines with adversarial inputs," *IEEE Trans. Autom. Control*, vol. 55, no. 3, pp. 755–761, Mar. 2010.
- [17] J. M. Yang, "Model matching inclusion for input/state asynchronous sequential machines," *Automatica*, vol. 47, no. 3, pp. 597–602, 2011.
- [18] G. Q. Zhang, "Automata, Boolean matrices, and ultimate periodicity," *Inform. Comput.*, vol. 152, pp. 138–154, 1999.

Multi-Resolution Explicit Model Predictive Control: Delta-Model Formulation and Approximation

Farhad Bayat, *Member, IEEE*, and
Tor Arne Johansen, *Senior Member, IEEE*

Abstract—This paper deals with the explicit solution and approximation of the constrained linear finite time optimal control problem for systems with fast sampling rates. To this aim, the recently developed explicit model predictive control (eMPC) is reformulated and characterized using the δ -operator to enjoy its promising advantages compared to the time-shift operator. Using the proposed δ -model eMPC formulation, a systematic method is proposed for first designing a low-complexity approximate eMPC solution and then improving its closed loop action without first determining an exact optimal solution that might be of prohibitive complexity. It is shown that the stability and feasibility of the proposed sub-optimal solution is guaranteed.

Index Terms—Approximate explicit model predictive control, delta-operator, explicit model predictive control, multi-parametric programming, multi-resolution model predictive control.

I. INTRODUCTION

Model predictive control (MPC) has proved its ability to handle constrained optimal control problems in which mathematical models play a crucial role in the design and analysis of the control system. Often such a model is derived from physical laws resulting in continuous-time descriptions. In general it is well known that using continuous-time models gives realistic insight into the system due to the fact that the physical systems typically evolve continuously. Unfortunately, the continuous-time models cannot be used directly for implementation in digital computers. A well-known and widely used method for describing discrete-time models is the time-shift operator which is described by $x_{k+1} = qx_k$. In [1] it is shown that not only is there no intuitive connection between discrete-time and continuous-time models but also serious numerical problems arise at the high sampling rate when the shift operator is used to describe a discrete-time model. To overcome this limitation, a very simple but powerful affine mapping named the δ -operator is introduced in [1] as $q = T_s\delta + 1$. Based on the results in [1]–[3] the main properties of the δ -model are: (i) the δ -operator offers a model with almost the same degree of flexibility and simplicity as the shift operator, (ii) the δ -operator provides a more direct insight into the system, (iii) the implementation is almost as simple as the shift operator, (iv) many results with the δ -model can be seen as an approximation for continuous-time systems with approximation error of order $\mathcal{O}(T_s)$, and (v) the δ -model makes it possible to avoid non-minimum-phase sampling zeros arising in high sampling rates when using the shift operator.

It is well known that the online MPC is mainly limited to the systems with relatively low sampling rate. Recently in order to handle this limitation, the power of multi-parametric quadratic programming (mpQP)

Manuscript received June 19, 2012; revised December 01, 2012. Date of publication April 24, 2013; date of current version October 21, 2013. Recommended by Associate Editor J. Bravasky.

F. Bayat is with the Control Systems Group, Department of Engineering, Faculty of Electrical Engineering, University of Zanjan, 45371-38791, Zanjan, Iran, (e-mail: bayat.farhad@znu.ac.ir).

T. A. Johansen is with Center for Autonomous Marine Operations and Systems, Department of Engineering Cybernetics, Norwegian University of Science and Technology, N-7491 Trondheim, Norway, (e-mail: Tor.Arne.Johansen@itk.ntnu.no).

Color versions of one or more of the figures in this paper are available online at <http://ieeexplore.ieee.org>.

Digital Object Identifier 10.1109/TAC.2013.2259982