Proceedings of the International Federation of Automatic Control
Symposium on Nonlinear Control Systems, Bologna, Italy,
September 2010

Output Feedback and Bursts: Overcoming Uncertainty in Asynchronous Sequential Machines

June Peng and Jacob Hammer

Department of Electrical and Computer Engineering, University of Florida, Gainesville, FL 32611; e-mail: hammer@mst.ufl.edu.

Abstract: The conditions under which output feedback controllers can overcome uncertainty in asynchronous sequential machines are broadened by the utilization of bursts - outbursts produced by asynchronous sequential machines during transitions. This note presents a framework for the design of output feedback controllers that utilize bursts to control asynchronous sequential machines with critical races. Necessary and sufficient conditions for the existence of such controllers are presented in terms of a numerical matrix derived from available data. The framework is based on a new class of generalized realizations introduced here.

1. INTRODUCTION

Asynchronous sequential machines play important roles in a diverse array of application areas, including high-speed digital computing systems, modeling of parallel computing environments, and mathematical representation of signaling chains in molecular biology ([5]). Defects and malfunctions may cause asynchronous sequential machines to become non-deterministic (e.g., [18]). In the present note, we discuss the application of feedback control to restore deterministic behavior to non-deterministic asynchronous sequential machines.

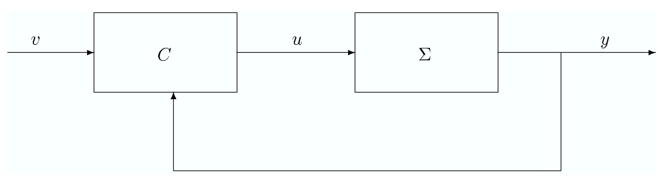

Fig. 1. The Control Configuration

The basic control configuration is described in Figure 1. Here, an asynchronous machine Σ is controlled by another asynchronous machine - the output feedback controller C. We denote the closed loop machine by Σ_c. The objective is to achieve *model matching*, namely, to find a controller C for which Σ_c emulates a specified deterministic model Σ'. If it exists, the controller C eliminates the effects of uncertainties present in Σ. This note presents necessary and sufficient conditions for the existence of such a controller C.

1.1 Basic Features

An asynchronous machine has *stable states* - states at which the machine rests until its input changes; and *transient states* - states the machine traverses quickly (ideally, in zero time) on its way from one stable state to another. Users are mindful only of stable states, as the machine does not linger at transient states.

During a transition from one stable state to another, an asynchronous machine may generate a burst - a quick succession of output characters representing the transient states passed along the way. A burst occurs (ideally) in zero time; nevertheless, a controller equipped with a sequential memory register can record the burst and use it to calculate controller action.

In [13] and [14], output feedback control of asynchronous sequential machines with critical races was examined without the utilization of bursts. The present note incorporates bursts into controller design, yielding more powerful controllers (see also [15]).

To guaranty proper operation, all asynchronous machines in this note are operated in *fundamental mode*, where changes at a machine's input is allowed only when the machine is in a stable state (e.g., [10]). This avoids input changes at unpredictable transient states during transitions. To assure fundamental mode operation of the configuration of Figure 1, we must abide by the following.

Condition 1. The configuration of Figure 1 operates in fundamental mode when the following are all valid:
(i) The machine Σ is in a stable state while the controller C is in transition;
(ii) The controller C is in a stable state while the machine Σ is in transition;
(iii) The external input v is constant while Σ or C are in transition. □

Our discussion in this note is within the framework of [11], [3], [4], [19], [20], [21], [13], [14], [15], and [22], where various aspects of the control of asynchronous sequential machines are considered. Additional studies on the control of sequential machines can be found in [16], [17], [5], [6], [7], [8],[9], [2], [1], and many others. The latter studies do not consider specialized issues related to the operation of asynchronous sequential machines, such as the distinction between stable and transient states and fundamental mode operation.

The material is organized as follows. Section 2 introduces basic concepts, and section 3 examines observers that utilize bursts. These observers are used in section 4 to develop model matching controllers.

2. BURSTS AND DETECTABILITY

2.1 Preliminaries

An asynchronous sequential machine $\Sigma = (A,Y,X,x_0,f,h)$ is characterized by its input alphabet A, its output alphabet Y, its state set X, its initial state x_0, its *recursion function* $f : X \times A \to X$, and its *output function* $h : X \to Y$. Denote by A^* the set of all strings of characters of A and by A^+ the set of all such non-empty strings. A *strict prefix* of a string $a_1a_2...a_m \in A^+$ is a substring $a_1a_2...a_q$ with $1 \le q < m$.

