Fault Tolerance for Input Faults in a Class of Asynchronous Sequential Machines

Jung-Min Yang and Seong Woo Kwak

Abstract—In this paper, we address the problem of tolerating the adverse effects of input faults on the operation of a class of asynchronous sequential machines. Occurring to either the external input or the control input generated by the controller, input faults can cause unauthorized changes of the input value. In the framework of corrective control, we present the necessary and sufficient condition for the existence of an appropriate controller that invalidates the effects of input faults, while controlling the considered machine so as to match the stable state behavior of the closed-loop system to a reference model. An illustrative example is provided for demonstrating the proposed control scheme.

I. INTRODUCTION

As a novel automatic control theory for asynchronous sequential machines, corrective control has been successfully utilized to amend the stable state behavior of asynchronous machines with various deficiencies [1]–[3]. Whereas the structure of the corrective control system bears resemblance to traditional automatic control, its control law is somewhat different. Discrete mathematics and automata theory are involved in generating control inputs of corrective controllers and especially, asynchronous mechanisms [4] are used to materialize compensation of the closed-loop system in a desirable manner.

Among subjects of corrective control, fault tolerance is a major accomplishment that has been validated both in theoretical analysis and experimental studies. In rough terms, fault-tolerant corrective control is classified according to the type of considered faults. [5] and [6] generalize the preview work on the elimination of critical races [1] so as to control nondeterministic asynchronous machines. [7] and [8] present corrective controllers that detect and tolerate transient faults causing unwanted state transitions in asynchronous machines. The result of [8] is applied to controlling FPGA-based asynchronous digital systems in [9]. [7]–[9] tackle fault tolerance against transient faults for which the influence of faults lasts only for an instantaneous moment. In [10], [11], by contrast, the corrective controllers invalidate the adverse effects of permanent faults in which the characteristic of the faulty transition remains indefinitely. In [12], intermittent faults are considered in which unauthorized state transitions occur by fault and the effect of a fault persists for finite time after the initial occurrence. Finally, [13] presents a corrective control law that invalidates the influence of pre-programmed adversarial software agents. Other approaches on corrective control schemes including fault tolerance are found in [14]–[16] and the references therein.

In this paper, we propose a fault-tolerant corrective control scheme to tolerate input faults that cause unauthorized changes of input values. We suppose that not only the external input to the controller but also the control input generated by the controller is influenced by the input fault. Unless counteracted immediately, further change of the input would violate the desired behavior. Based on the corrective control scheme for model matching, we present necessary and sufficient conditions for the existence of a corrective controller that invalidates any input fault occurring to the asynchronous machine. The closed-loop system will be driven to follow a reference model as if no input fault occurs. Note that all the prior works on fault-tolerant corrective control [7]–[12] focus on state transition faults and do not consider detection and tolerance methodologies for input faults.

We first represent a modeling formalism for a class of asynchronous machines with input faults, and describe the basic control configuration. Then we address the reachability analysis on the considered machine and present the existence condition for a corrective controller that achieves fault tolerance against any input fault. The proposed controller adjusts the stable state behavior of the machine so as to match it to a reference model, while invalidating all the occurrences of unauthorized switches of input values. The proposed notion and examination of the controller existence are demonstrated in a case study using a synthetic asynchronous machine.

II. MODELING AND PROBLEM STATEMENT

A. Asynchronous Machines with Input Faults

The considered asynchronous sequential machine is input/state type in which the output is equal to the present state of the machine. We represent an input/state asynchronous machine $\Sigma$ as the following deterministic finite state machine.

$$\Sigma := (A, X, x_0, f)$$

where $A$ is the input set, $X$ is the set of $n$ states, $x_0 \in X$ is the initial state, and $f : X \times A \to X$ is the state transition function.

