21st International Symposium on
Mathematical Theory of Networks and Systems
July 7-11, 2014. Groningen, The Netherlands

Asynchronous Sequential Machines: Static State Feedback and Model Matching

Jung–Min Yang*[1] and Jacob Hammer[2]

***Abstract*— The control of asynchronous sequential machines by static state feedback is considered with the objective of matching a prescribed model. A necessary and sufficient condition for model matching by static state feedback is derived. This condition is based on counting the number of members of certain sets; it is expressed in terms of features of a matrix derived from the given recursion functions of the controlled machine and the desired model.**

I. INTRODUCTION

Asynchronous sequential machines are dynamic automata that operate without a clock. They form the clockless logic components of high speed computer systems as well as the foundation for the mathematical modeling of signaling chains in molecular biology ([1]). In [2], an effort was initiated to develop control theoretic techniques that help overcome deficiencies in the operation of asynchronous sequential machines. These techniques are based on the use of a controller that is connected to the deficient machine to regulate, rectify, and amend its operation. They are often more efficient than replacing a deficient machine by a redesigned one. Moreover, in many cases, such as in the case of defective biological signaling chains or remote computing systems, replacement of a defective machine is not an option.

Along this line of work, control theoretic techniques were developed to eliminate the adverse effects of critical races on asynchronous machines ([2], [3] and [4]); to design input/output controllers for asynchronous machines with no state access ([5]); to overcome the effects of infinite cycles in asynchronous machines ([6]); to counteract the effects of adversarial interventions on asynchronous machines ([7] and [8]); and to design adaptive controllers for asynchronous machines that are incompletely specified ([7] and [12]). An analysis of several practical applications of controllers for asynchronous machines can be found in [10] and [11].

The studies mentioned in the previous paragraph investigate the existence and the design of dynamic controllers, namely, controllers formed by asynchronous machines that include memory elements. The present note concentrates on the existence and the design of static controllers, namely, controllers formed by logical gates with no memory elements. Specifically, we concentrate on the existence and the design of static state feedback controllers, with the main objective of deriving controllers that achieve model matching. Although the conditions under which model matching can be achieved with static state feedback controllers are somewhat more stringent than the conditions under which model matching can be achieved with dynamic feedback controllers, the use of static controllers is often preferable when possible, as static controllers are simpler to implement and may offer lower cost and higher reliability. In section IV, we derive a necessary and sufficient condition for the existence of static state feedback controllers that solve a specified model matching problem. This section also presents a design methodology for such controllers, whenever they exist.

The existence conditions and the design methodology presented in section IV are based on a matrix which, in qualitative terms, distills certain common reachability properties shared by the controlled machine and the desired model. This matrix can be derived directly from data given about the two machines. The basic operation involved in checking the existence conditions centers on counting the number of members of certain sets. A detailed example is provided in section V to demonstrate the results.

II. PRELIMINARIES

An input/state asynchronous machine Σ is represented by a triplet $\Sigma = (A, X, f)$, where A is the input set, X is the state set, and $f : X \times A \to X$ is the recursion function. The machine operates through the recursion

$$x_{k+1} = f(x_k, u_k), k = 0, 1, 2, ...$$

In general, f is a partial function defined only on a subset of $X \times A$. A pair $(x, u) \in X \times A$ at which f is defined is called a *valid pair*. A valid pair (x, u) is a *stable combination* if $f(x, u) = x$, namely, if it is a stationary point of f; otherwise, (x, u) is a *transient combination*. The machine Σ rests in a stable combination (x, u) until the input character is changed to another character, say to the character v. Upon this change, Σ may engage in a chain of transitions. Assuming that Σ has no infinite cycles, this chain of transitions ends at a stable state. The state x' of Σ at the end of the transition chain is called the *next stable state* of the pair (x, v). It defines the *stable recursion function* $s : X \times A \to X$ of Σ through the relation $s(x, v) := x'$, namely, the stable recursion function provides the next stable state of a state/input pair. Using

*The work of J.-M. Yang was supported in part by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIP) (No. NRF–2012R1A2A2A01003419), and in part by Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education (No. NRF–2010–0007271).

[1]Jung–Min Yang is with the School of Electronics Engineering, Kyungpook National University, 1370 Sankyuk-dong, Buk-gu, Daegu, 702-701, Republic of Korea. jmyang@ee.knu.ac.kr

[2]Jacob Hammer is with the Department of Electrical and Computer Engineering, University of Florida, Gainesville, FL 32611, USA hammer@mst.ufl.edu

ISBN: 978-90-367-6321-9

s as the recursion function yields the *stable state machine* $\Sigma_{|s} = (A, X, s)$ induced by Σ.

