Proceedings of the 17th World Congress
The International Federation of Automatic Control
Seoul, Korea, July 6-11, 2008

Counteracting the Effects of Adversarial Inputs on Asynchronous Sequential Machines

Jung-Min Yang* and Jacob Hammer**

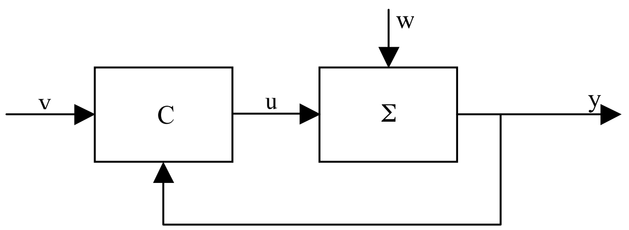
**Department of Electrical Engineering, Catholic University of Daegu, 330 Kumrak, Hayang, Kyongsan, Kyongbuk, 712-702, Republic of Korea; e-mail: jmyang@cu.ac.kr*
***Department of Electrical and Computer Engineering, University of Florida, P.O. Box 116130, Gainesville, FL 32611, USA; e-mail: hammer@mst.ufl.edu*
***The work of this author was supported in part by the US Air Force, grant number FA8750-06-1-0175.*

Abstract: The problem of counteracting the effects of adversarial inputs on the operation of an asynchronous sequential machine is considered. The objective is to build an automatic state-feedback controller that returns an asynchronous sequential machine to its original state, after the machine has undergone a state transition caused by an adversarial input. It is shown that the existence of such a controller depends on certain reachability and detectability properties of the affected machine.

1. INTRODUCTION

In modern computing systems, one often encounters unauthorized and adversarial input agents that attempt to interfere with the proper operation of the system. We address the question of how a computing system can be made immune to such interferences. Our approach is based on automatic control: we deploy feedback controllers that take corrective action whenever an adversarial input attempts to affect the operation of the underlying computing system.

Asynchronous sequential machines are important building blocks of high-speed digital computer systems. In this note, we consider asynchronous sequential machines with two inputs: an input for controlling the machine (the *control input*), and an input used by an adversarial agent (the *adversarial input*). The control diagram is as follows.



(1.1)

Here, Σ is the asynchronous sequential machine being controlled, and C is another asynchronous machine serving as the controller. The control input of Σ is u and the adversarial input is w; the closed loop machine is denoted by $\Sigma_c(v,w)$, with v being the external command input.

Our objective is to design a controller C that counteracts action at w, making it possible for the closed loop machine to operate without interference. Necessary and sufficient conditions for the existence of C are presented in section 6, which also includes a description of the controller's structure. There seem to be no earlier reports of this problem in the literature.

Recall that an asynchronous sequential machine can be in a stable state or in a transient state. At a stable state, the machine dwells indefinitely until the input character is changed. Transient states are traversed by the machine very quickly (ideally, in zero time), and are imperceptible by the user. Thus, when counteracting the effects of an adversarial input, it is only necessary to eliminate the effects on stable states. The operation of the controller C is, in fact, based on this principle: when the adversarial input causes a state transition of Σ, the controller turns the new state into a transient state of the closed loop machine, and returns Σ to the stable state it had before the interference. Thus, Σ resumes its original state very quickly (ideally, in zero time), and the effect of the adversarial input is eliminated.

Our discussion is within the framework developed by MURPHY, GENG, and HAMMER [2002 and 2003], GENG and HAMMER [2004 and 2005], and VENKATRAMAN and HAMMER [2006a, b, and c]. Studies dealing with other aspects of discrete event systems can be found in RAMADGE and WONHAM [1987], HAMMER [1994, 1995, 1996a, 1996b, 1997], DIBENEDETTO, SALDANHA, and SANGIOVANNI–VINCENTELLI [1994], THISTLE and WONHAM [1994], BARRETT and LAFORTUNE [1998], the references cited in these papers, and others. These publications do not address issues peculiar to the operation of asynchronous machines, such as the avoidance of critical races and the distinction between stable and transient states. To the best of our knowledge, the problem considered in this paper has not been previously addressed in the literature.

2. NOTATION AND BASICS

The machines we consider have a control input and an adversarial input, and they provide their state as output. Such machines are represented by a triplet $(A\times B,X,f)$, where A

978-1-1234-7890-2/08/$20.00 © 2008 IFAC 10.3182/20080706-5-KR-1001.1960

is the control input alphabet, B is the adversarial input alphabet, X is a set of states, and $f : X\times A\times B \rightarrow X$ is the *recursion function*. The operation is according to

$$x_{k+1} = f(x_k,u_k,w_k); \qquad (2.1)$$

here, $u_0, u_1, u_2, \ldots$ is the *control input sequence*; $w_0, w_1, w_2, \ldots$ is the *adversarial input sequence*; and $x_0, x_1, x_2, \ldots$ is the sequence of the machine's states. The *step counter* k advances by one at a change of the machine's inputs or state.