A state–input combination $(x, v) \in X \times A$ is termed valid if $f(x, v)$ is defined in $\Sigma$. A valid combination $(x, v)$ is divided into stable and transient combinations. If $f(x, v) = x$, $(x, v)$ is a stable combination with $x$ a stable state. On the other hand, if $f(x, v) \neq x$, it is a transient combination with $x$ a transient state. $x$ can be either stable or transient depending
on the present input value. Since no global synchronizing clock exists in asynchronous machines, $\Sigma$ responds with only the change of the external input and $\Sigma$ stays at a stable combination indefinitely unless the input value changes. In this paper, we assume that $f$ is a total function on $X \times A$, that is, every input in $A$ makes a valid combination with every state in $X$. This assumption does not lose much generality since if an input $v$ that is not valid with the present state $x$ is received, $\Sigma$ would be unresponsive. Hence we can regard $(x, v)$ as a stable combination. To elucidate this, define for $x \in X$ the following two subsets of $A$.

$$U(x) := \{v \in A | f(x, v) = x\}$$

$$T(x) := \{v \in A | f(x, v) \neq x\}.$$

$U(x)$ and $T(x)$ denote, respectively, the set of inputs that make stable and transient combinations with $x$. Clearly, $U(x)$ and $T(x)$ satisfy the following relations.

$$U(x) \cap T(x) = \emptyset$$

$$U(x) \cup T(x) = A.$$

Assume that $\Sigma$ has been staying at a stable state $x$ when the input changes to $v \in T(x)$ which makes a transient combination with $x$. $\Sigma$ then initiates a chain of transient transitions, say,

$$f(x, v) = x_1, f(x_1, v) = x_2, \ldots$$

during which the input $v$ remains unchanged. Assuming no infinite cycles, $\Sigma$ reaches a stable state $x_k$ such that

$$x_k = f(x_{k-1}, v) = f(x_k, v), \ \exists k < \infty,$$

i.e., $v \in U(x_k)$. $x_k$ is called the next stable state of $(x, v)$. Due to the absence of a synchronizing clock, the transient transitions lapse away instantaneously. Hence, from outer users’ viewpoint, only stable states are perceptible in the operation of $\Sigma$. To characterize this feature, we define the stable recursion function $s$ by [1]

$$s : X \times A \to X$$

$$s(x, v) := x_k,$$

where $x_k$ is the next stable state of $(x, v)$. A chain of transitions from one stable combination to another, as described by $s$, is called a stable transition. It is convenient to extend the domain of $s$ from $X \times A$ to $X \times A^*$ recursively, where $A^*$ is the set of non-empty strings made of characters in $A$. For $x \in X$ and \(v_1 v_2 \cdots v_k \in A^*\), we define

$$s(x, v_1 v_2 \cdots v_k) := s(s(x, v_1), v_2 \cdots v_k).$$

For two states $x, x' \in X$, $x'$ is said to be stably reachable from $x$ [1] if an input sequence $t \in A^+$ is found such that $x' = s(x, t)$ and $|t| \leq n - 1$ where $|t|$ is the length of $t$ and $n = \#X$ is the cardinality of $X$.

The input fault is modeled by a relation that maps an input value to a collection of faulty ones. In formal terms, we define a set $F(v) \subset A$ for an input $v \in A$ as follows (see also [17]).

**Definition 1.** Given $\Sigma = (A, X, x_0, f)$, the input fault for $v \in A$ is an unauthorized switch of the input value from $v$ to an element of $F(v) \subset A$. If $F(v) = \emptyset$, $v$ is a fault-free character.

Slightly abusing the terminology, we will use $F(v)$ when referring to the input fault happening at $v$. The input fault is attributed to a variety of malfunctions of the system. For instance, in the case of digital systems working in space, radiation-related errors such as single event upsets (SEU) [18] may switch the logic value of memory bits, which in turn may cause abrupt change of the input. Other reports on input faults and the modeling formalisms for fault events are found in the literature [19]–[21].

### B. Problem Statement

![Fig. 1. Corrective control system for tolerating input faults.](image)

Fig. 1 illustrates the structure of the corrective control system accommodating input faults. $\Sigma$ is the considered asynchronous machine and $C$ is the corrective controller that is implemented also as an asynchronous machine. The closed-loop system consisting of $C$ and $\Sigma$ is denoted by $\Sigma_r$. $\Sigma_r$, a reference model whose behavior must be matched by $\Sigma_r$, is described as

$$\Sigma_r = (A, X, x_0, s_r)$$

where $s_r$ is the stable recursion function of $\Sigma_r$. $C$ receives the external input $v \in A$ and state feedback values $x$ and $x_t$ from $\Sigma$ and $\Sigma_r$, respectively, to generate the control input $u \in A$. The control objective of $C$ is two folds as follows.

