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 Σ 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 Σ. 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

This research was supported in part by Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Science, ICT and future Planning (No. 2015R1A2A1A15054026), in part by Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education (No. NRF–2015R1D1A1A01056764), and in part by the Bio & Medical Technology Development Program of the National Research Foundation (NRF) funded by the Korean government (MSIP) (No. NRF–2015M3A9A7067220).

J.–M. Yang is with the School of Electronics Engineering, Kyungpook National University, Daegu, 41566, Republic of Korea. E-mail: jmyang@ee.knu.ac.kr.

S. W. Kwak is with the Department of Electronic Engineering, Keimyung University, Daegu, 42601, Republic of Korea. E-mail: ksw@kmu.ac.kr (corresponding author).

on the present input value. Since no global synchronizing clock exists in asynchronous machines, Σ responds with only the change of the external input and Σ 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, Σ 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.

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

Assume that Σ has been staying at a stable state x when the input changes to $v \in T(x)$ that makes a transient combination with x. Σ 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, Σ 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's viewpoint, only stable states are perceptible in the operation of Σ. 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 $|t| \leqslant n - 1$ 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 $F(v) = \varnothing$, 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

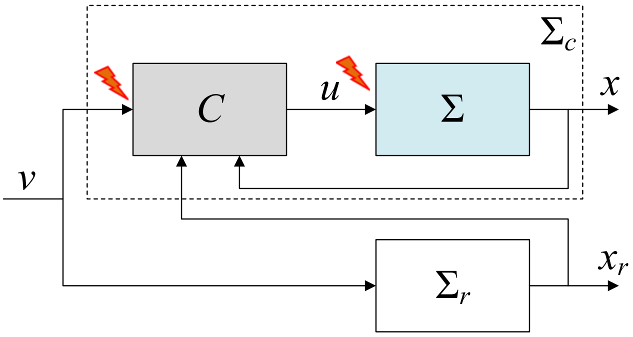

Fig. 1. Corrective control system for tolerating input faults.

Fig. 1 illustrates the structure of the corrective control system accommodating input faults. Σ 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 Σ is denoted by Σ_c. Σ_r, a reference model whose behavior must be matched by Σ_c, is described as

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

where s_r is the stable recursion function of Σ_r. C receives the external input $v \in A$ and state feedback values x and x_r from Σ and Σ_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 Σ_c and Σ_r. Here, *matching* means that the stable state input/ouput behavior of Σ_c equals that of Σ_r. If Σ_c and Σ_r 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 Σ_r must be the same as those of Σ and only the stable recursion function s_r of the reference model Σ_r is given with no loss of generality.

Next, C overcomes the effects of any input fault occurring to Σ. 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 Σ_c 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 Σ_c to incorrect next stable states, thus violating matching between Σ_c and Σ_r. By employing corrective control laws, we design C so that Σ_c 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 Σ_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 Σ_c of Fig. 1 operates in fundamental mode when all the following conditions are valid:*

(i) *Among C and Σ, 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 Σ are both at stable states.*
(iii) *The input fault occurs only when both C and Σ are at stable states.*

Conditions (i) and (ii) are the design specifications that must be satisfied in the construction of the closed-loop system Σ_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 Σ_c are in the middle 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 Σ_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 Σ_c and Σ_r. We temporarily assume that no input fault occurs in the operation of Σ_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 Σ is an $n \times n$ matrix of zeros and ones whose (i, j) entry is*

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

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

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

$$K(\Sigma_r) \leqslant K(\Sigma)$$

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

We outline the process of corrective control for model matching provided that $K(\Sigma_r) \leqslant K(\Sigma)$ 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 Σ is the input/state machine whereas C is the input/output machine that provides the output value which differs from the present state.

$$C = (X \times X \times A, A, \Xi, \xi_0, \phi, \eta)$$

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

$$\begin{aligned} &\phi : \Xi \times X \times X \times A \to \Xi \\ &\eta : \Xi \to A. \end{aligned}$$

In the beginning, C is at the initial state ξ_0. Assuming that model matching has been successful so far, suppose that both Σ and Σ_r reach the same stable state x_i for which there exists a non-empty set $D(x_i, x_j) \subset A$ such that