(2.2) EXAMPLE. An asynchronous machine Σ with the control input alphabet $A = \{a, b\}$; the adversarial input alphabet $B = \{\alpha, \beta\}$; and the state set $X = \{x^1, x^2, x^3\}$. The recursion function f is described by the transition table:

state	(a,α)	(a,β)	(b,α)	(b,β)
x^1	x^1	x^1	x^1	x^2
x^2	x^3	x^1	x^2	x^2
x^3	x^3	x^3	x^1	x^2

♦

A triplet (x,u,w) is a *stable combination* if $x = f(x,u,w)$, i.e., if the state x is a fixed point of the function f. A machine lingers at a stable combination until an input changes. A triplet (x,u,w) that is not a stable combination starts a chain of transitions $x_1 = f(x,u,w)$, $x_2 = f(x_1,u,w)$, ... If this chain terminates, then there is an integer $q \geq 1$ for which $x_q = f(x_q,u,w)$; then, (x_q,u,w) is a stable combination and x_q is the *next stable state*. If this chain of transitions does not terminate, then (x,u,w) is part of an *infinite cycle*. In this paper, we consider only machines with no infinite cycles; thus, every triplet (x,u,w) has a next stable state, as follows.

(2.3) LEMMA. In an asynchronous machine without infinite cycles, there is always a next stable state. ♦

To prevent unpredictable outcomes, it is common to enforce a policy where only one variable of an asynchronous machine is allowed change value at any instant of time (e.g., KOHAVI [1970]). This is referred to as *fundamental mode operation*. All the machines in this paper operate in fundamental mode.

(2.4) DEFINITION. An asynchronous machine Σ *operates in fundamental mode* if its inputs change value only when Σ is in a stable combination, and then at most one at a time. ♦

In fundamental mode operation of the configuration (1.1), only one of the machines Σ or C can undergo transitions at any instant of time. This leads us to:

(2.5) PROPOSITION. Configuration (1.1) operates in fundamental mode if and only if the following hold:

(i) C is in a stable combination while Σ undergoes transitions, and Σ is in a stable combination while C undergoes transitions.

(ii) The inputs u, w, and v change only while Σ and C are in a stable combination, and then only one at a time. ♦

Thus, the controller C must be designed so that (i) it commences transitions only after verifying that Σ is in a stable combination, and (ii) it adopts a stable combination before inducing a change in the input of Σ. This assures that the closed loop system is unambiguous and deterministic. As transitions of asynchronous machines are very quick (ideally, in zero time), fundamental mode operation is not restrictive.

3. ADVERSARIAL INPUTS

In general, the adversarial input character w_k is not specified; it is only known that it belongs to a specified subset $\nu \subset B$ called the *adversarial uncertainty*. To include this information, we write $\Sigma = (A\times B,X,f,\nu)$. Starting from the initial state x_0 and applying the control input character u_0, the next state of Σ can be any member of the set

$$f[x_0\times u_0\times\nu] := \cup_{w\in\nu} f(x_0,u_0,w) \subset X.$$

To describe stable transitions of the machine Σ, let x' be the next stable state of (x,u,w). The *stable recursion function* s is defined by setting $s(x,u,w) := x'$. Considering adversarial uncertainty, all possible next stable states form the set

$$s^{\nu}(x,u) := s[x,u,\nu] = \{s(x,u,w) : w \in \nu\} \subset X. \qquad (3.1)$$

4. DETECTABILITY AND REACHABILITY

By Proposition 2.5, fundamental mode operation requires the controller C to remain in a stable combination until Σ has reached its next stable state. To examine the conditions under which such a controller can be implemented, let w be an adversarial input character. Assume that Σ is in a stable combination at the state x, when the control input changes to u. Then, Σ embarks on the string of transitions

$$\theta(x,u,w) := \{x_1 := f(x,u,w),\ x_2 := f(x_1,u,w),\ \ldots,\ x_{i(u,w)} := f(x_{i(u,w)-1},u,w)\}, \qquad (4.1)$$

where $x_{i(u,w)}$ is the next stable state. The set of all transition strings consistent with the adversarial uncertainty ν is:

$$\theta[x,u,\nu] := \{\theta(x,u,w) : w \in \nu\}. \qquad (4.2)$$

The next notion characterizes our ability to determine by state feedback whether or not Σ has reached its next stable state.

(4.3) DEFINITION. Let Σ be in a stable combination with the state x, when the control input character changes to u. The pair (x,u) is *detectable* if it is possible to determine by state feedback whether Σ has reached its next stable state. ♦

Here is a test to determine whether a pair is detectable.

(4.4) THEOREM. Let Σ be in a stable combination with the state x, when the control input character changes to u. Then, (i) and (ii) are equivalent for adversarial uncertainty ε:

(i) The pair (x,u) is detectable.

(ii) States of the set $s^{\varepsilon}(x,u)$ appear only at the end of strings belonging to $\theta[x,u,\varepsilon]$.