First, $C$ achieves model matching between $\Sigma$ and $\Sigma_r$. Here, matching means that the stable state input/output behavior of $\Sigma_r$ equals that of $\Sigma$. If $\Sigma_r$ and $\Sigma$ stay at the same stable state and if an identical external input enters each machine, they must transfer to the same next stable state. A key aspect is that the control behavior is validated only in terms of the stable states since the transient states are meaningless in asynchronous machines. For this reason, the input and state set and the initial state of $\Sigma_r$ must be the same as those of $\Sigma$ and only the stable recursion function $s_r$ of the reference model $\Sigma_r$ is given with no loss of generality.

Next, $C$ overcomes the effects of any input fault occurring to $\Sigma$. As marked in Fig. 1, the input fault may occur to either $v$ or $u$. In other words, the value of $v$ or $u$ may undergo unauthorized changes in the operation of $\Sigma_r$ from $v$ to one of $F(v)$ and from $u$ to one of $F(u)$, respectively (assuming both $F(v)$ and $F(u)$ are non-empty). If the fault is not recovered instantaneously, further change of the external input will drive $\Sigma_r$ to incorrect next stable states, thus violating matching between $\Sigma_r$ and $\Sigma$. By employing corrective control laws, we design $C$ so that $\Sigma_r$ can seem to maintain the desired input/state behavior despite occurrences of input faults.
To prohibit asynchronous machines from falling into unpredictable behaviors, the machines have to be constructed such that they comply with the principle of the fundamental mode operation [22], an operating policy that forbids the simultaneous change of two or more system variables. This policy helps to prevent uncertainties arising from simultaneous changes in two or more variables in the behavior of asynchronous machines. For $\Sigma_c$ to operate in fundamental mode, the following condition must be always valid. Note that this result is taken from the former studies [1], [7].

**Condition 1.** The closed-loop system $\Sigma_c$ of Fig. 1 operates in fundamental mode when all the following conditions are valid:

(i) Among $C$ and $\Sigma$, when one machine goes on transient transitions, the other must stay at a stable state.

(ii) The external input $v$ changes only while $C$ and $\Sigma$ are both at stable states.

(iii) The input fault occurs only when both $C$ and $\Sigma$ are at stable states.

Conditions (i) and (ii) are the design specifications that must be satisfied in the construction of the closed-loop system $\Sigma_c$. (iii) imposes a restriction on the occurrences of input faults, which are independent adversarial entities. One must suppose that input faults never happen when $\Sigma_c$ are in the midst of transient transitions. Nevertheless, as transient transitions of asynchronous machines occur very quickly, (iii) is not a burdensome requirement. Throughout this paper, we assume that $\Sigma_c$ always preserves the principle of the fundamental mode operation.

### III. Model Matching

Let us first address the existence condition and design procedure for a corrective controller for model matching between $\Sigma$ and $\Sigma_c$. We temporarily assume that no input fault occurs in the operation of $\Sigma_c$. In the former results [1]–[3], it is found that the existence condition is described by certain reachability properties of the machine and model. These properties can be characterized in terms of a numerical matrix, called the skeleton matrix, defined as follows.

**Definition 2.** Given $\Sigma = (A, X, x_0, f)$, let $X = \{x_1, \ldots, x_n\}$. The skeleton matrix $K(\Sigma)$ of $\Sigma$ is an $n \times n$ matrix of zeros and ones whose $(i, j)$ entry is

$$K_{ij}(\Sigma) = \begin{cases} 1 & \exists r \in A^+ \text{ such that } s(x_i, r) = x_j \\ 0 & \text{otherwise} \end{cases}$$

where $i, j \in \{1, \ldots, n\}$.