Transitions through transient combinations of an asynchronous machine are extremely quick, occurring ideally in zero time. As a result, users observing an asynchronous machine Σ see, in fact, the stable state machine $\Sigma_{|s}$.

As mentioned, our discussion concentrates on static state feedback. A static state feedback configuration is depicted in Figure 1; the controller is represented by a function $\varphi : X \times A \to A$ that generates the current input character u_k of Σ from the current state x_k of Σ and the current external input character v_k according to

$$u_k = \varphi(x_k, v_k), k = 0, 1, 2, \ldots$$

The configuration generates the closed loop machine Σ_φ given by

$$\Sigma_\varphi : x_{k+1} = f(x_k, \varphi(x_k, v_k)), k = 0, 1, \ldots$$

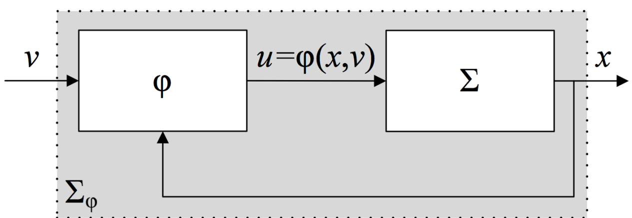

Fig. 1. Static state feedback.

Our objective is to investigate the model matching problem. Given a stable-state machine model $\Sigma' = (A, X, s')$, we seek a static state feedback function $\varphi : X \times A \to A$ for which

$$\Sigma_\varphi = \Sigma'. \tag{1}$$

The equality (1) refers to the stable state machines induced by Σ_φ and Σ' since, as indicated earlier, users are aware only of stable transitions. In section IV, we provide a necessary and sufficient condition for the existence of such feedback functions as well as a process of their construction, whenever they exist.

Due to the speed of transients and the lack of synchrony, asynchronous machines are usually operated in *fundamental mode* (e.g., [13]), where changes to the input are allowed only when the machine is in a stable combination. This prevents an unpredictable response that may arise when the input change occurs at an unpredictable stage of a chain of transients. Thus, a string $v_1 v_2 \cdots v_k$ of input characters must be applied to Σ in a step-by-step manner, where the next input character is applied only after the machine has reached a stable state. Then, the final stable state reached is $s(x, v_1 v_2 \cdots v_k) := s(s(s(x, v_1), v_2) \cdots v_k)$.

To apply the notion of fundamental mode operation to a static state feedback configuration around the machine Σ, consider a string of states $x_1 x_2 \cdots x_m, m > 2$, that form a chain of transient transitions from a stable combination with x_1 to a stable combination with x_m. If the feedback function φ changes value at an intermediate state in this transition chain, say, at x_2, we are faced with the uncertainty of whether the change in φ occurs before or after the transition of Σ to x_3. This creates a potential uncertainty and violates fundamental mode operation. Consequently, φ must maintain the same value at all intermediate states in a transition chain; it can change value only upon reaching a stable state, as follows.

Proposition 1: Let $\Sigma = (A, X, f)$ be an input/state asynchronous machine and let $\varphi : X \times A \to A$ be a static feedback function as shown in Figure 1. Then, the following two statements are equivalent.

(i) The closed-loop system Σ_φ operates in fundamental mode.

(ii) At every valid pair (x, v) of Σ_φ, the function φ satisfies $\varphi(x, v) = \varphi(f(x, \varphi(x, v)), v)$ whenever the next step $(f(x, \varphi(x, v)), \varphi(x, v))$ is a transient combination of Σ. □

When the next step $(f(x, \varphi(x, v)), \varphi(x, v))$ forms a stable combination, there is no restriction on the value $\varphi(f(x, \varphi(x, v)), v)$. On the other hand, when $(f(x, \varphi(x, v)), \varphi(x, v))$ is a transient combination, then φ must have the same value at the two pairs (x, v) and $(f(x, \varphi(x, v)), v)$; otherwise, the control input u of Σ will change during a transient, violating fundamental mode operation. Proposition 1*(ii)* restricts the class of feedback functions φ that can be used for Σ.

III. Reachability Matrices

Let $\Sigma = (A, X, f)$ be an asynchronous machine with the stable recursion function s and, for a set A, denote by A^+ the set of all strings of members of A. A state x' is *stably reachable* from a state x if there is an input string $t \in A^+$ such that $x' = s(x, t)$ ([2]). Stable reachability from x to x' is necessary and sufficient for the existence of a dynamic feedback controller that steers Σ from x to x' in fundamental mode operation ([2], [5], [7]). However, the situation with static feedback is different, as one must align with the requirement of Proposition 1; here, in addition to the endpoints of a stable transition, the transient states encountered along the transition must also be considered. The feedback function φ must maintain the same value on all transient states of a one-step stable transition.