In response to an input sequence $u_0u_1... \in A^+$, the machine Σ generates a sequence of states $x_0x_1... \in X^+$ and an output sequence $y_0y_1... \in Y^+$ according to the recursion

$$\Sigma: \quad \begin{array}{l} x_{k+1} = f(x_k,u_k), \\ y_k = h(x_k), k = 0,1,2,... \end{array} \tag{1}$$

Here, k is a *step counter* that advances by one after a change of the input or of the state.

A pair $(x,u) \in X \times A$ at which the function f is defined is a *valid pair*; (x,u) is a *stable combination* if $x = f(x,u)$, i.e., if the machine Σ lingers at x until the input is changed. When (x,u) is not a stable combination, it creates a chain of transitions

$$x_1 = f(x,u), x_2 = f(x_1,u), ... \tag{2}$$

which may or may not terminate. If the chain terminates, then there is a state $x_i \in X$ such that $x_i = f(x_i,u)$ and (x_i,u) forms a stable combination; then, x_i is the *next stable state* of x with the input u. If the chain (2) does not terminate, it forms an *infinite cycle*. The present note concentrates on machines with no infinite cycles, so there is always a next stable state.

As indicated earlier, from a user's perspective, the behavior of an asynchronous machine is determined by its stable states. To characterize the stable state behavior, define the *stable recursion function* s of Σ by $s(x,u) := x'$, where x' is the next stable state of (x,u). This yields the *stable state machine* $\Sigma_{|s} := (A,Y,X,x_0,s,h)$.

Two asynchronous machines Σ and Σ' are *stably equivalent* if the stable state machines associated with them are equal, i.e., if $\Sigma_{|s} = \Sigma'_{|s}$. As discussed earlier, stably equivalent machines are indistinguishable by a user. Consequently, we write $\Sigma = \Sigma'$ when Σ and Σ' are stably equivalent. Our discussion revolves around

Problem 2. Model Matching: Given two asynchronous machines Σ and Σ', with Σ' serving as a model, find necessary and sufficient conditions for the existence of a controller C for which $\Sigma_c = \Sigma'$. □

Considering non-deterministic behavior, a *critical race* $(r,v) \in X \times A$ is a pair whose next stable state can be one of several states $r^1, r^2, ..., r^m$ called *outcomes* of the race. At a critical race, the stable recursion function s is set-valued $s(r,v) := \{r^1, r^2, ..., r^m\}$. When critical races are involved, the symbols $x_1, x_2, ...$ in (2) may represent sets of states. A set of states $S \subset X$ and an input character $u \in A$ form a *valid pair* (S,u) if (x,u) is a valid pair for all $x \in S$. The following notion is from [4].

Definition 3. Let $y_1,...,y_q \in Y$ be a set of characters satisfying $y_{i+1} \neq y_i$ for all $i = 1,...,q-1$. Then, the *burst* of a string $y = y_1y_1...y_1y_2y_2...y_2...y_qy_q...y_q$ is $\beta(y) := y_1y_2...y_{q-1}y_q$, i.e., the string obtained by removing all repeats of consecutive characters.

For a valid pair (x,u) of an asynchronous machine $\Sigma = (A,Y,X,x_0,f,h)$, let $x_1 \in f(x,u), x_2 \in f(x_1,u), ..., x_m \in f(x_{m-1},u)$, $x_m = f(x_m,u)$ be a string of transitions generated by Σ from (x,u) and ending at the stable state x_m. Then, the corresponding *burst* is the string of output characters $\beta(x,u,x_1x_2...x_m) := \beta\big(h(x)h(x_1)h(x_2)...h(x_{m-1})h(x_m)\big)$. □

In the transition chain $x, x_1, x_2, ..., x_m$ of Definition 3, set $x' := x_m$. The presence of critical races raises the possibility of a different chain of transitions leading from x to x', say one passing the states $x, x'_1, ...x'_q$, where $x'_q = x'$. The resulting burst $\beta(x,u,x'_1x'_2...x'_q)$ could be different from $\beta(x,u,x_1x_2...x_m)$. Consequently, a transition from one state to another may induce multiple bursts. We denote all such bursts by

$$\beta(x,u,x') := \left\{ \begin{array}{l|l} \beta(x,u,x'_1x'_2...x'_q) & \begin{array}{l}\text{where } x'_1, x'_2, ..., x'_q \text{ are the} \\ \text{states of a transition chain} \\ \text{from } x \text{ to } x' \text{with input } u.\end{array} \\ & \\ \varnothing & \begin{array}{l}\text{if there is no transition} \\ \text{from } (x,u) \text{ to } x',\end{array} \end{array} \right. \tag{3}$$

where $\varnothing$ is the empty set. For a critical race pair (r,v) with the outcomes $r^1, r^2, ..., r^m$, the set of possible bursts is

$$\beta(r,v) := \bigcup_{i=1,...,m} \beta(r,v,r^i).$$

For a transition chain of $q \ge 1$ steps through the states $x_1, x_2, ...x_q$, the burst induced by the first $q-1$ steps is

$$\beta_{-1}(x,u,x_1,x_2,...,x_q) := \left\{ \begin{array}{l} \beta(x,u,x_1,x_2,...,x_{q-1}) \text{ if } q > 1, \\ \varnothing \text{ if } q = 1. \end{array} \right.$$

2.2 Detectability

To achieve fundamental mode operation of the closed loop machine Σ_c, the controller C must keep its output constant until Σ has reached its next stable state. As C can access only input and output data of Σ, this leads to the following notion (compare to [3], [4], [13], [14], [15]).