$K(\Sigma)$ shows in a compact way the stable reachability between any pair of states in $\Sigma$. The skeleton matrix $K(\Sigma_c)$ for $\Sigma_c$ is similarly defined. The existence condition for a corrective controller that realizes model matching between $\Sigma_c$ and $\Sigma$ is written as [1]–[3]

$$K(\Sigma_c) \preceq K(\Sigma)$$

where the inequality is valid entry by entry. The above relation means that the stable reachability of $\Sigma$ must be greater than or equal to that of $\Sigma_c$ for ensuring model matching corrective control.

We outline the process of corrective control for model matching provided that $K(\Sigma_c) \preceq K(\Sigma)$ is valid. This process will be also applied to constructing the module of the fault-tolerant controller. Referring to Fig. 1, we formulate $C$ as the following finite state machine; note that $\Sigma$ is the input/state machine whereas $C$ is the input/output machine that provides the output value which differs from the present state.

$$C = (X \times X \times A, A, \Xi, x_0, \phi, \eta)$$

where $X \times X \times A$ is the input set, $A$ is the output set, $\Xi$ is the state set, and $\xi_0 \in \Xi$ is the initial state. $\phi$ and $\eta$ are the recursion function and output function, respectively, with the mappings

$$\phi : \Xi \times X \times X \times A \rightarrow \Xi$$

$$\eta : \Xi \rightarrow A.$$

In the beginning, $C$ is at the initial state $\xi_0$. Assuming that model matching has been successful so far, suppose that both $\Sigma$ and $\Sigma_c$ reach the same stable state $x_i$ for which there exists a non-empty set $D(x_i, x_j) \subset A$ such that

$$D(x_i, x_j) = \{a \in A | s(x, a) = x_j \text{ and } s(x, a) \neq s(x, a)\}.$$

Any input in $D(x_i, x_j)$ would cause model mismatch in the transition from $x_i$ to $x_j$. Upon receiving the state feedback $x_i$, $C$ transfers to $\xi_0$, termed the transition state [3]. In the fundamental mode operation, an input change can occur only when the machine stays at a stable combination. Anticipating that an input character in $D(x_i, x_j)$ may enter the system, $C$ prepares the correction behavior at $\xi_0$. In order to realize the latter functionality, we assign $\phi$ and $\eta$ at $\xi_0$ and $\xi_1$ as follows.

$$\phi(\xi_0, x, x, v) = \xi_0 \forall (x, x, v) \in X \times X \times A \setminus \{(x, x)\} \times U(x_i) \phi(\xi_0, x_i, v, v) = \xi_1 \forall v \in U(x_i).$$

(1)

Note that the first character in the input variables of $\phi$ denotes the state feedback from $\Sigma$. Since no actual control is conducted at either $\xi_0$ or $\xi_1$, $C$ relays the external input $v$ to the control input $u$ without modification:

$$\eta(\xi_0) = v$$

$$\eta(\xi_1) = v.$$  

(2)

If the external input $v$ changes to a character that invokes no model mismatch, $C$ will go back to $\xi_0$. On the other hand, if $v$ changes to a character $a \in D(x_i, x_j)$, the next behavior of $\Sigma$ would violate the desired input/state specification if not corrected. By Definition 2, $s_i(x, a) = x_j$ implies $K_{ij}(\Sigma_c) = 1$. Since $K(\Sigma_c) \preceq K(\Sigma)$ by assumption, $K_{ij}(\Sigma) = 1$ leads to $K_{ij}(\Sigma) = 1$ (every entry of the skeleton matrix is either zero or one) and hence there exists an input sequence $t := u_1 u_2 \cdots u_k \in A^+$ ($k \leq n - 1$) such that $x_j = s(x, t)$. We denote by $z_1, \ldots, z_{k-1}$ in $X$ all the intermediate stable states $\Sigma$ passes through with $t$, that is,

$$z_i = s(z_{i-1}, u_i)$$

$$z_i = s(z_{i-1}, u_i), \; i = 1, \ldots, k$$

(3)