Let $(x, v) \in X \times A$ be a valid pair of the machine $\Sigma = (A, X, f)$, and let $x_1 = f(x, v), x_2 = f(x_1, v), \cdots, x_i = f(x_i, v)$ be the chain of transitions from the pair (x, v) to the next stable state $x_i = s(x, v)$, $i \geq 1$. Let N be a character not in A or X. Using the foregoing notation, define a function $\tau : X \times A \to (X \times A)^+ \cup N : (x, v) \mapsto \tau(x, v)$ that generates the string of all state/input pairs traversed by Σ as it undergoes the stable transition from (x, v) to $s(x, v)$:

$$\tau(x, v) = \begin{cases} (x, v)(x_1, v) \cdots (x_i, v) & \text{if } (x, v) \text{ is valid,} \\ N & \text{otherwise.} \end{cases} \tag{2}$$

We refer to τ as the *transition chain function* of Σ.

It is sometimes convenient to split the chain of pairs of (2) into a set of pairs by using the *splitting operator* Δ defined for a string of pairs $a = (x,v)(x_1,v)\cdots(x_i,v), i \geq 1$, by

$$\Delta(a) := \begin{cases} \{(x,v),(x_1,v),\cdots,(x_i,v)\} & \text{if } i \geq 1; \\ \{(x,v)\} & \text{if } a = (x,v); \\ \varnothing & \text{if } a = N. \end{cases}$$

In addition, we define the operator Δ^- that truncates the last pair from the outcome of Δ:

$$\Delta^-(a) := \begin{cases} \{(x,v),(x_1,v),\cdots,(x_{i-1},v)\} & \text{if } i > 1; \\ \{(x,v)\} & \text{if } i = 1; \\ \varnothing \quad \text{if } a = (x,v) & \text{or } a = N. \end{cases}$$

Next, recalling the stable recursion function s of Σ, let $s[x \times A]$ be the set of all next stable states of x, namely,

$$s[x \times A] := \{x' \in X : x' = s(x,a) \text{ for a character } a \in A\}.$$

Definition 1: Let Σ be an asynchronous machine with state set $X = \{x^1,\ldots,x^n\}$ and transition chain function τ. The *one-step chain reachability matrix* $\rho^+(\Sigma)$ is an $n \times n$ matrix whose (i,j) entry, $i,j = 1,2,...,n$, is

$$\rho^+_{ij}(\Sigma) = \begin{cases} \{\Delta(\tau(x^i,v)) : v \in A \text{ and } s(x^i,v) = x^j\} \\ \qquad \text{if } x^j \in s[x^i \times A]; \\ N \qquad \text{otherwise.} \end{cases}$$

Similarly, the *truncated one-step chain reachability matrix* $\rho^-(\Sigma)$ is

$$\rho^-_{ij}(\Sigma) = \begin{cases} \{\Delta^-(\tau(x^i,v)) : v \in A \text{ and } s(x^i,v) = x^j\} \\ \qquad \text{if } x^j \in s[x^i \times A]; \\ N \qquad \text{otherwise.} \end{cases} \quad \square$$

As we can see, $\rho^+_{ij}(\Sigma)$ consists of sets of pairs; each such set of pairs includes all state/input pairs encountered in a one-step stable transition from x^i to x^j. The entry $\rho^-_{ij}(\Sigma)$ is obtained from $\rho^+_{ij}(\Sigma)$ by removing the last stable combination from each member. Clearly, for $i \neq j$, we have $\rho^+_{ij}(\Sigma) \neq N$ and $\rho^-_{ij}(\Sigma) \neq N$ if and only if there is a one-step stable transition from x^i to x^j.

We turn now to the question of when stable transitions can be implemented by a static state feedback controller. Denote by $\Pi_X : X \times A \to X : \Pi_X(x,a) \mapsto x$ the projection that extracts the state x from a state/input pair (x,a); and by $\Pi_A : X \times A \to A : \Pi_A(x,a) \mapsto a$ the projection that extracts the input character a from (x,a). We expand the domain of Π_X and Π_A to $(X \times A) \cup N$ by setting $\Pi_X N = \Pi_A N = \varnothing$.

Definition 2: Let $S \subseteq (X \times A) \cup N$ be a set that may include state/input pairs and the character N. Then, S is an *implementable set* if the following is true for all members $\alpha, \alpha' \in S$: if $\Pi_A \alpha \neq \Pi_A \alpha'$, then also $\Pi_X \alpha \neq \Pi_X \alpha'$. $\square$
When $S \subseteq (X \times A) \cup N$ is an implementable set, we can use the members of S to define a function $\phi : X \to A$ by setting $\phi(x) := a$ for every pair $(x,a) \in S$. On the other hand, when S is not an implementable set, such a function does not exist, since then a single member of X is assigned to two or more different characters of A. This proves the following.