Definition 4. Let S be a set of states of an asynchronous machine Σ. Assume that Σ is at a stable combination with an unspecified member $x \in S$, when the input character of Σ switches to u, where (S,u) is a valid pair. Then, Σ is *detectable* at (S,u) if it is possible to determine from u and the output burst of Σ whether Σ has reached its next stable state. □

To analyze Definition 4, consider a transition chain of Σ through the states $x_1, x_2, ...x_q$ to a stable combination with x_q. In order to determine whether x_q has been reached, it must be possible to determine when the related burst terminates; this is possible only if there is a switch at the last burst character, namely if (see [3] and [4])

$$\beta_{-1}(x,u,x_1,x_2,...,x_q) \neq \beta(x,u,x_1,x_2,...,x_q). \tag{4}$$

However, the uncertainty caused by critical races makes this condition insufficient. For example, consider a situation where Σ rests at one of two states, say x' or x'', with the exact state being unspecified, and let u be an input character for which $(\{x',x''\},u)$ forms a valid pair. Assume that x'_1 is the next stable state of (x',u) and that x''_1 is the next stable state of (x'',u). Assume further that $\beta' := y_1y_2y_3$ is the burst generated by (x',u), while $\beta'' := y_1y_2y_3y_4$ is the burst generated by

(x'', u). Then, $\beta'_{-1} = y_1y_2 \neq \beta'$ and $\beta''_{-1} = y_1y_2y_3 \neq \beta''$, so that (4) is valid. However, when receiving the burst $y_1y_2y_3$, it is impossible to tell whether Σ has reached x'_1 or whether Σ is on its way to x''_1, as it is not known whether Σ started from x' or from x''. These observations essentially prove the following characterization of detectability. Given a valid pair (S,u), where S is a subset of states of Σ, denote by

$$B(S,u) := \bigcup_{x\in S, x'\in s(x,u)} \beta(x,u,x') \quad (5)$$

the set of all bursts that can be generated from states of S by the input character u. Then, the next statement is true (see also [15]).

Proposition 5. Let S be a set of states of an asynchronous machine Σ, let $u \in A$ be an input character that forms a valid pair with S, and let $B(S,u)$ be the set of all bursts that can be generated by (S,u). Then, (S,u) is detectable if and only if the following are true for all bursts $b \in B(S,u)$:
(i) $b_{-1} \neq b$, and
(ii) b is not a strict prefix of any burst in $B(S,u)$.

Example 6. Consider an asynchronous machine Σ with the input alphabet $A = \{a,b\}$, the output alphabet $Y = \{0,1\}$, the state set $X = \{x^1,x^2,x^3,x^4\}$, and the transition table:

	a	b	Y
x^1	$\{x^2,x^3\}$	x^1	0
x^2	x^2	x^4	0
x^3	x^3	x^4	0
x^4	x^4	x^1	1

For the set $S := \{x^2,x^3\}$, the pair (S,b) is valid and induces the transitions $x^2 \to x^4 \to x^1$ and $x^3 \to x^4 \to x^1$, with the output bursts $\beta(x^2,b,x^1) = 010$ and $\beta(x^3,b,x^1) = 010$, respectively. Thus, $B(S,b) = \{010\}$, and (S,b) is detectable by Proposition 5. In contrast, (S,b) is not strongly detectable in the sense of [13]. Thus, the use of bursts enhances control capabilities. □

3. GENERALIZED REALIZATIONS

Following the general ideas of ([21], [13], and [14]), we develop in this section a notion of generalized realization that facilitates the utilization of bursts. It is based on the following equivalence relation, under which states are equivalent if the process of reaching them generates the same burst.

Definition 7. Let Σ be an asynchronous machine with the stable recursion function s, let S be a set of states of Σ, and let $u \in A$ be an input character for which (S,u) is a valid pair. For a burst β, the set of *burst equivalent states* $S(\beta)$ is the set of all states $x' \in s(S,u)$ for which $\beta \in \beta(x,u,x')$ for some state $x \in S$. □

A stable transition of Σ that starts at an unspecified state in S and produces a burst β, ends at a member of $S(\beta)$. It is impossible to determine from input and output data at which member of $S(\beta)$ the machine rests after the transition.