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

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 ξ_t, 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 ξ_t. In order to realize the latter functionality, we assign ϕ and η at ξ_0 and ξ_t as follows.

$$\begin{aligned} &\phi(\xi_0, x, x, v) = \xi_0 \ \forall (x, x, v) \in X \times X \times A \backslash \{(x_i, x_i)\} \times U(x_i) \\ &\phi(\xi_0, x_i, x_i, v) = \xi_t \ \forall v \in U(x_i). \end{aligned} \tag{1}$$

Note that the first character in the input variables of ϕ denotes the state feedback from Σ. Since no actual control is conducted at either ξ_0 or ξ_t, C relays the external input v to the control input u without modification:

$$\begin{aligned} \eta(\xi_0) &= v \\ \eta(\xi_t) &= v. \end{aligned} \tag{2}$$

If the external input v changes to a character that invokes no model mismatch, C will go back to ξ_0. On the other hand, if v changes to a character $a \in D(x_i, x_j)$, the next behavior of Σ would violate the desired input/state specification if not corrected. By Definition 2, $s_r(x_i, a) = x_j$ implies $K_{ij}(\Sigma_r) = 1$. Since $K(\Sigma_r) \leqslant K(\Sigma)$ by assumption, $K_{ij}(\Sigma_r) = 1$ leads to $K_{ij}(\Sigma) = 1$ (every entry of the skeleton matrix is either zero or one) and hence there exists an input sequence $t := u_1 u_2 \cdots u_k \in A^+$ $(k \leqslant n - 1)$ such that $x_j = s(x_i, t)$. We denote by $z_1, \ldots, z_{k-1} \in X$ all the intermediate stable states Σ passes through with t, that is,

$$\begin{aligned} &z_i = s(z_{i-1}, u_i) \\ &z_i = s(z_i, u_i), \ i = 1, \ldots, k \\ &z_0 := x_i \\ &z_k := x_j. \end{aligned} \tag{3}$$

Note that between the adjacent stable states z_i and z_{i+1}, Σ may pass through some transient states. Asynchrony and fundamental mode operations of Σ_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, termed $\xi_1, \ldots, \xi_k \in \Xi$, into the correction trajectory. As soon as C receives the external input a, it transfers to ξ_1, the first auxiliary state, and provides Σ with the first control input character u_1. To this end, set ϕ and η as

$$\begin{aligned} \phi(\xi_t, x_i, x_i, a) &= \xi_1 \ \forall a \in D(x_i, x_j) \\ \phi(\xi_1, x_i, x_i, a) &= \xi_1 \ \forall a \in D(x_i, x_j). \\ \eta(\xi_1) &= u_1. \end{aligned} \tag{4}$$

In response to u_1, Σ moves from x_i to $z_1 = s(x_i, u_1)$, the first intermediate stable state. Receiving the state feedback z_1, C in turn transfers to the second auxiliary state ξ_2 and generates the second control input u_2. In response to u_2, Σ moves to the second intermediate stable state $z_2 = s(z_1, u_2)$, and so on. This iterative procedure continues for k steps. The following assignment of ϕ and η realizes this operation.

$$\begin{aligned} \phi(\xi_h, x_i, z_h, a) &= \xi_{h+1} \\ \phi(\xi_{h+1}, x_i, z_h, a) &= \xi_{h+1} \\ \eta(\xi_h) &= u_h \\ h &= 1, \ldots, k-1. \end{aligned} \tag{5}$$

Finally, at ξ_k, Σ reaches the desired next stable state x_j. To preserve the principle of the fundamental mode operation, we design C so that it receives the state feedback from Σ_r only after reaching the final state ξ_k. Receiving x_j from Σ_r, C returns to ξ_0.

$$\begin{aligned} \phi(\xi_k, x_i, x_j, a) &= \xi_k \\ \phi(\xi_k, x_j, x_j, a) &= \xi_0 \\ \eta(\xi_k) &= u_k. \end{aligned} \tag{6}$$

Due to the lack of no global synchronizing clock, all stable transitions and the interactions between C and Σ can be executed very fast. Therefore, the closed-loop system Σ_c seems to transfer from x_i directly to x_j in response to any input $a \in D(x_i, x_j)$. Note that the foregoing design solves model matching only for the stable transition from x_i to x_j. The controller modules for other stable transitions can be made in a similar way. The overall model matching controller is accomplished by combining each controller module.