Proposition 2: Let $S \subseteq (X \times A) \cup N$ be a nonempty set. Then, there is a function $\phi : X \to A$ such that $(x,\phi(x)) \in S$ for all $x \in \Pi_X S$ if and only if S is an implementable set. $\square$
Implementable sets play a critical role in our discussion; they form the foundation for characterizing the existence of static feedback functions. A slight reflection shows that the following is true, where $\#S$ denotes the cardinality of a set S and '$\setminus$' denotes set difference.

Proposition 3: The union $S \cup S'$ of two implementable sets is an implementable set if and only if $\#[(S \cap S') \setminus N] = \#(\Pi_X S \cap \Pi_X S')$. $\square$
The condition of Proposition 3 leads us to a new operation between compatible sets of pairs, which combines them whenever the outcome is an implementable set; otherwise, it generates the character N.

Definition 3: The *compatible union* $S \sqcup S'$ of two sets $S, S' \subseteq (X \times A) \cup N$ is given by

$$S \sqcup S' := \begin{cases} N & \text{if } S = N; \\ N & \text{if } S' = N; \\ N & \text{if } \#[(S \cap S') \setminus N] \neq \#(\Pi_X S \cap \Pi_X S'); \\ S \cup S' & \text{otherwise.} \end{cases} \quad \square$$

The following operation turns two sets $S_1, S_2 \subseteq (X \times A) \cup N$ into a (non-ordered) list of two members:

$$S_1 \oplus S_2 := \begin{cases} \{S_1, S_2\} & \text{if } S_1 \neq N \text{ and } S_2 \neq N; \\ S_1 & \text{if } S_2 = N; \\ S_2 & \text{if } S_1 = N. \end{cases}$$

The compatible union of two lists $\{S_1, S_2, ..., S_m\}$ and $\{S'_1, S'_2, ..., S'_{m'}\}$ of subsets of $(X \times A) \cup N$ is the list of compatible unions of all combinations of two members, one from each list:

$$\{S_1, S_2, ..., S_m\} \sqcup \{S'_1, S'_2, ..., S'_{m'}\} := \bigoplus_{\substack{i=1,\ldots,m, \\ j=1,\ldots,m'}} S_i \sqcup S'_j.$$

For lists of implementable sets, every member of $\{S_1, S_2, ..., S_m\} \sqcup \{S'_1, S'_2, ..., S'_{m'}\}$ is an implementable set, unless the result of the entire operation is N.

Using the foregoing operations, we can define an operation akin to matrix multiplication among matrices whose entries are lists of implementable sets or N. Let B and C be two such $n \times n$ matrices; the *product* BC is also an $n \times n$ matrix whose (i,j) entry, $i,j \in \{1,2,...,n\}$, is

$$(BC)_{ij} := \left\{\{B_{i1} \sqcup C_{1j}\} \oplus \cdots \oplus \{B_{in} \sqcup C_{nj}\}\right\}. \tag{3}$$

Using the matrix product, we can define "powers" of the one-step chain reachability matrix by setting

$$\begin{aligned} \rho^p(\Sigma) &:= \rho^+(\Sigma) \text{ for } p = 1, \\ \rho^p(\Sigma) &:= (\rho^-(\Sigma))^{p-1}\rho^+(\Sigma) \text{ for } p \geq 2. \end{aligned} \tag{4}$$

For notational purposes, it is convenient to define the zero power of $\rho(\Sigma)$ as a matrix that includes all stable combinations of Σ. Let $\sigma_i \subseteq \{x^i\} \times A$ be the set of all stable combinations of the state x^i. Then, the $n\times n$ matrix $\rho^0(\Sigma)$ has the entries:

$$\rho^0_{ij}(\Sigma) := \begin{cases}\sigma_i & \text{if } j=i,\\ N & \text{else,}\end{cases} \; i,j\in\{1,\dots,n\}.$$

Next, the *sum* $B\oplus C$ of the two $n\times n$ matrices B,C is akin to matrix addition and yields an $n\times n$ matrix whose entries are lists of implementable sets or N:

$$(B\oplus C)_{ij} := B_{ij}\oplus C_{ij}, \;\; i,j\in\{1,\dots,n\}.$$

Definition 4: Let Σ be an asynchronous machine with n states. The *chain reachability matrix* $\rho(\Sigma)$ is

$$\rho(\Sigma) := \bigoplus_{p=0,1,\dots,n-1} \rho^p(\Sigma),$$

where $\rho^p(\Sigma)$ is given by (4). □

Note that, for a machine with n states $x^1,x^2,...,x^n$, the sum in Definition 4 goes only up to step $n-1$. The next statement points out the significance of the chain reachability matrix: the entry $\rho_{ij}(\Sigma)$ includes all stable transitions from x^i to x^j that can be implemented by a static state feedback controller.