Proof (sketch). Consider a string $\theta(x,u,w) = \{x_0, x_1, x_2, \ldots, x_{i(u,w)}\}$, where $w \in \varepsilon$, and assume, by contradiction, that (ii) is not valid. Then, $x_j \in s^{\varepsilon}(x,u)$ for an integer $0 \leq j < i(u,w)$, so that (x_j,u,w) is a transient combination, since $j < i(u,w)$. The inclusion $x_j \in s^{\varepsilon}(x,u)$ implies that there is an adversarial input character $w' \in \varepsilon$ for which (x_j,u,w') is a stable combination. Thus, x_j is a transient state in (x_j,u,w) while being a stable state in (x_j,u,w'), so that, at the state x_j, one cannot tell whether Σ is in a stable state. Whence, (i) implies (ii). Conversely, if every state $x' \in s^\varepsilon(x,u)$ appears only at the end of a string in $\theta[x,u,\varepsilon]$, then x' is always a stable state of Σ, and (ii) implies (i). ♦

Thus, we can determine whether Σ has reached its next stable state by checking if the current state of Σ is in $s^\varepsilon(x,u)$. Next, we adapt to our present setting the following notion from MURPHY, GENG, and HAMMER [2002] and [2003].

(4.5) DEFINITION. Let Σ be an asynchronous machine with the state set $X = \{x^1, x^2, ..., x^n\}$ and the stable transition function s. Let $w \in B$ be an adversarial input character. The *one-step matrix of stable transitions* $\rho(\Sigma,w)$ of Σ is an n×n matrix whose (i,j) entry $\rho_{ij}(\Sigma,w)$ consists of all pairs $w|u$, where u is a control input character satisfying $x^j = s(x^i,u,w)$; if there is no such u, then $\rho_{ij}(\Sigma,w) := w|N$, where N is a character not in A or in B:

$$\rho_{ij}(\Sigma,w) = \begin{cases} w|N \text{ if } \{u \in A : x^j = s(x^i,u,w)\} = \emptyset, \\ \{w|u : u \in A \text{ and } x^j = s(x^i,u,w)\} \text{ otherwise.} \end{cases}$$ ♦

(4.6) EXAMPLE. For the machine Σ of Example 2.2,

$$\rho(\Sigma,\alpha) = \begin{pmatrix} \{\alpha|a,\alpha|b\} & \{\alpha|N\} & \{\alpha|N\} \\ \{\alpha|N\} & \{\alpha|b\} & \{\alpha|a\} \\ \{\alpha|b\} & \{\alpha|N\} & \{\alpha|a\} \end{pmatrix}$$ ♦

To work with $\rho(\Sigma,w)$, we use the following projections: (A^+ is the set of all non-empty strings of characters of A.)

$$\Pi_a\, w|u := \begin{cases} w \text{ if } u \neq N, \\ \emptyset \text{ else,} \end{cases} \text{ (onto adversarial value), and}$$

$\Pi_c\, w|u := u$ for all $w|u \in B|(A^+ \cup N)$ (onto control value).

For two sets of strings $s_1, s_2 \subset B|(A^+ \cup N)$, we define an operation $s_1 \vee s_2$ akin to union by using the difference set

$$s_1 \vee s_2 := [s_1 \cup s_2] \setminus s_N,$$

where s_N is the set of all elements $w|N \in s_1 \cup s_2$ for which $[w|A^+] \cap [s_1 \cup s_2] \neq \emptyset$, i.e., all elements that appear both with N and non-N control input strings. Next, *concatenation* of strings $w_1|u_1, w_2|u_2 \in B|(A^+ \cup N)$ is given by

$$\text{conc}(w_1|u_1,w_2|u_2) := \begin{cases} w_1|u_1u_2 \text{ if } w_1 = w_2 \text{ and } u_1, u_2 \neq N, \\ w_1|N,\ w_2|N \text{ otherwise.} \end{cases}$$

For subsets of strings $\sigma_1, \sigma_2 \subset B|(A^+ \cup N)$:

$$\text{conc}(\sigma_1,\sigma_2) := \vee_{s_1\in\sigma_1, s_2\in\sigma_2} \text{conc}(s_1,s_2).$$

Now, define an operation similar to matrix multiplication for two n×n matrices P, Q with entries in $B|(A^+ \cup N)$:

$$(PQ)_{ij} := \vee_{k=1,...,n} \text{conc}(P_{ik},Q_{kj}),\ i, j = 1, 2, ..., n.$$

We can use the powers $\rho^k(\Sigma,w) = \rho^{k-1}(\Sigma,w)\rho(\Sigma,w)$, k = 1, 2, ... The i, j entry of $\rho^k(\Sigma,w)$ consists of all strings $w|u$ that take Σ from a stable combination with the state x^i to a stable combination with the state x^j in exactly k steps; if there is no such string, then u = N.