IV. Fault-Tolerant Corrective Control

A. Faults at the External Input

We now consider the problem of tolerating input faults occurring to the external input v. It is supposed that the reachability condition $K(\Sigma_r) \leqslant K(\Sigma)$ is valid so that the model matching corrective controller C is implemented in front of Σ as shown in Fig. 1.

Assume that Σ (and Σ_r) has been staying at a stable combination (x_i, v) where $v \in U(x_i)$ and $F(v) \neq \varnothing$, i.e., v is not fault-free, and that C has been staying at the transition state ξ_t. Assume further that an input fault happens to v, causing an unauthorized switch to a character $v' \in F(v)$. Since C is not equipped with any fault detection module, it cannot discriminate between an occurrence of the input fault and the normal transmission of a new input character. In this study, we propose a policy that every change of the external input is first interpreted as the transmission of a new input character. Note that this policy is preferable because the rate of fault occurrences is usually much less than that of the inflow of input characters.

The next behavior of C is determined by v'. First, assume $v' \in U(x_i)$. Since model matching between Σ_c and Σ_r is supposed to be maintained, the present state of Σ_r 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 Σ.

$$\begin{aligned} \phi(\xi_t, x_i, x_i, v') &= \xi_t \\ \eta(\xi_t) &= v' \ \forall v' \in F(v) \cap U(x_i). \end{aligned}$$

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 $s(x_i, v') = s_r(x_i, v')$, $v' \notin D(x_i, x_j)$ 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 Σ reaches the next stable state $s(x_i, v')$, $s(x_i, v')$ is delivered to C as the state feedback. C then compares it with the state feedback x_r coming from Σ_r. Since v' occurs by fault, the external input to Σ_r is still v and Σ_r 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.

$$\begin{aligned} &\text{If } v \in U(x_i), v' \in T(x_i) \text{ and } s(x_i, v') = s_r(x_i, v'), \\ &\exists t' \in A^+ \text{ such that } s(s(x_i, v'), t') = x_i, \end{aligned} \tag{7}$$

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

On the other hand, if $s(x_i, v') \neq s_r(x_i, v')$, v' is an input character that causes model mismatch at the state x_i. Let $s_r(x_i, v') := x_j$. Then $v' \in D(x_i, x_j)$ and $K_{ij}(\Sigma_r) = 1$. Since $K(\Sigma_r) \leqslant K(\Sigma)$ by assumption, $K_{ij}(\Sigma_r) = 1$ leads to $K_{ij}(\Sigma) = 1$ 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 Σ toward the goal state $s_r(x_i, v')$. When Σ reaches $s_r(x_i, v')$, C compares the two state feedback values from Σ and Σ_r. The rest of the procedure equals the former case. C executes another correction procedure from $s_r(x_i, v')$ to x_i. The reachability condition needed to realize this control is written as

$$\begin{aligned} &\text{If } v \in U(x_i), v' \in T(x_i) \text{ and } s(x_i, v') \neq s_r(x_i, v'), \\ &\exists t' \in A^+ \text{ such that } s(s_r(x_i, v'), t') = x_i. \end{aligned} \tag{8}$$

The design procedure is almost identical to (1)–(6). In fact, if $D(x_j, x_i) \neq \varnothing$, 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 $\xi'_1 \in \Xi$ be the first auxiliary state of C that makes the correction trajectory from x_j to x_i. We assign ϕ and η as follows.

$$\begin{aligned} \phi(\xi_t, x_i, x_j, v') &= \xi'_1 \\ v' \in F(v) \cap T(x_i) &\text{ and } x_j = s_r(x_i, v'). \end{aligned}$$

After reaching ξ'_1, 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

$$\begin{aligned} &\text{If } v \in U(x_i), \forall v' \in F(v) \cap T(x_i), \\ &\exists t(v') \in A^+ \text{ such that } s(s_r(x_i, v'), t(v')) = x_i. \end{aligned} \tag{9}$$

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 Σ and Σ_r 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 Σ_r 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 counteracting 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

$$\begin{aligned} &\text{If } u \in U(x_i), \forall u' \in F(u) \cap T(x_i), \\ &\exists t(u') \in A^+ \text{ such that } s(s(x_i, u'), t(u')) = x_i. \end{aligned} \tag{10}$$