$$z_0 := x_i$$

$$z_{k+1} := x_j.$$

Note that between the adjacent stable states $z_i$ and $z_{i+1}$, $\Sigma$ may pass through some transient states. Asynchrony and fundamental mode operations of $\Sigma_c$ make these stable transitions $(z_0, u_1), (z_1, u_2), \ldots, (z_{k-1}, u_k)$ show transient characteristics temporarily by inserting $k$ auxiliary states of the controller.
The next behavior of C is determined by \( v' \). First, assume \( v' \in U(x_i). \) Since model matching between \( \Sigma_i \) and \( \Sigma_r \) is supposed to be maintained, the present state of \( \Sigma_i \) is also \( x_i. \) In this case, the switched input \( v' \) does not violate the desired specification. Hence \( C \) delivers \( v' \) to the control input \( u \) and no state transition is induced either in \( C \) or \( \Sigma. \)

\[
\phi(\xi, x_i, x_i, v) = \xi_i \\
\eta(\xi_i) = v' \quad \forall v' \in F(v) \cap U(x_i).
\]

Next, assume \( v' \in T(x_i). \) With no knowledge of the fault occurrence, \( C \) must determine the next operation only in terms of whether or not \( v' \) would cause model mismatch. If \( s(x_i, v') = s_i(x_i, v'), v' \notin D(x_i, x_j) \) for all \( j \in \{1, \ldots, n\}. \) Thus \( C \) does not execute any control action; it just relays \( v' \) to the control input channel, i.e., \( u = v' \) as before. When \( \Sigma \) reaches the next stable state \( s(x_i, v') = s_i(x_i, v') \) is delivered to \( C \) as the state feedback. \( C \) then compares it with the state feedback \( x_i \) coming from \( \Sigma_r. \) Since \( v' \) occurs by fault, the external input to \( \Sigma \) is still \( v \) and \( \Sigma \) stays at \( x_i. \) When the two state feedback values are found to be different, an occurrence of the input fault is perceived, and \( C \) should initiate another correction procedure from \( s(x_i, v') \) to \( x_i \) immediately to maintain model matching. The condition for making a correction trajectory from \( s(x_i, v') \) to \( x_i \) is similar to the case of model matching and is described as follows.

If \( v \in U(x_i) \) and \( s(x_i, v') = s_i(x_i, v'), \)
\[ \exists v' \in A^+ \text{ such that } s(s(x_i, v'), t') = x_i, \]

namely, the original state \( x_i \) must be stably reachable from the deviated state \( s(x_i, v') \) in \( \Sigma. \)

On the other hand, if \( s(x_i, v') \neq s_i(x_i, v'), v' \) is an input character that causes model mismatch at the state \( x_i. \) Let \( s_i'(x_i, v') := x_i. \) Then \( v' \notin D(x_i, x_j) \) and \( K_i(\Sigma_r) = 1. \) Since \( K_i(\Sigma) \leq K(\Sigma) \) by assumption, \( K_i(\Sigma) = 1 \) leads to \( K_i(\Sigma) \) and \( C \) already materializes a correction trajectory from \( x_i \) to \( x_j \) using an input sequence \( t' \in A^+ \) such that \( s(x_i, t) = x_j. \) Upon receiving \( v' \), \( C \) initiates the correction procedure that takes \( \Sigma \) toward the goal state \( s(x_i, v'). \) When \( \Sigma \) reaches \( s(x_i, v'), C \) compares the two state feedback values from \( \Sigma \) and \( \Sigma_i. \) The rest of the procedure equals the former case. \( C \) executes another correction procedure from \( s(x_i, v') \) to \( x_i. \)

The reachability condition needed to realize this control is written as

If \( v \in U(x_i) \) and \( s(x_i, v') \neq s_i(x_i, v'), \)
\[ \exists v' \in A^+ \text{ such that } s(s(x_i, v'), t') = x_i, \]