(4.7) EXAMPLE. Continuing Example 4.6:

$$\rho^2(\Sigma,\alpha) = \begin{pmatrix} \{\alpha|aa,\alpha|ab,\alpha|ba,\alpha|bb\} & \{\alpha|N\} & \{\alpha|N\} \\ \{\alpha|ab\} & \{\alpha|bb\} & \{\alpha|ba,\alpha|aa\} \\ \{\alpha|ab,\alpha|ba,\alpha|bb\} & \{\alpha|N\} & \{\alpha|aa\} \end{pmatrix}$$ ♦

The *matrix of m stable transitions* of Σ is defined by

$$R(m,\Sigma,w) := \vee_{i=1,...,m} \rho^i(\Sigma,w); \tag{4.8}$$

it characterizes all the transitions that can be accomplished in m or fewer stable steps. Allowing m to grow indefinitely yields the *extended matrix of stable transitions* $R^*(\Sigma,w) := \vee_{i\geq 1} \rho^i(\Sigma,w)$ that characterizes all stable transitions of Σ. The next statement resembles MURPHY, GENG, and HAMMER [2003, Proposition 3.9].

(4.9) LEMMA. The following are equivalent for all integers $m \geq n-1$ and all i, j = 1, 2, ..., n.
(i) The entry $R_{ij}(m,\Sigma,w)$ includes a string $w|u$ with $u \neq N$.
(ii) The entry $R^*_{ij}(\Sigma,w)$ includes a string $w|u$ with $u \neq N$. ♦

Thus, when $m \geq n-1$, the matrix $R(m,\Sigma,w)$ characterizes all stable transitions of Σ for the adversarial input character w.

(4.10) DEFINITION. For the adversarial input uncertainty ν, the *one-step stable transitions matrix* $\rho(\Sigma,\nu)$ is an n×n matrix with entries $\rho_{ij}(\Sigma,\nu) := \vee_{w\in\nu}\, \rho(\Sigma,w)$, i, j = 1, 2, ..., n. ♦

The matrix $\rho(\Sigma,\nu)$ includes all one-step stable transitions that are compatible with an adversarial input character in ν.

5. COMPLETE SETS OF STRINGS

The next notion is critical for feedback control (compare to VENKATRAMAN and HAMMER [2006a, b, c]).

(5.1) DEFINITION. Let Σ be an asynchronous machine with the adversarial uncertainty ν, and let x^i, x^j be states of Σ. There is a *feedback path* from x^i to x^j if there is a state feedback controller that takes Σ from a stable combination with x^i to a stable combination with x^j in fundamental mode, given only that the adversarial input is within ν. ♦

Below, we develop a test to determine whether there is a feedback path from x^i to x^j. If a feedback path exists, then an automatic controller can undo undesirable transitions from x^j to x^i. Note that, due to fundamental mode operation, the adversarial input remains constant along a feedback path.

Adversarial uncertainty may decline along a feedback path, since the machine's response provides information about the adversarial input. For example, let the adversarial uncertainty be $\nu = \{w^1, w^2\}$, and let s be the stable recursion function of Σ. Assume that Σ is at a stable combination with the state x and the control input character u, when the control input changes to u′. We have two options for the next stable state:

$x' := s(x,u',w^1)$ when the adversarial input character is w^1;
$x'' := s(x,u',w^2)$ when the adversarial input character is w^2.

Clearly, if $x' \neq x''$, then we can determine the adversarial input character from the next stable state, resolving the uncertainty. Thus, the adversarial uncertainty may decline along a feedback path. To discuss the general case, we need some notation. Let $S \subset B|A^+$ be the set of all strings that take Σ from a stable combination with the state x to a stable combination with the state x', i.e., all strings $w|u = w|u_0u_1... \in B|A^+$ for which $s(x,u,w) = x'$. For a string $\sigma = w|u_0u_1...u_k \in S$ and an integer $q \geq 0$, denote

$$\sigma|_q := \begin{cases} w|u_0u_1...u_q & \text{if } q \leq k, \\ w|u_0u_1...u_k & \text{if } q > k. \end{cases}$$

The string $\sigma|_q$ takes Σ to a stable combination with the state $x_q := s(x,\sigma|_q) := s(x,u_0u_1...u_q,w)$, passing through the stable states $x_0(\sigma) := s(x,\sigma|_0)$, $x_1(\sigma) := s(x,\sigma|_1)$, ..., $x_q(\sigma) := s(x,\sigma|_q)$, where $x_0(\sigma) = x$ and $x_k(\sigma) = x'$.