Theorem 1: Let Σ be an asynchronous machine with state set $X=\{x^1,\dots,x^n\}$. Then, the following two statements are equivalent.

(i) $\rho_{ij}(\Sigma)\neq N$.
(ii) There is a state feedback function $\phi : X\to A$ for which Σ_ϕ has a stable transition from x^i to x^j in fundamental mode operation.

Proof: (Sketch) First, assume that *(i)* is valid. Then, $\rho_{ij}(\Sigma)$ includes a nonempty implementable set $S\subseteq X\times A$, say $S\in\rho^p_{ij}(\Sigma)$ for some $0\le p\le n-1$. By Proposition 2, there is a partial function $\phi : X\to A$ that satisfies

$$S=\{(x,\phi(x)) : x\in\Pi_X S\}.$$

If $p=0$ or 1, then ϕ induces a stable combination or a one-step stable transition from x^i to x^j. Further, consider the case where $2\le p\le n-1$. Then, in view of (3)), the set S originates from a string of p consecutive one-step stable transition chains of Σ, say the transitions

$$\begin{gathered}(x_{1,1},v_1),(x_{1,2},v_1),\cdots,(x_{1,k_1-1},v_1),\\ (x_{2,1},v_2),(x_{2,2},v_2),\cdots,(x_{1,k_1-1},v_2),\\ \vdots\\ (x_{p-1,1},v_{p-1}),(x_{p-1,2},v_{p-1}),\cdots,(x_{p-1,k_p-1},v_{p-1}),\\ (x_{p,1},v_p),(x_{p,2},v_p),\cdots,(x_{p,k_p},v_p),\end{gathered} \tag{5}$$

where v_ℓ is the (constant) input character of the $\ell-$th one step stable transition in this string of transitions, and k_ℓ is the number of states traversed by Σ in this step. The last step, i.e., $\ell=p$, ends at the stable combination reached at the target state x^j. By (5), the function ϕ is given by

$$\begin{aligned}&\phi(x_{a,b})=v_a, b=1,2,...,(k_a-1), a=1,2,...,p,\\ &\phi(x_{p,k_p})=v_p.\end{aligned}$$

Note that, for $a\in\{1,\dots,p-1\}$, the function ϕ changes its value from v_a to v_{a+1} at the state $x_{a,k_a}=x_{a+1,1}$; this change, which occurs at a stable state, starts the next one-step stable transition in our chain. As this change occurs at a stable state of Σ, fundamental mode operation is preserved by Proposition 1. At the last state x_p, the function ϕ maintains the value v_p, thus keeping Σ in the stable combination (x_{p,k_p},v_p). In this way, the closed-loop system Σ_ϕ moves through the entire string of consecutive transitions (5), going from the state $x_{1,1}$ to the state x_{p,k_p} and resting at the last stable combination (x_{p,k_p},v_p). Hence, ϕ induces the required stable transition without violating fundamental mode operation, proving that *(i)* implies *(ii)*.

Conversely, assume that *(ii)* is valid. Then, there must be a string of stable transitions that takes Σ from the state x^i to the state x^j. Using an argument analogous to the one used in [2], it follows that, if x^j is stably reachable from x^i, it can be so reached in $n-1$ or fewer stable one-step transitions (see [9] for details). As this string of stable transitions is implemented by a state feedback function ϕ, it follows by Proposition 2 that the state/input pairs encountered during the transition must form an implementable set. These facts lead to the conclusion that *(ii)* implies *(i)*. ■

IV. MODEL MATCHING BY STATIC STATE FEEDBACK

Given an asynchronous machine $\Sigma=(A,X,f)$ with state set $X=\{x^1,\dots,x^n\}$, let $\Sigma'=(A,X,s')$ be a stable state machine having the same input set and state set as Σ and serving as a model. Our objective is to find a state feedback function $\varphi : X\times A\to A$ for which $\Sigma_\varphi=\Sigma'$, where the equality refers to the stable state machine induced by Σ_φ.

Recall from [2] the one-step matrix of stable transitions $R^1(\Sigma')$ — an $n\times n$ matrix that describes all one-step stable transitions of the model Σ'. Specifically, denoting by $c_{ij} := \{v\in A : s(x^i,v)=x^j\}$ the set of all input characters that take Σ' from the state x^i to the state x^j in a one step stable transition, we have

$$R^1_{ij}(\Sigma) = \begin{cases} c_{ij} & \text{if } c_{ij}\neq\varnothing,\\ N & c_{ij}=\varnothing,\end{cases} \; i,j=1,...,n. \tag{6}$$