Example 8. For the set S of Example 6, we have $S(010) = x^1$. Consequently, there is no uncertainty after the transition in this case. □

The notion of generalized realization ([21], [13], [14], and [15]) can be adapted to our present situation as follows. (We denote by $\#X$ the cardinality of a set X and by $P(X)$ the family of all subsets of X.)

Definition 9. Let $\Sigma = (A,Y,X,x_0,f,h)$ be an asynchronous machine with the stable recursion function s, let χ be a set disjoint from X with at least $2^{\#X}$ elements, and let $\Phi : P(X) \to X \cup \chi$ be an injective function satisfying $\Phi(x) = x$ for all states $x \in X$. With a burst equivalent set of states S, associate the element $\xi := \Phi(S)$.

If $\#S > 1$, then ξ is a *group state* of Σ, while S is the *underlying set* of ξ and is denoted by $S(\xi)$. For an input character $u \in A$, the pair (ξ,u) is *valid* if (S,u) is a valid pair.

A *generalized state set* $\tilde{X}$ of Σ is a union of the original state set X with a set of group states such that the following is true for all valid pairs $(\xi,u) \in \tilde{X} \times A$: every burst equivalent subset of $s(S(\xi),u)$ is either a single state or is represented by a group state in $\tilde{X}$. □

The recursion function associated with a generalized realization is as follows.

Definition 10. Let $\Sigma = (A,Y,X,x_0,f,h)$ be an asynchronous machine with the stable recursion function s, and let $\tilde{X}$ be a generalized state set of Σ. For a member $\zeta \in \tilde{X}$, denote by $S(\zeta)$ the underlying set of states, where $S(\zeta) := \zeta$ when $\zeta \in X$. For a valid pair $(\zeta,u) \in \tilde{X} \times A$, let $\{S_1,...,S_m\}$ be the family of all burst equivalent subsets of $s[S(\zeta),u]$, and let $\zeta^i \in \tilde{X}$ be the generalized state associated with $S_i, i = 1,...,m$. Then, the *generalized stable recursion function* s_g is defined by

$$s_g(\zeta,u) := \{\zeta^1,...,\zeta^m\} \text{ for all valid pairs } (\zeta,u) \in \tilde{X} \times A. \quad (6)$$

The *generalized output function* $h_g : \tilde{X} \to Y$ is

$$h_g(\zeta) := h[S(\zeta)] \text{ for all } \zeta \in \tilde{X}. \quad (7)$$

The sextuple $\Sigma_g := (A,Y,\tilde{X},x_0,s_g,h_g)$ forms a *generalized realization* of Σ. □

Recalling that members of the set $S(\zeta)$ are all burst equivalent, we conclude that set $h[S(\zeta)]$ of (7) consists of a single character (equal to the last output character of the burst generated by members of $S(\zeta)$). Thus, the generalized realization Σ_g has the same input/output behavior as the original machine Σ and hence just forms another realization of Σ.

By Definition 9, the generalized state of a machine is uniquely determined by the burst of the most recent stable state transition, since each generalized state represents a specific burst equivalent set. Consequently, a generalized realization creates a deterministic relationship between output values and generalized states of a possibly non-deterministic machine. An algorithm for the construction of generalized realizations is developed in [15].

Example 11. Applying Definitions 9 and 10 to the machine Σ of Example 6, leads to the group state $x^5 := \{x^2,x^3\}$ and the generalized state set $\tilde{X} = \{x^1,x^2,x^3,x^4,x^5\}$ for Σ. The corresponding generalized transition function s_g and output function h_g are described by the following transition table.

	a	b	Y
x^1	x^5	x^1	0
x^2	x^2	x^4	0
x^3	x^3	x^4	0
x^4	x^4	x^1	1
x^5	x^5	x^1	0

□

4. OBSERVERS

We decompose the controller C of Figure 1 into a combination an observer ϑ and a control unit F, as described in Figure 2.

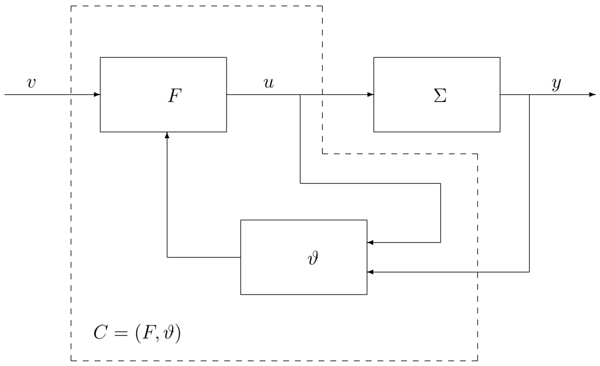

Fig. 2. Controller Structure

Here, Σ is the asynchronous machine being controlled; ϑ is an asynchronous machine that serves as an observer, determining the latest stable generalized state of Σ; and F is an asynchronous machine serving as a controller generating input strings that drive Σ to a desired outcome.