For a string $\sigma = w|u_0u_1...u_k \in S$, denote

$$\Pi^p \sigma := \begin{cases} u_p & \text{for } p = 0, 1, ..., k, \\ u_k & \text{for all } p > k. \end{cases}$$

Now, let Σ be in a stable combination with the state z when the control input value changes to u, and let z'' be the next stable state of Σ. The set of all adversarial input characters $w \in \nu$ compatible with the transition $s(z,u,w) = z''$ is

$$s^a(z,u,z'') := \{w \in \nu : s(z,u,w) = z''\}. \quad (5.2)$$

In particular, when Σ is at a stable combination with the initial state $x_0 := x$ and the control input character u_0, the adversarial input character w must satisfy

$$w \in \nu(x_0,u_0) := s^a(x_0,u_0,x_0) \cap \nu \subset \nu \quad (5.3)$$

Thus, $\nu(x_0,u_0)$ is the true initial adversarial uncertainty. For the transition from x_0 to x' to be possible, S must contain a path for each adversarial input character $w \in \nu(x_0,u_0)$, i.e., we must have $\nu(x_0,u_0) \subset \Pi_a S$. Otherwise, S would be incompatible with potential adversarial inputs. Further, let u_1 be a control input character, and define the set

$$S(x_0,u_0u_1) = \{\sigma \in S : \sigma|_1 = w|u_0u_1 \text{ for some } w \in B\}$$

of all strings of S whose control input starts with u_0u_1. Clearly, u_1 can be a next control character only if it is compatible with all possible adversarial inputs, i.e., only if

$$\nu(x_0,u_0) \subset \Pi_a S(x_0,u_0u_1).$$

Also, the pair (x_0,u_1) must be detectable to facilitate fundamental mode operation of the closed loop machine, since the controller must react at the next stable state.

Now, let x_1 be the next stable state reached with the control input character u_1; the state x_1 can be read by the state feedback controller. The fact that Σ reached x_1 implies that the adversarial input value w must be within the set

$$\nu(x_0x_1,u_0u_1) := s^a(x_0,u_1,x_1) \cap \nu(x_0,u_0).$$

Continuing in this way, suppose that we are at step p of the path. Let $u_0u_1...u_p$ be the control input characters applied so far, and let $x_0x_1...x_p$ be the stable states Σ has passed as a result. The current uncertainty $\nu(x_0...x_p,u_0...u_p) \subset B$ about the adversarial input value is called the *residual adversarial uncertainty*. By iterating the earlier steps, we get

$$\nu(x_0...x_p,u_0...u_p) := s^a(x_{p-1},u_p,x_p) \cap \nu(x_0...x_{p-1},u_0...u_{p-1}). \quad (5.4)$$

Now, let $S(x_0x_1...x_p,u_0u_1...u_p)$ be the set of all strings $\sigma \in S$ having the control inputs $u_0u_1...u_p$ and taking Σ through the stable states $x_0, x_1, ..., x_p$. For a control input character d, denote by $S(x_0x_1...x_p,u_0u_1...u_pd)$ the set of all strings $\sigma \in S(x_0x_1...x_p,u_0u_1...u_p)$ that have the character d as their next control input character. Then, the set of all adversarial input characters compatible with d is $\Pi_aS(x_0x_1...x_p,u_0u_1...u_pd)$. This set must be compatible with the residual adversarial uncertainty, namely,

(5.5) LEMMA. The character $d \in A$ can be used as the next control input character of the machine Σ only if $\nu(x_0x_1...x_p,u_0u_1...u_p) \subset \Pi_a S(x_0x_1...x_p,u_0u_1...u_pd)$. ♦

We show later that the condition of Lemma 5.5 is critical for the existence of a controller that automatically counteracts adversarial transitions. This leads us to the following.

(5.6) DEFINITION. Let $S \subset B|A^+$ be a set of strings taking Σ from a stable combination with the state x_0 to a stable combination with the state x'. Then, S is a *complete set* if the following hold for all $p = 0, 1, ...$ and for all control input characters $d \in \Pi^{p+1} S(x_0x_1...x_p,u_0u_1...u_p)$:
(i) $\nu(x_0x_1...x_p,u_0u_1...u_p) \subset \Pi_a S(x_0x_1...x_p,u_0u_1...u_pd)$, and
(ii) The pair (x_p,d) is detectable with respect to the residual adversarial uncertainty $\nu(x_0x_1...x_p,u_0u_1...u_p)$. ♦

A complete set of strings can be replaced by one of bounded length, as follows. (For a set of strings $S \subset B|A^+$, denote by $|S|$ the maximal length of a control input string in S. For a finite set Z, denote by $\#Z$ the number of elements in Z.)

(5.7) LEMMA. Let Σ be in a stable combination at the state x_0 and the control input value u_0. If there is a complete set of strings from x_0 to x', then there also is such a complete set S satisfying $|S| \leq [\#\nu(x_0,u_0)](n-1)$.

Proof (sketch). Consider a string $\sigma = w|u = w|u_0u_1...u_k \in S$ and let $x_0x_1...x_k$ be the stable states through which Σ passes as a result of receiving the control input string u. Let ν_i be the residual uncertainty at step i of the path, where $\nu_0 := \nu(x_0,u_0)$. Then, ν_i is a monotone declining function of i, and its minimal value is not less than 1. Divide the interval $[0, k]$ into segments of constant residual uncertainty $[0, i_1]$, $[i_1+1, i_2]$, ..., $[i_m+1, k]$, where ν_i is constant over each one of these subintervals. Since ν_i is monotonously declining and its minimum cannot be less than 1, we get $m + 1 \leq \nu(x_0,u_0)$, or $m \leq \nu(x_0,u_0) - 1$.