The design procedure is almost identical to (1)–(6). In fact, if \( D(x_j, x_i) \subseteq \emptyset \), \( C \) already has the corresponding correction trajectory from \( x_j \) to \( x_i. \) It is efficient to use this trajectory instead of adding another one. To this end, let \( \xi_i \in \Sigma \) be the first auxiliary state of \( C \) that makes the correction trajectory from \( x_j \) to \( x_i. \) We assign \( \phi \) and \( \eta \) as follows.

\[
\phi(\xi, x_i, x_i, v') = \xi_i \\
v' \in F(v) \cap T(x_i) \text{ and } x_j = s_i(x_i, v').
\]

After reaching \( \xi_i, C \) continues the recursive operation as described in (5) and (6).
Note that for all $v' \in F(v) \cap T(x_i)$, conditions (7) and (8) can be combined into

$$\exists t(v') \in A^+ \text{ such that } s(s_i(x_i, v'), t(v')) = x_i.$$  \hspace{1cm} (9)

### B. Faults at the Control Input

Let us consider the problem of tolerating input faults occurring to the control input $u$. Unlike the case of input faults occurring to the external input $v$, $C$ can diagnose the occurrence of this fault since the unauthorized state transition is easily detected by observing the change of the state feedback $x$ while the external input $v$ remains unchanged.

More specifically, assume that both $\Sigma$ and $\Sigma_u$ have been staying at a stable combination $(x_i, u)$ with $F(u) \neq \emptyset$ when the input fault occurs to $u$, causing an unwanted switch of the control input from $u$ to $u' \in F(u)$. The fault occurrence is identified when the state feedback is observed to change from $x_i$ to $s(x_i, u')$ while the external input and the state feedback from $\Sigma_u$ remain fixed. Of course, if $u' \in U(x_i)$, the state feedback remains the same. In this case, the fault is latent in that it does not incur any change to the input or state.

Assume now $u' \in T(x_i)$. The condition for countering the unauthorized state transition from $x_i$ to $s(x_i, u')$ is similar to that used in tolerating the faults to the external input. The original state $x_i$ must be stably reachable from the deviated state $s(x_i, u')$. In formal terms, the latter is written as

$$\exists t(u') \in A^+ \text{ such that } s(s(x_i, u'), t(u')) = x_i.$$  \hspace{1cm} (10)

Using the results (9) and (10), we now address the condition for the existence of the fault-tolerant controller that achieves model matching with respect to $\Sigma$ as well as counteracts any input fault in $\Sigma$ that occurs to either the external input $v$ or the control input $u$. The following theorem is the main result of this paper.

**Theorem 1.** Given $\Sigma = (A, X, x_0, f)$ with $X = \{x_1, \ldots, x_n\}$, let $\Sigma_u = (A, X, x_0, f_x)$ be the reference model, and let $K(\Sigma)$ and $K(\Sigma_u)$ be the skeleton matrix of $\Sigma$ and $\Sigma_u$, respectively. Suppose that $A$ has at least one character $\nu$ such that $F(\nu) \neq \emptyset$. Then, there exists a corrective controller $C$ of Fig. 1 that matches the stable state behavior of $\Sigma_u$ to that of $\Sigma$ while invalidating the influence by any input faults if and only if the following conditions are held true.

1. $K(\Sigma_u) \leq K(\Sigma)$.
2. $\forall x_i, i = 1, \ldots, n, \forall v \in A$ with $v \in U(x_i)$ and $F(v) \neq \emptyset$,

$$\exists t(v') \in A^+ \text{ such that } s(s_i(x_i, v'), t(v')) = x_i.$$  \hspace{1cm} (11)

The above theorem means that if an input fault occurs to a stable state, fault tolerance against the fault is possible if and only if the original state is stably reachable from the deviated state in both $\Sigma$ and $\Sigma_u$, as described in (b). If this condition and that of model matching (item (a)) are valid, we can design a corrective controller that materializes both fault tolerance and model matching. Employing the basic corrective controller module for model matching described in (1)–(6), we can construct a corrective controller that realizes fault tolerance against the fault input occurring to each combination $(x_i, v)$, namely $C(x_i, v)$. The overall controller $C$ is obtained by adding all the controller modules $C(x_i, v)$’s to $C$ using join operation (refer to [3] for a detailed algorithm for assembling corrective controller modules).