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 Σ as well as counteracts any input fault in Σ 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 $\Sigma_r = (A, X, x_0, s_r)$ be the reference model, and let $K(\Sigma)$ and $K(\Sigma_r)$ be the skeleton matrix of Σ and Σ_r, respectively. Suppose that A has at least one character v such that $F(v) \neq \emptyset$. Then, there exists a corrective controller C of Fig. 1 that matches the stable state behavior of Σ_c to that of Σ_r while invalidating the influence by any input faults if and only if the following conditions are held true.*

(a) $K(\Sigma_r) \leqslant K(\Sigma)$.

(b) $\forall x_i$, *$i = 1, \ldots, n$, and $\forall v \in A$ with $v \in U(x_i)$ and $F(v) \neq \emptyset$,*

$$\begin{aligned} &\forall v' \in F(v) \cap T(x_i), \\ &\exists t(v') \in A^+ \text{ such that } s(s_r(x_i, v'), t(v')) = x_i \\ &\exists t'(v') \in A^+ \text{ such that } s(s(x_i, v'), t'(v')) = x_i. \end{aligned}$$

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 Σ and Σ_r, 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 Σ is given by Fig. 3. Since the next stable state of the pair (x_3, d) is x_1, $s(x_3, d) = x_1$ 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.

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

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 $\Sigma_r = (A, X, x_0, s_r)$ that must be matched by the closed-loop system Σ_c is shown in Fig. 4.

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

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

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

Refer to the former works on corrective control [1], [3], [5] for the formal steps of computing the skeleton matrix. Since $K(\Sigma_r) \leqslant K(\Sigma)$, 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

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

We first investigate the pair (x_2, b). As $F(b) \cap T(x_2) = \{a\}$, the only deviated state Σ 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 Σ. Thus, a fault-tolerant controller exists that invalidates the input faults $F(b)$, while realizing model matching between Σ_c and Σ_r.

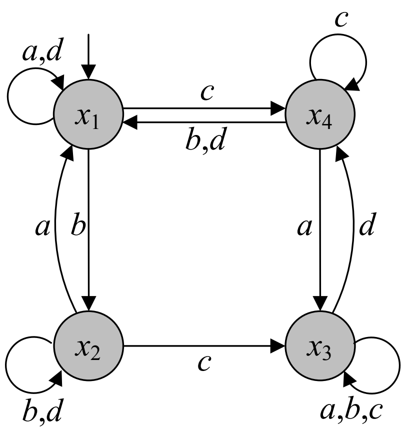

Fig. 2. State flow diagram of Σ.

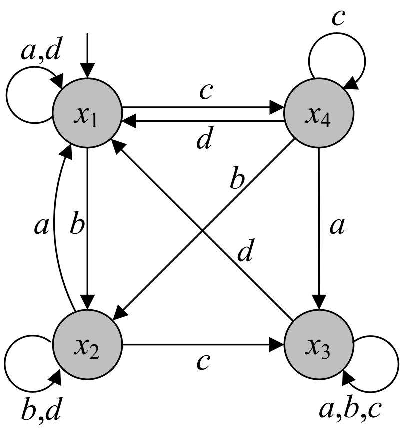

Fig. 3. State flow diagram of Σ in the stable states.

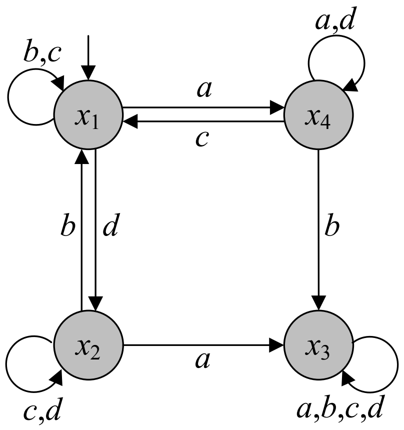

Fig. 4. State flow diagram of Σ_r.

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

[1] T. E. Murphy, X. Geng, and J. Hammer, “On the control of asynchronous machines with races,” *IEEE Trans. Autom. Control*, vol. 48, no. 6, pp. 1073–1081, 2003.