The observer ϑ is based on a generalized realization $\Sigma_g = (A, Y, \tilde{X}, x_0, s_g, h_g)$ of Σ. It is an input/state machine $\vartheta = (A \times Y^*, \tilde{X}, \tilde{X}, x_0, \sigma, I)$ with two inputs: the input $u \in A$ of Σ and the output burst $\beta \in Y^*$ of Σ. The state set of ϑ is identical to the generalized state set $\tilde{X}$ of the observed machine Σ, and the output of ϑ is its state. Recall that $\beta(x, u, \zeta)$ is the set of all bursts that may be generated by Σ when going from the pair (x, u) to a stable combination with the generalized state ζ. For a burst β of Σ, denote by β_k the part of the burst β up to step k. Then, the recursion function of σ of ϑ is defined by

$$\sigma(x, u, \beta_k) := \begin{cases} \zeta \in s_g(x, u) \text{ if } \beta_k \in \beta(x, u, \zeta), \\ x \text{ otherwise.} \end{cases} \tag{8}$$

For a detectable transition of Σ, the value of $\sigma(x, u, \beta_k)$ is uniquely determined by (8), since the end of the burst β is uniquely identifiable. Being an input/state machine, the observer ϑ is described by the recursion

$$\vartheta : \ z_{k+1} = \sigma(z_k, u_k, \beta_k), k = 0, 1, 2, ..., \tag{9}$$

where z_k is the current state (and output) of ϑ. Note that ϑ rests in its current state until the end of the burst β, namely, ϑ rests until Σ has reached its next stable state. This assures fundamental mode operation of the combination of Σ and ϑ, when the operation of Σ is restricted to detectable transitions. Non-detectable transitions of Σ lead to a violation of fundamental mode operation and consequently cannot be utilized.

In summary, the current state of the observer ϑ is equal to the last generalized stable state reached by the observed machine Σ through a detectable transition.

Example 12. An observer for the generalized realization Σ_g of Example 11: Following (8), we obtain the following transition diagram for ϑ (only detectable combinations are listed; a hyphen indicates that the corresponding combination is not used).

	$a,0$	$a,1$	$b,0$	$b,10$	$b,010$
x^1	-	-	x^1	-	-
x^2	x^2	-	-	-	x^1
x^3	x^3	-	-	-	x^1
x^4	-	x^4	-	x^1	-
x^5	x^5	-	-	-	-

□

5. THE CONTROL UNIT

In general terms, the structure of the control unit F of Figure 2 is analogous the control unit of [4], except that a generalized realization is used instead of a (regular) realization, as follows.

5.1 The Generalized Reachability Matrix

Definition 13. Let Σ be an asynchronous machine with the generalized realization $\Sigma_g = (A, Y, \tilde{X}, x_0, s_g, h_g)$, where $\tilde{X} = \{x^1, ..., x^q\}$. The set of input characters that induce a detectable transition from x^i to x^j is

$$\alpha(x^i, x^j) := \{a \in A : (x^i, a) \text{ is detectable and } x^j \in s_g(x^i, a)\} \tag{10}$$

The *generalized one-step reachability matrix* $R_g(\Sigma)$ is a $q \times q$ matrix with the entries

$$R_{g_{ij}}(\Sigma) := \begin{cases} \alpha(x^i, x^j) \text{ if } \alpha(x^i, x^j) \neq \varnothing, \\ N \text{ if } \alpha(x^i, x^j) = \varnothing, \end{cases} \tag{11}$$

where $i, j = 1, 2, ..., q$, and N is a character not in A. □

Example 14. The generalized one-step reachability matrix of the generalized realization of Example 11 is

$$R_g(\Sigma) = \begin{pmatrix} \{b\} & N & N & N & N \\ \{b\} & \{a\} & N & N & N \\ \{b\} & N & \{a\} & N & N \\ \{b\} & N & N & \{a\} & N \\ N & N & N & N & \{a\} \end{pmatrix}. \square$$

To consider transitions that consist of multiple steps, we use the operations on strings defined in [12]. In brief terms, let $w_1, w_2 \in A^+ \cup N$ be two strings, where N is a character not in A. The *unison* is then

$$w_1 \uplus w_2 := \begin{cases} w_1 \cup w_2 \text{ if } w_1, w_2 \in A^+; \\ w_1 \text{ if } w_1 \in A^+ \text{ and } w_2 = N; \\ w_2 \text{ if } w_1 = N \text{ and } w_2 \in A^+; \\ N \text{ if } w_1 = w_2 = N. \end{cases}$$