### V. Example

Consider an input/state asynchronous sequential machine $\Sigma = (A, X, x_0, f)$ whose state flow diagram is shown in Fig. 2. Here, $A = \{a, b, c, d\}$, $X = \{x_1, x_2, x_3, x_4\}$, and $x_0 = x_1$. An examination of Fig. 2 shows that the corresponding stable state behavior of $\Sigma$ is given by Fig. 3. Since the next stable state of the pair $(x_1, d)$ is $x_1$, $s(x_1, d) = x_1$ and the corresponding stable transition is marked in Fig. 3. $(x_4, b)$ is another transient pair for which the stable transition is different from the transient transition $(f(x_4, b) = x_1$ but $s(x_4, b) = x_2)$. Among the input alphabet $A$, we assume the following fault scenario.

$$F(b) = \{a, d\}$$
$$F(v) = \emptyset, \forall v = \{a, c, d\},$$

that is, the input $b$ may be switched to one of $a$ and $d$ by the input fault $F(b)$ and the other input characters are fault-free. The reference model $\Sigma_u = (A, X, x_0, f_x)$ that must be matched by the closed-loop system $\Sigma$ is shown in Fig. 4.

Let us first investigate the possibility of model matching between $\Sigma_u$ and $\Sigma$. The skeleton matrices $K(\Sigma)$ and $K(\Sigma_u)$ are derived as

$$K(\Sigma) = \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \end{pmatrix}$$
$$K(\Sigma_u) = \begin{pmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 0 & 0 & 1 & 0 \\ 1 & 1 & 1 & 1 \end{pmatrix}$$

Refer to the former works on corrective control [1], [3], [5] for the formal steps of computing the skeleton matrix. Since $K(\Sigma_u) \leq K(\Sigma)$, a model matching corrective controller $C$ exists in the structure of Fig. 1.

Next, we examine the possibility of fault-tolerant control against $F(b)$. In Fig. 3, $b$ makes a stable combination with $x_2$ and $x_3$, that is, $b \in U(x_2)$ and $b \in U(x_3)$. Further, we have

$$F(b) \cap T(x_2) = \{a\}$$
$$F(b) \cap T(x_3) = \{d\}.$$

We first investigate the pair $(x_2, b)$. As $F(b) \cap T(x_2) = \{a\}$, the only deviated state $\Sigma$ may reach from $x_2$ as a result of the input fault is $s(x_2, a) = x_1$. But we already know that $x_2$ is stably reachable from $x_1$, i.e., $K_{12}(\Sigma) = 1$. Hence, fault tolerance against $F(b)$ at the state $x_2$ is possible. Secondly, we consider the pair $(x_3, b)$. From $F(b) \cap T(x_3) = \{d\}$, it follows that the deviated state is $s(x_3, d) = x_1$. Since $K_{13}(\Sigma) = 1$, fault tolerance against $F(b)$ at the state $x_3$ is also possible. This analysis implies that condition (b) of Theorem 1 is satisfied by $\Sigma$. Thus, a fault-tolerant controller exists.
that invalidates the input faults $F(b)$, while realizing model matching between $\Sigma_c$ and $\Sigma_r$.

Fig. 2. State flow diagram of $\Sigma$.

Fig. 3. State flow diagram of $\Sigma$ in the stable states.

Fig. 4. State flow diagram of $\Sigma_r$.

VI. SUMMARY

We have presented a corrective control scheme for asynchronous sequential machines with input faults. We have focused our concern on proposing a fault-tolerant control law for input faults using only state feedback values from the machine and the model. No observer is employed in the proposed control architecture. Necessary and sufficient conditions for the existence of the controller are presented. It has been found that to tolerate any input fault, both the machine and the model must be able to reach the original state from the deviated state. The design procedure for a controller has been outlined based on the basic corrective controller module. To demonstrate the applicability of the proposed scheme, the procedure of checking the controller existence is addressed in the illustrative example. An application of the proposed fault-tolerant control scheme to real-world asynchronous sequential machines is under way as a further study.

REFERENCES