[2] X. Geng and J. Hammer, “Input/output control of asynchronous sequential machines,” *IEEE Trans. Autom. Control*, vol. 50, no. 12, pp. 1956–1970, 2005.

[3] N. Venkatraman and J. Hammer, “On the control of asynchronous sequential machines with infinite cycles,” *Int. J. Control*, vol. 79, no. 7, pp. 764–785, 2006.

[4] J. Sparsø and S. Furber, *Principles of Asynchronous Circuit Design — A Systems Perspective*, Kluwer Academic Publishers, 2001.

[5] J. Peng and J. Hammer, “Input/output control of asynchronous sequential machines with races,” *Int. J. Control*, vol. 83, no. 1, pp. 125–144, 2010.

[6] J. Peng and J. Hammer, “Bursts and output feedback control of non-deterministic asynchronous sequential machines,” *Euro. J. Control*, vol. 18, no. 3, pp. 286–300, 2012.

[7] J.–M. Yang, “Corrective control of asynchronous sequential machines in the presence of adversarial input,” *IET Control Theory Appl.*, vol. 2, no. 8, pp. 706–716, 2008.

[8] J.–M. Yang, “Corrective control of input/output asynchronous sequential machines with adversarial inputs,” *IEEE Trans. Autom. Control*, vol. 55, no. 3, pp. 755–761, 2010.

[9] J.–M. Yang and S. W. Kwak, “Output feedback control of asynchronous sequential machines with disturbance inputs,” *Inf. Sci.*, vol. 259, pp. 87–99, 2014.

[10] J.–M. Yang and S. W. Kwak, “Fault diagnosis and fault-tolerant control of input/output asynchronous sequential machines,” *IET Control Theory Appl.*, vol. 6, no. 11, pp. 1682–1689, 2012.

[11] J.–M. Yang, “Fault-tolerant control of a class of asynchronous sequential machines with permanent faults,” *Automatica*, vol. 50, no. 3, pp. 989–993, 2014.

[12] J.–M. Yang and S. W. Kwak, “On diagnosing intermittent faults in input/output asynchronous sequential machines,” *Latest Trends in Circuits, Automatic Control and Signal Processing*, pp. 110–115, 2012.

[13] J. Hammer, “Automatic defensive control of asynchronous sequential machines,” *Int. J. Control*, vol. 88, 2015, in press, doi:10.1080/00207179.2015.1064547.

[14] J.–M. Yang, T. Xing, and J. Hammer, “Adaptive control of asynchronous sequential machines with state feedback,” *Euro. J. Control*, vol. 18, no. 6, pp. 503–527, 2012.

[15] X. Xu and Y. Hong, “Matrix approach to model matching of asynchronous sequential machines,” *IEEE Trans. Autom. Control*, vol. 58, no. 11, pp. 2974–2979, 2013.

[16] J.–M. Yang, “Corrective controllers with switching capability for modeling matching of input/state asynchronous machines,” in *Proc. Int. MultiConf. Engineers and Computer Scientists*, vol. 2, 2015.

[17] Y. Nke and J. Lunze, “A fault modeling approach for input/output automata,” in *Proc. the 18th IFAC World Congress*, pp. 8657–8662, 2011.

[18] L. Sterpone and M. Violante, “Analysis of the robustness of the TMR-architecture in SRAM-based FPGAs,” *IEEE Trans. Nucl. Sci.*, vol. 53, no. 5, pp. 1545–1549, 2005.

[19] C. M. Krishina and K. G. Shin, *Real-Time Systems*, McGraw-Hill: New York, 1997.

[20] J. Lunze and J. Schröder, “Sensor and actuator fault diagnosis of systems with discrete inputs and outputs,” *IEEE Trans. Syst. Man. Cybern. B*, vol. 34, no. 2, pp. 1096–1107, 2004.

[21] A. Paoli, M. Sartini, and S. Lafortune, “Active fault tolerant control of discrete event systems using online diagnostics,” *Automatica*, vol. 47, no. 4, pp. 639–649, 2011.

[22] Z. Kohavi and N. K. Jha, *Switching and Finite Automata Theory*, 3rd ed., Cambridge University Press: Cambridge, UK, 2010.