Now, if any of these subintervals $[i, i']$ has length $\ell \geq n$, then the string of states $x_ix_{i+1}...x_{i'}$ must contain a repeating state, say $x := x_p = x_r$, where $i \leq p < r \leq i+\ell$. Since $\nu_p = \nu_r$ by construction, the control input value u_p can be replaced by the control value u_r without disturbing the stable combination at step p. Then, steps $p+1, p+2, ..., r$ can be eliminated from the string, yielding a new segment with the length of $\ell - (r - p)$. This process can be repeated again and again, until the length of the resulting segment is less than n. Applying the same procedure to each one of the segments, we obtain a new path of length not exceeding $(m+1)(n-1) = [\#\nu(x_0,u_0)](n-1)$. ♦

This brings us to the main result of this section.

(5.8) THEOREM. Let $\Sigma = (A\times B,X,f,\nu)$ be an asynchronous machine and let x and x' be two states of Σ. Then, the following two statements are equivalent.
(i) There is a state feedback controller C that drives Σ from a stable combination with x to a stable combination with x' in fundamental mode operation.
(ii) There is a complete set of strings $S \subset B|A^+$ taking Σ from a stable combination with x to a stable combination with x'.

Proof (sketch). Assume that (ii) is valid. We use S to build a state feedback controller $F(x,x',\nu)$ which, upon receiving the external input character $v \in A$, generates a string of control input characters that takes Σ from a stable combination with $x_0 := x$ to a stable combination with x' in fundamental mode operation. To this end, let Σ be in a stable combination with x_0, and pick a control input character $u_1 \in \Pi^1 S$. Since S is a complete set of strings, (x_0,u_1) is detectable with respect $\nu(x_0,u_0)$. Also, $\nu(x_0,u_0) \subset \Pi_a S(x_0,u_0u_1)$, so u_1 is compatible with all possible adversarial inputs. Denote by Ξ the state set of $F(x,x',\nu)$, by ϕ the recursion function of $F(x,x',\nu)$, and by η the output function of $F(x,x',\nu)$; let ξ_0 be the initial state of $F(x,x',\nu)$. We construct now ϕ and η.

Upon a detectable transition of Σ to x_0 with the control input character u_0, the controller moves to a stable combination with its state ξ_1, readying for controller action at the command v. To this end, set

$$\phi(\xi_0,(z,t)) := \xi_0 \text{ for all } (z,t) \neq (x_0,u_0),$$
$$\phi(\xi_0,(x_0,u_0)) := \xi_1,\ \phi(\xi_1,(x_0,u_0)) := \xi_1.$$

While in the states ξ_0 or ξ_1, the controller applies to the control input of Σ the external input it receives, namely

$$\eta(\xi_0,(z,t)) := t,\ \eta(\xi_1,(z,t)) := t \text{ for all } (z,t) \in X\times A,$$

If, while at ξ_1, the controller $F(x,x',\nu)$ receives the external input character v (the command to start controller action), it moves to a stable combination with its state ξ_2:

$$\phi(\xi_1,(z,t)) := \xi_1 \text{ for all } (z,t) \neq (x_0,v),$$
$$\phi(\xi_1,(x_0,v)) := \xi_2,\ \phi(\xi_2,(x_0,v)) := \xi_2.$$

At ξ_2, the $F(x,x',\nu)$ applies the first character u_1 of the control input string that takes Σ to the state x', so we set

$$\eta(\xi_2,(x_0,t)) := u_1 \text{ for all } t \in A.$$

Since S is a complete set, u_1 makes Σ move to the state x_1 through a detectable transition. Whence, Σ is in a stable combination when it reaches x_1. Upon detecting x_1, the controller moves to a stable combination with its state ξ_3:

$$\phi(\xi_2,(z,t)) := \xi_2 \text{ for all } (z,t) \neq (x_1,u_1),$$
$$\phi(\xi_2,(x_1,v)) := \xi_3,\ \phi(\xi_3,(x_1,v)) := \xi_3.$$

At ξ_3, the controller applies the next control input value $u_2 \in \Pi^2 S(x_0x_1,u_0u_1)$: $\eta(\xi_3,(x_1,t)) := u_2$ for all $t \in A$. Since S is a complete set of strings, the pair (x_1,u_2) is detectable for the current adversarial uncertainty $\nu(x_0x_1,u_0u_1)$ and $\nu(x_0x_1,u_0u_1) \subset \Pi_a S(x_0x_1,u_0u_1u_2)$. We continue in this way until the controller $F(x,x',\nu)$ generates the last input character of the string, bringing Σ to x'. By Lemma (5.7), the state x' can be reached in at most $(n-1)[\#\nu(x_0,u_0)]$ steps.