To examine the conditions under which Σ can match the model Σ' via a static state feedback controller, consider the case where an input character $v\in A$ induces in Σ' two unrelated one-step stable transitions: one x^i to x^j, and another one from $x^{i'}$ to $x^{j'}$, so that $v\in R^1_{ij}(\Sigma')\cap R^1_{i'j'}(\Sigma')$. Now, assume that there is a static state feedback function $\varphi : X\times A\to A$ for which the closed-loop system Σ_φ emulates the stable transitions of the model Σ', namely, that $\Sigma_\varphi=\Sigma'$. Let $\rho(\Sigma)$ be the chain reachability matrix of Σ. Then, the external input character v must induce a stable transition from x^i to x^j as well as a stable transition from $x^{i'}$ to $x^{j'}$ in Σ_φ (refer to Figure 1). As these two transitions have the same external input character v, the feedback function φ depends for these two transitions only on the state of Σ. In view of Proposition 2, this is possible if and only if the union $\rho_{ij}(\Sigma)\cup\rho_{i'j'}(\Sigma)$ contains an implementable set, or, by Definition 3, if and only if $\rho_{ij}(\Sigma)\sqcup\rho_{i'j'}(\Sigma)\neq N$. The same argument holds for all one-step stable transitions of Σ' that are induced by the external input character v. This fact leads to a necessary and sufficient condition under which model matching by static state feedback is possible. To this end, define the following set of pairs for each character $v \in A$:

$$\mathscr{F}(v|\Sigma') = \left\{(i,j) \in \{1,...,n\} \times \{1,...,n\} : v \in R^1_{ij}(\Sigma')\right\} \quad (7)$$

Theorem 2: Let $\Sigma = (A,X,f)$ and $\Sigma' = (A,X,s')$ be asynchronous machines, where Σ' is a stable state machine. Let $\rho(\Sigma)$ be the chain reachability matrix of Σ, let $R^1(\Sigma')$ be the one-step matrix of stable transitions of Σ', and, for an input character $v \in A$, let $\mathscr{F}(v|\Sigma')$ be given by (7). Then, the following two statements are equivalent.

(i) There is a static state feedback function $\varphi : X \times A \to A$ for which $\Sigma_\varphi = \Sigma'$.

(ii) $\sqcup_{(i,j)\in\mathscr{F}(v|\Sigma')}\rho_{ij}(\Sigma) \neq N$ for all $v \in A$.

Proof: The fact that *(i)* implies *(ii)* follows from the discussion preceding the theorem. Conversely, assume that *(ii)* is valid. For a character $v \in A$, denote

$$\sigma(\Sigma|\Sigma',v) := \bigsqcup_{(i,j)\in\mathscr{F}(v|\Sigma')} \rho_{ij}(\Sigma).$$

If $\sigma(\Sigma|\Sigma',v) \neq N$, then, by Theorem 1, there is a feedback function $\phi_v : X \to A$ for which Σ_{ϕ_v} implements all one-step stable transitions of Σ' that are induced by the input character v. Now, for all pairs $(x,v) \in X \times A$ for which $\mathscr{F}(v|\Sigma') \neq \varnothing$, combine the functions ϕ_v, $v \in A$, into one partial function $\varphi : X \times A \to X : \varphi(x,v) := \phi_v(x)$. Then, Σ_φ simulates all one-step stable transitions of the model Σ'. As a result, Σ_φ also implements any succession of stable transitions of Σ', so that $\Sigma_\varphi = \Sigma'$, and it follows that *(ii)* implies *(i)*. ■

Note that, in view of Proposition 3, condition *(ii)* of Theorem 2 just involves counting the number of members of the appropriate sets.

A few special cases of Theorem 2 are of interest. First, clearly, if $\mathscr{F}(v|\Sigma') = \varnothing$, then v is not a permissible input character of the model Σ', and hence v will never be applied to the closed-loop system Σ_φ. As a result, there is no need to define the feedback function φ on any of the pairs $(x,v), x \in X$, as these pairs will not be used. Similarly, if there is an integer $i \in \{1,\dots,n\}$ for which $(i,j) \notin \mathscr{F}(v|\Sigma')$ for all integers $j \in \{1,\dots,n\}$, then, the pair (x^i,v) is not used by the model Σ', and hence it is not used by Σ_φ. Consequently, φ does not need to be defined on the pair (x^i,v).

V. EXAMPLE

Consider the asynchronous machine $\Sigma = (A,X,f)$, where $A = \{a,b,c\}$, $X = \{x^1,x^2,x^3\}$, and the recursion function f is described by the state flow diagram of Figure 2.

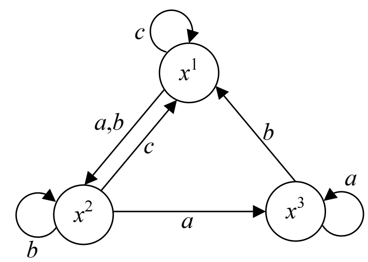

Fig. 2. State flow diagram of Σ.