For two subsets $\sigma_1, \sigma_2 \subset A^+ \cup N$, the *unison* is

$$\sigma_1 \uplus \sigma_2 := \{w_1 \uplus w_2 : w_1 \in \sigma_1 \text{ and } w_2 \in \sigma_2\}.$$

The *concatenation* is

$$conc(w_1, w_2) := \begin{cases} w_2 w_1 \text{ if } w_1, w_2 \in A^+; \\ N \text{ if } w_1 = N \text{ or } w_2 = N, \end{cases}$$

and for two subsets $W, V \subset A^+ \cup N$

$$conc(W, V) := \uplus_{w \in W, v \in V} conc(w, v).$$

The *product* $Z := CD$ of two $n \times n$ matrices C, D whose entries are subsets of $A^+ \cup N$ is an $n \times n$ matrix whose (i, j) entry is

$$Z_{ij} := \uplus_{k=1,2,...,n} conc(C_{ik}, D_{kj}), i, j = 1, ..., n.$$

Then, the power

$$R_g^t(\Sigma) := R_g^{t-1}(\Sigma) R_g(\Sigma), t = 2, 3, ...$$

has an i,j entry that consists of all strings of t characters that take Σ form a stable combination with x^i to a stable combination with x^j through a string of stable and detectable transitions (x^j might be just one outcome of a critical race). Combining the powers of $R_g(\Sigma)$, we obtain

$$R_g^{(t)}(\Sigma) := \Cup_{r=1,\dots,t} R_g^r(\Sigma), t = 2,3,\dots \tag{12}$$

The i,j entry of $R_g^{(t)}(\Sigma)$ consists of all strings of t or fewer characters that take Σ form a stable combination with x^i to a stable combination with x^j through a string of stable and detectable transitions (as before, x^j might be just one possible outcome of a critical race).

Definition 15. The *generalized stable reachability matrix* of Σ is $\Gamma_g(\Sigma) := R_g{}^{(q-1)}(\Sigma)$, where q is the number of generalized states of Σ. □

The significance of the generalized stable reachability matrix is demonstrated by the following statement, whose proof is similar to the proof of [11, Lemma 3.2].

Lemma 16. Let Σ be an asynchronous machine with the generalized state set $\tilde{X} = \{x^1,\dots,x^q\}$, and let $\Gamma_g(\Sigma)$ be the generalized stable reachability matrix of Σ. Then the following two statements are equivalent for all pairs of states $x^i, x^j \in \tilde{X}$.
(i) x^j is stably reachable from x^i through a string of stable and detectable transitions, possibly as one outcome of a critical race.
(ii) The (i,j) entry of $\Gamma_g(\Sigma)$ is not N. □

Example 17. Using the one-step reachability matrix of Example 14, we obtain

$$\Gamma_g(\Sigma) = \begin{pmatrix} \{b\} & N & N & N & N \\ \{b,ab\} & \{a\} & N & N & N \\ \{b,ab\} & N & \{a\} & N & N \\ \{b,ab\} & N & N & \{a\} & N \\ N & N & N & N & \{a\} \end{pmatrix}. \square$$

5.2 Feedback Paths

The notion of feedback path is intimately related to feedback control of non-deterministic asynchronous machines, as we discuss next (compare to [19], [20], and [21]). Below, $\Pi_x : \tilde{X} \times A \to \tilde{X} : \Pi_x(x,u) := x$ denotes the standard projection onto the generalized state set.

Definition 18. Let Σ be an asynchronous machine with the generalized state set $\tilde{X}$ and the generalized stable recursion function s_g, and let x', x'' be two generalized states of Σ. A *detectable feedback path* from x' to x'' is a list $S_0, S_1, \dots, S_p \subset \tilde{X} \times A$ of sets of detectable pairs with the following features:
(i) $S_0 = \{(x', u_0)\}$ for some $u_0 \in A$ (i.e., S_0 consists of a single pair);
(ii) $s_g[S_i] \subset \Pi_x[S_{i+1}], i = 0,\dots,p-1$; and
(iii) $s_g[S_p] = \{x''\}$. □

A detectable feedback path singles out a string of single-step stable and detectable transitions, where pairs of the set S_i transition to pairs of the set S_{i+1}. A *deterministic transition* is a transition or a string of transitions with a unique outcome. The importance of detectable feedback paths originates from the following (for proof, see analogous results in [19], [20], and [21]).

Proposition 19. Let Σ be an asynchronous machine with the generalized realization Σ_g, and let x' and x'' be generalized states of Σ. Then, the following are equivalent.
(i) There is an output feedback controller C for which the closed loop machine Σ_{g_c} has a deterministic stable transition from x' to x'' in fundamental mode operation.
(ii) There is a detectable feedback path from x' to x''. □

Recall that the entries of the generalized stable reachability matrix consist of input strings that take the machine from one stable combination to another through several detectable and stable transitions. By investigating these entries, one can determine whether they include detectable feedback paths. (A formal algorithm for finding detectable feedback paths is developed in [15].) The set of all pairs of generalized states that are connected by a detectable feedback path are characterized by a matrix of zeros and ones called the *generalized skeleton matrix* and denoted by $K_g(\Sigma)$. Entry i,j of $K_g(\Sigma)$ is 1 when there is a detectable feedback path from the generalized state x^i to the generalized state x^j; otherwise, the entry is zero.