Conversely, assume that (i) is valid, and let $F(x,x',\nu)$ be the corresponding controller. Let $S \subset B|A^+$ be the set of strings that $F(x,x',\nu)$ may generate for the various possible adversarial input characters. To show that S is a complete set, consider a control input string $u_0u_1...u_p$ that $F(x,x',\nu)$ applies to Σ, and let $x_0x_1...x_p$ be the stable states through which Σ passes as a result. By (5.4), the adversarial uncertainty at step $p \geq 0$ is $\nu(x_0x_1...x_p,u_0u_1...u_p)$. By fundamental mode operation of the closed loop machine, the pair (x_p,u_{p+1}) is detectable with respect to $\nu(x_0x_1...x_p,u_0u_1...u_p)$. By Lemma 5.5, $\nu(x_0x_1...x_p,u_0u_1...u_p) \subset \Pi_a S(x_0x_1...x_p,u_0u_1...u_pd)$. Hence, S is a complete set. ♦

An algorithm for finding complete sets of strings is described in YANG and HAMMER [2007]. We turn now to an important definition. By (5.3), we have $\nu(x_0,u_0) \subset \nu$, so that $\#\nu(x_0,u_0) \leq \#\nu$. Invoking Lemma 5.7, we conclude that a complete set of strings S can always be selected so that

$$|S| \leq (n-1)(\#\nu). \tag{5.9}$$

Recalling the matrix $R(m,\Sigma,w)$, taking $m = (n-1)(\#\nu)$, and including all adversarial characters of ν, we arrive at the following.

(5.10) DEFINITION. The $n\times n$ matrix

$$R(\Sigma,\nu) := \vee_{w\in\nu} R((n-1)(\#\nu),\Sigma,w)$$

is the *combined matrix of stable transitions* of an asynchronous machine Σ with adversarial uncertainty ν. ♦

Considering (5.9), Lemma 5.7, and Theorem 5.8, we reach the following conclusion.

(5.11) CORROLARY. Let Σ be an asynchronous machine with adversarial uncertainty ν and state set $X = \{x^1, ..., x^n\}$. The statements below are equivalent for all $i, j = 1, 2, ..., n$:
(i) There is a state feedback controller that takes Σ from a stable combination with x^i to a stable combination with x^j in fundamental mode operation.
(ii) The i, j entry of $R(\Sigma,\nu)$ includes a complete set of strings. ♦

6. COUNTERACTING ADVERSARIAL TRANSITIONS

Let Σ be an asynchronous machine at a stable combination (x,u,w), when the adversarial input character switches to w'. An *adversarial transition* occurs if this switch causes Σ to move to a new stable state $x' \neq x$. In this section, we discuss state feedback controllers that automatically counteract adversarial transitions. In order to operate in fundamental mode, it must be possible for the controller to determine from the state of Σ whether or not it has reached the next stable state. The following is analogous to Definition 4.3.

(6.1) DEFINITION. Let Σ be at a stable combination with the state x and the control input character u. The pair (x,u) is *adversarially detectable* if, after an adversarial transition, it can be determined from the current state of Σ whether or not Σ has reached its next stable state. ♦

Assume then that Σ is at a stable combination (x,u,w), when the adversarial input character changes to w', causing Σ to move to a stable combination with the state $x' \neq x$. This transition may consist of a number of intermediate steps, say $x_0 := x$, $x_1 := f(x_0,u,w')$, $x_2 = f(x_1,u,w')$, ..., $x_q := f(x_{q-1},u,w') = x'$, $x_q := f(x_q,u,w')$. Similarly to (4.1) and (4.2), we denote

$$\begin{cases} \theta(x,u,w') := x_1 \dots x_q, \\ \theta[x,u,\varepsilon] := \{\theta(x,u,w') : w' \in \varepsilon\}. \end{cases} \tag{6.2}$$

The following has a proof similar to that of Theorem 4.4.

(6.3) THEOREM. The two statements are equivalent:

(i) The pair (x,u) is adversarially detectable with respect to the adversarial uncertainty ν.
(ii) States of the set $s^{\nu}(x,u)$ appear only at the end of strings belonging to $\theta[x,u,\nu]$. ♦

To guarantee fundamental mode operation of the closed loop machine, the use of the machine Σ must be restricted to adversarially detectable pairs. This leads us to the following notion. (For a string $\sigma = w|u_1u_2 \dots u_q \in B \times A^+$, denote by $\Pi_c^+ \sigma := u_q$ the last control input character of the string.)