The stable recursion function s of Σ can be derived from Figure 2, and is given in Table I.

TABLE I

STABLE RECURSION FUNCTION s OF Σ.

	a	b	c
x^1	x^3	x^2	x^1
x^2	x^3	x^2	x^1
x^3	x^3	x^2	-

To find the chain reachability matrix $\rho(\Sigma)$, we use Table I to find $\rho^0(\Sigma)$, $\rho^+(\Sigma)$, and $\rho^-(\Sigma)$:

$$\rho^0(\Sigma) = \begin{bmatrix} \{(x^1,c)\} & N & N \\ N & \{(x^2,b)\} & N \\ N & N & \{(x^3,a)\} \end{bmatrix};$$

$$\rho^+(\Sigma) = \begin{bmatrix} \{(x^1,c)\} & \{(x^1,b),(x^2,b)\} & \rho^+_{13}(\Sigma) \\ \{(x^2,c),(x^1,c)\} & \{(x^2,b)\} & \{(x^2,a),(x^3,a)\} \\ N & \rho^+_{32}(\Sigma) & \{(x^3,a)\} \end{bmatrix},$$

where

$$\rho^+_{13}(\Sigma) = \{(x^1,a),(x^2,a),(x^3,a)\},$$
$$\rho^+_{32}(\Sigma) = \{(x^3,b),(x^1,b),(x^2,b)\};$$

$$\rho^-(\Sigma) = \begin{bmatrix} N & \{(x^1,b)\} & \{(x^1,a),(x^2,a)\} \\ \{(x^2,c)\} & N & \{(x^2,a)\} \\ N & \{(x^3,b),(x^1,b)\} & N \end{bmatrix}.$$

As the cardinality of X is $n = 3$, it follows by Definition 4 that

$$\begin{aligned} \rho(\Sigma) &= \rho^0(\Sigma) \oplus \rho^1(\Sigma) \oplus \rho^2(\Sigma) \\ &= \rho^0(\Sigma) \oplus \rho^+(\Sigma) \oplus [\rho^-(\Sigma)\rho^+(\Sigma)] \\ &= \begin{bmatrix} \{(x^1,c)\} & \{(x^1,b),(x^2,b)\} & \rho_{13}(\Sigma) \\ \{(x^2,c),(x^1,c)\} & \{(x^2,b)\} & \rho_{23}(\Sigma) \\ N & \rho_{32}(\Sigma) & \{(x^3,a)\} \end{bmatrix}, \end{aligned}$$

where

$$\rho_{13}(\Sigma) = \{\{(x^1,b),(x^2,a),(x^3,a)\},\{(x^1,a),(x^2,a),(x^3,a)\}\},$$
$$\rho_{23}(\Sigma) = \{(x^2,a),(x^3,a)\},$$
$$\rho_{32}(\Sigma) = \{(x^3,b),(x^1,b),(x^2,b)\}.$$

Consider now the model $\Sigma'_1 = (A,X,s'_1)$ of Figure 3. According to (6), we have

$$R^1(\Sigma'_1) = \begin{bmatrix} \{c\} & \{b\} & \{a\} \\ \{c\} & \{b\} & \{a\} \\ \{c\} & \{b\} & \{a\} \end{bmatrix}.$$

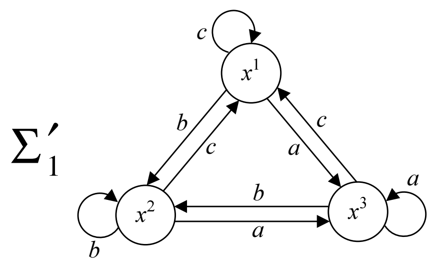

Fig. 3. The model Σ_1'.

To check whether Σ can match the model Σ_1' via static state feedback, we derive $\mathscr{F}(v|\Sigma_1')$ of (7) for every character of the input set A:

$$\mathscr{F}(a|\Sigma_1') = \{(1,3),(2,3),(3,3)\},$$
$$\mathscr{F}(b|\Sigma_1') = \{(1,2),(2,2),(3,2)\},$$
$$\mathscr{F}(c|\Sigma_1') = \{(1,1),(2,1),(3,1)\}.$$

Checking condition *(ii)* of Theorem 2 for the input character c, Definition 3 yields

$$\begin{aligned}\sqcup_{(i,j)\in\mathscr{F}(c|\Sigma_1')}\rho_{ij}(\Sigma) &= \rho_{11}(\Sigma)\sqcup\rho_{21}(\Sigma)\sqcup\rho_{31}(\Sigma)\\ &= \rho_{11}(\Sigma)\sqcup\rho_{21}(\Sigma)\sqcup N\\ &= N.\end{aligned}$$

Thus, by Theorem 2, the machine Σ cannot match the model Σ_1' by static state feedback.