Example 20. The generalized skeleton matrix of the machine Σ of Example 6:

$$K_g(\Sigma) = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix}. \square$$

5.3 Model Matching

Model matching depends on the following notion (compare to [3] and [4]).

Definition 21. Let $\Lambda = \{\Lambda^1,\dots,\Lambda^q\}$ and $W = \{W^1,\dots,W^q\}$ be two lists of subsets of a set $\tilde{X}$. The *length* of the list Λ is the number q of its members. The list W is a *subordinate list* of the list Λ ($W \prec \Lambda$) if W has the same length as Λ and if $W^i \subset \Lambda^i$ for all $i = 1,\dots,q$. A list is *deficient* if one of its members is the empty set. □

Given two sets S^1 and S^2 and a function $g : S^1 \to S^2$, the inverse set function g^I of g assigns to each element $s \in S_2$ the set $g^I(s)$ of all elements $\alpha \in S^1$ satisfying $g(\alpha) = s$.

Definition 22. Let Σ be an asynchronous machine with the generalized realization $\Sigma_g = (A, Y, \tilde{X}, x_0, s_g, h_g)$, and let $\Sigma' = (A, Y, X', \zeta_0, s', h')$ be a stable-state asynchronous machine with the state set $X' = \{\zeta^1,\dots,\zeta^q\}$. The *generalized output equivalence list* of Σ with respect to Σ' is $E_g(\Sigma,\Sigma') := \{E^1,\dots,E^q\}$, where $E^i := h_g^I h'(\zeta^i), i = 1,\dots,q$. □

Member i of $E_g(\Sigma,\Sigma')$ consists of all states of Σ that produce the same output as state i of Σ'.

Example 23. Let Σ' be a stable state asynchronous machine with the state set $\{\zeta^1,\zeta^2\}$, the input alphabet $A = \{a,b\}$, the output alphabet $Y = \{0,1\}$, and the transition table

	a	b	Y
ζ^1	ζ^1	ζ^2	1
ζ^2	ζ^1	ζ^2	0

Then, for the machine Σ with the generalized realization of Example 11, the generalized output equivalence list with respect to Σ' is $E_g(\Sigma,\Sigma') = \{E^1,E^2\}$, where $E^1 = \{x^4\}$ and $E^2 = \{x^1,x^2,x^3,x^5\}$. □

Definition 24. Let Σ be an asynchronous machine with the generalized state set $\tilde{X}$, and let Λ^1 and Λ^2 be two nonempty subsets of $\tilde{X}$. The *generalized reachability indicator* $r_g(\Sigma,\Lambda^1,\Lambda^2)$ is 1 if there is a detectable feedback path from every element of Λ^1 to an element of Λ^2; otherwise, $r_g(\Sigma,\Lambda^1,\Lambda^2) := 0$.

Let $\Lambda = \{\Lambda^1,...,\Lambda^q\}$ be a list of $q \geq 1$ non-empty subsets of $\tilde{X}$. The *generalized fused skeleton matrix* $\Delta_g(\Sigma,\Lambda)$ is a $q \times q$ matrix whose (i,j) entry is $\Delta_{g_{ij}}(\Sigma,\Lambda) := r_g(\Sigma,\Lambda^i,\Lambda^j), i,j = 1,2,...,q$. □

The significance of fused skeleton matrices originates from the following statement, whose proof parallels the proof of [4, Corollary 53]. (Skeleton matrices of deterministic asynchronous machines are discussed in [11].)

Theorem 25. Let Σ be an asynchronous machine with the initial condition x_0. Let Σ' be a stably minimal asynchronous machine with the skeleton matrix $K(\Sigma')$, the state set $X' = \{\zeta^1,...,\zeta^q\}$, and the initial condition $\zeta_0 = \zeta^d$, where $d \in \{1,2,...,q\}$. Then the following two statements are equivalent.
(i) There is a controller C for which $\Sigma_c = \Sigma'$, where Σ_c operates in fundamental mode and is well posed.
(ii) There is a non-deficient generalized subordinate list $\Lambda = \{\Lambda^1,...,\Lambda^q\}$ of the generalized output equivalence list $E_g(\Sigma,\Sigma')$ such that $\Delta_g(\Sigma,\Lambda) \geq K(\Sigma')$ and $x_0 \in \Lambda^d$.