(6.4) DEFINITION. Let Σ be an asynchronous machine with n states, adversarial uncertainty ν, and combined matrix of stable transitions $R(\Sigma,\nu)$. The *reduced matrix of stable transitions* $R^r(\Sigma,\nu)$ is obtained by removing from each column $j = 1, 2, ..., n$ of $R(\Sigma,\nu)$ all strings σ for which the pair $(x^j,\Pi_c^+ \sigma)$ is not adversarially detectable. ♦

(6.5) EXAMPLE. In Example 2.2, Σ has only one adversarial transition: $(x^1,b,\alpha) \rightarrow (x^1,b,\beta) \rightarrow (x^2,b,\beta)$. Then, $\theta[x^1,b,\alpha] = x^1$ and $\theta[x^1,b,\beta] = x^2$, so $\theta[x^1,b,\nu] = \{x^1, x^2\}$. Also, $s^{\nu}(x^1,u) = \{x^1, x^2\}$ here. Thus (x^1,b) is adversarially detectable by Theorem a381, and $R^r(\Sigma,\nu) = R(\Sigma,\nu)$. ♦

The set of adversarial input characters that give rise to an adversarial transition from a stable combination with the pair (x^s,u) to a stable combination with the pair (x^t,u) is

$$\nu(x^s,x^t,u) := s^a(x^s,u,x^t) \cap \nu. \tag{6.6}$$

Here, a transition occurs if and only if $\nu(x^s,x^t,u) \neq \emptyset$. We can state now the main result of this section; the proof is similar to that of Theorem 5.8.

(6.7) THEOREM. Let Σ be an asynchronous machine with the state set $\{x^1, x^2, ..., x^n\}$ and the reduced matrix of stable transitions $R^r(\Sigma,\nu)$, and let x^s and x^t be states for which $\nu(x^s,x^t,u) \neq \emptyset$. Then, the following are equivalent:

(i) There is a state feedback controller that automatically reverses an adversarial transition from the state x^s to the state x^t in fundamental mode operation.
(ii) The entry $R^r_{ts}(\Sigma,\nu)$ includes a complete set of strings with respect to the adversarial uncertainty $\nu(x^s,x^t,u)$. ♦

7. REFERENCES

BARRETT, G. and LAFORTUNE, S. [1998], "Bisimulation, the supervisory control problem, and strong model matching for finite state machines," *Discrete Event Syst.: Theory Appl.*, **vol. 8**, **no. 4**, pp. 377–429.

DIBENEDETTO, M.D., SALDANHA, A. and SANGIOVANNI–VINCENTELLI, A. [1994], "Model matching for finite state machines," *Proc. IEEE Conf. on Decision and Control*, **vol. 3**, pp. 3117–3124.

GENG, X.J. and HAMMER J. [2004], "Asynchronous sequential machines: input/output control", *Proc. 12th Mediterranean Conf. on Control and Automation*, Kusadasi, Turkey, June 2004. [2005], "Input/output control of asynchronous sequential machines," *IEEE Trans. Automat. Contr.*, **vol. 50**, **no. 12**, pp. 1956-1970.

HAMMER, J. [1994], "On some control problems in molecular biology," *Proc. IEEE Conf. on Decision and Control*, 1994. [1995], "On the modeling and control of biological signal chains," *Proc. IEEE Conf. on Decision and Control*, 1995. [1996a], "On the corrective control of sequential machines," *Int. J. Contr.*, **vol. 65**, **no. 65**, pp. 249-276. [1996b], "On the control of incompletely described sequential machines," *Int. J. Contr.*, **vol. 63**, **no. 6**, pp. 1005-1028. [1997], "On the control of sequential machines with disturbances," *Int. J. Contr.*, **vol. 67**, **no. 3**, pp. 307-331.

KOHAVI, Z., [1970], *Switching and Finite Automata Theory*, McGraw-Hill Book Company, New York.

MURPHY, T.E., GENG, X.J. and HAMMER, J., [2002], "Controlling races in asynchronous sequential machines," *Proc. IFAC World Congress*, Barcelona, Spain, 2002. [2003], "On the control of asynchronous machines with races," *IEEE Trans. Automat. Contr.*, **vol. 48**, **no. 6**, pp. 1073-1081.

RAMADGE, P.J.G. and WONHAM, W.M., [1987], "Supervisory control of a class of discrete event processes," *SIAM J. Contr. Optim.*, **vol. 25**, **no. 1**, pp. 206-230.

THISTLE, J.G. and WONHAM, W.M., [1994], "Control of infinite behavior of finite automata," *SIAM J. Contr. Optim.*, **vol. 32**, **no. 4**, pp. 1075-1097.

VENKATRAMAN, N. and HAMMER, J., [2006a], "Stable realizations of asynchronous sequential machines with infinite cycles," *Proc. 2006 Asian Control Conf.,* Bali, Indonesia. [2006b], "Controllers for asynchronous machines with infinite cycles," *Proc. 17th Int. Symp. on Mathematical Theory of Networks and Systems*, Kyoto, Japan. [2006c], "On the control of asynchronous sequential machines with infinite cycles," *Int. J. Contr.*, **vol. 79**, **no. 7**, pp. 764-785.

YANG J.M. and HAMMER, J., [2007], "State feedback control of asynchronous sequential machines with adversarial inputs," International J. Control (to appear).