Next, let's examine a different model – the model $\Sigma_2' = (A, X, s_2')$ of Figure 4.

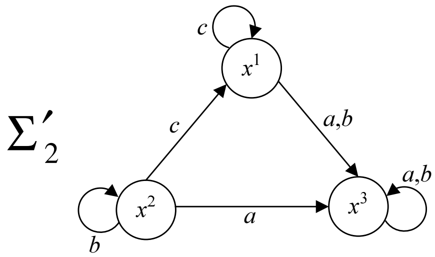

Fig. 4. The model Σ_2'.

From Figure 4, we obtain

$$R^1(\Sigma_2') = \begin{bmatrix} \{c\} & N & \{a,b\} \\ \{c\} & \{b\} & \{a\} \\ N & N & \{a,b\} \end{bmatrix}.$$

Here, a calculation similar to the one done for Σ_1' above shows that $\sqcup_{(i,j)\in\mathscr{F}(a|\Sigma_2')}\rho_{ij}(\Sigma) \neq N$, $\sqcup_{(i,j)\in\mathscr{F}(b|\Sigma_2')}\rho_{ij}(\Sigma) \neq N$, and $\sqcup_{(i,j)\in\mathscr{F}(c|\Sigma_2')}\rho_{ij}(\Sigma) \neq N$. Consequently, by Theorem 2, the machine Σ can match the model Σ_2' via static state feedback control.

A static state feedback function φ that achieves the model matching $\Sigma_\varphi = \Sigma_2'$ can be derived by following the proof of Theorem 1. For instance, for the input character c, we have $\mathscr{F}(c|\Sigma_2') = \{(1,1),(2,1)\}$; a direct calculation shows that $\rho_{11}(\Sigma)\sqcup\rho_{21}(\Sigma) = \{(x^1,c),(x^2,c)\}$. Hence, we assign $\varphi(x^1,c) = c$ and $\varphi(x^2,c) = c$. The construction of φ for the remaining input characters a and b is similar.

VI. SUMMARY

We presented a methodology for the derivation of static state feedback controllers that achieve model matching for asynchronous sequential machines. The methodology includes a necessary and sufficient condition for the existence of such controllers as well as a procedure for their construction. The necessary and sufficient condition is relatively simple, revolving around the counting of members of certain sets. When feedback functions exists, they can be constructed by following the proof of Theorem 1.

Static state feedback controllers are the simplest and fastest controllers, as they are implemented by logical gates and require no memory elements.

REFERENCES

[1] J. HAMMER [1994], "On some control problems in molecular biology," in *Proc. IEEE conf. Decision and Control*, pp. 4098–4103, December 1994.
[2] T. E. 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.
[3] J. Peng and J. Hammer, "Input/output control of asynchronous sequential machines with races," *Int. J. Control*, vol. 83, no. 1, pp. 125–144, 2010.
[4] J. Peng and J. Hammer, "Bursts and output feedback control of nondeterministic asynchronous sequential machines," *Eur. J. Control*, vol. 18, no. 3, pp. 286–300, 2012.
[5] X. Geng and J. Hammer, "Input/output control of asynchronous sequential machines," *IEEE Trans. Autom. Control*, vol. 50, no. 12, pp. 1956–1970, 2005.
[6] 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.
[7] J.–M. Yang and J. Hammer, "State feedback control of asynchronous sequential machines with adversarial inputs," *Int. J. Control*, vol. 81, no. 12, pp. 1910–1929, 2008.
[8] J.–M. Yang and J. Hammer, "Asynchronous sequential machines with adversarial intervention: the use of bursts," *Int. J. Control*, vol. 83, no. 5, pp. 956–969, 2010.
[9] J.–M. Yang and J. Hammer, "Static State Feedback Control of Asynchronous Sequential Machines", submitted for publication.
[10] J.–M. Yang and S. W. Kwak, "Realizing fault-tolerant asynchronous sequential machines using corrective control," *IEEE Trans. Control Syst. Technol.*, vol. 18, no. 6, pp. 1457–1463, 2010.
[11] J.–M. Yang and S. W. Kwak, "Fault diagnosis and fault-tolerant control of input/output asynchronous sequential machines," *IET Control Theory Appl.*, vol. 6, no. 11, pp. 1682–1689, 2012.
[12] J.–M. Yang, T. Xing, and J. Hammer, "Adaptive control of asynchronous sequential machines with state feedback," *Eur. J. Control*, vol. 18, no. 6, pp. 503–527, 2012.
[13] Z. Kohavi, *Switching and Finite Automata Theory*, 2nd ed. New York: McGraw-Hill, 1978.