Moreover, when (ii) is valid, the controller C can be implemented as a combination of an observer ϑ and a control unit F as depicted in Figure 2, with ϑ being given by (9). □

Theorem 25 shows that model matching depends on the derivation of an appropriate subordinate list of the output equivalence list. This can be accomplished by using a variant of [4, Algorithm 54]. Then, a controller that achieves model matching can be constructed by a procedure that is analogous to the construction described in the proof of [4, Theorem 51].

To conclude, the present note presents the underpinnings of a methodology for the design of output feedback controllers that utilize bursts to achieve deterministic model matching for non-deterministic asynchronous machines. The controller design process is based on a new class of generalized realizations. A comprehensive presentation of tools for the construction of appropriate controllers is provided in [15].

REFERENCES

[1] G. Barrett and S. Lafortune [1998] "Bisimulation, the supervisory control problem, and strong model matching for finite state machines," Discrete Event Dynamic Systems: Theory and Application, vol. 8, no. 4, pp. 377-429. M.

[2] D. Dibenedetto, A. Saldanha and A. Sangiovanni-Vincentelli [1994] "Model matching for finite state machines," Proceedings of the IEEE Conference on Decision and Control, vol. 3, 1994, pp. 3117-3124.

[3] X. J. Geng and J. Hammer [2004] "Asynchronous sequential machines: input/output control," Proceedings of the 12th Mediterranean Conference on Control and Automation, Kusadasi, Turkey, June 2004.

[4] X. J. Geng and J. Hammer [2005] "Input/output control of asynchronous sequential machines," IEEE Transactions on Automatic Control, vol. 50, no. 12, pp. 1956-1970.

[5] J. Hammer [1994] "On some control problems in molecular biology," Proceedings of the IEEE conference on Decision and Control, December 1994.

[6] J. Hammer [1995] "On the modeling and control of biological signal chains," Proceedings of the IEEE conference on Decision and Control, December 1995.

[7] J. Hammer [1996a] "On the corrective control of sequential machines," International Journal of Control, vol. 65, no. 2, pp. 249-276.

[8] J. Hammer [1996b] "On the control of incompletely described sequential machines," International Journal of Control, vol. 63, no. 6, pp. 1005-1028.

[9] J. Hammer [1997] "On the control of sequential machines with disturbances," International Journal of Control, vol. 67, no. 3, pp. 307-331.

[10] Z. Kohavi [1970] "Switching and Finite Automata Theory," McGraw-Hill Book Company, New York.

[11] T. E. Murphy, X. J. Geng and J. Hammer [2002] "Controlling races in asynchronous sequential machines," Proceeding of the IFAC World Congress, Barcelona, July 2002.

[12] T. E. Murphy, X. J. Geng and J. Hammer [2003] "On the control of asynchronous machines with races," IEEE Transactions on Automatic Control, vol. 48, no. 6, pp. 1073-1081.

[13] J. Peng and J. Hammer [2009a] "Input/Output Control of Asynchronous Sequential Machines with Races", International J. of Control, Vol. 83, No. 1, pp. 125-144.

[14] J. Peng and J. Hammer [2009b] "Generalized Realizations and Output Feedback Control of Asynchronous Sequential Machines with Races", Proceedings of the European Control Conference, Budapest, Hungary, August 2009.

[15] J. Peng and J. Hammer [2009c] "Bursts and Output Feedback Control of Non-Deterministic Asynchronous Sequential Machines", submitted for publication.

[16] P. J. G. Ramadge and W. M. Wonham [1987] "Supervisory control of a class of discrete event processes," SIAM Journal of Control and Optimization, vol. 25, no. 1, pp. 206-230.

[17] J. G. Thistle and W. M. Wonham [1994] "Control of infinite behavior of finite automata," SIAM Journal on Control and Optimization, vol. 32, no. 4, pp. 1075-1097.

[18] S.H. Unger [1995] "Hazards, critical races, and metastability," IEEE Transactions on Computers, vol.44, no.6, pp. 754-768.

[19] N. Venkatraman and J. Hammer [2006a] "Stable realizations of asynchronous sequential machines with infinite cycles," Proceedings of the 2006 Asian Control Conference, Bali, Indonesia, 2006.

[20] N. Venkatraman and J. Hammer [2006b] "Controllers for asynchronous machines with infinite cycles," Proceedings of the 17th International Symposium on Mathematical Theory of Networks and Systems, Kyoto, Japan, 2006.

[21] N. Venkatraman and J. Hammer [2006c] "On the control of asynchronous sequential machines with infinite cycles," International Journal of Control, vol. 79, no. 7, pp. 764-785.

[22] J.M. Yang and J. Hammer [2008] "State Feedback Control of Asynchronous Sequential Machines with Adversarial Inputs", International J. Control, Vol. 81, No. 12, pp. 1910-1929.