applied sciences

MDPI

Article

State-Burst Feedback Control for Fault Recovery of Input/State Asynchronous Sequential Machines

Jung-Min Yang [1], Seong-Jin Park [2,*] and Seong Woo Kwak [3,*]

[1] School of Electronics Engineering, Kyungpook National University, Daegu 41566, Korea; jmyang@ee.knu.ac.kr
[2] Department of Electrical and Computer Engineering, Ajou University, Suwon 16499, Korea
[3] Department of Control and Instrumentation Engineering, Pukyong National University, Busan 48513, Korea
* Correspondence: parksjin@ajou.ac.kr (S.-J.P.); ksw@pknu.ac.kr (S.W.K.); Tel.: +82-31-219-2659 (S.-J.P.); +82-51-629-6325 (S.W.K.)

Abstract: Static corrective controllers are more efficient than dynamic ones since they consist of only logic elements, whereas their existence conditions are more restrictive. In this paper, we present a static corrective control scheme for fault diagnosis and fault tolerant control of input/state asynchronous sequential machines (ASMs) vulnerable to transient faults. The design flexibility of static controllers is enlarged by virtue of using a diagnoser and state bursts. Necessary and sufficient conditions for the existence of a diagnoser and static fault tolerant controller are presented, and the process of controller synthesis is addressed based on the derived condition. Illustrative examples on practical ASMs are provided to show the applicability of the proposed scheme.

Keywords: asynchronous sequential machines (ASMs); static corrective control; fault tolerant control; diagnoser; state bursts

Citation: Yang, J.-M.; Park, S.-J.; Kwak, S.W. State-Burst Feedback Control for Fault Recovery of Input/State Asynchronous Sequential Machines. *Appl. Sci.* **2021**, *11*, 9790. https://doi.org/10.3390/app11219790

Academic Editor: Mohsen Soltani

Received: 11 August 2021
Accepted: 15 October 2021
Published: 20 October 2021

1. Introduction

Aiming at compensating for the stable-state behavior of asynchronous sequential machines (ASMs), corrective control theory has been extensively studied in both theoretical [1–5] and experimental studies [6–8]. While the performance of corrective control is remarkable, especially in fault tolerant control against various kinds of faults [9–11], most of the developed controllers are dynamic ones having the form of ASMs with states, which need significant resource usage [7,8].

Static corrective controllers [12,13] are a promising alternative to dynamic ones. Consisting of only combinational logic with no states, static controllers are superior to dynamic ones in terms of not only resource usage, but also robustness against exogenous disturbances, as memory elements representing system states are frequently offended by faults [14,15]. On the other hand, they have a drawback in that their existence conditions are more restrictive than dynamic ones, due to the absence of the controller's states. In this paper, a novel methodology of static fault tolerant control is proposed for ASMs subject to transient faults causing unauthorized state transitions. We address the existence condition and synthesis procedure for a static controller that realizes immediate fault recovery.

Compared with the prior work [12,13], this study has the following contributions. First, we show that static corrective control can be adopted to solve various control problems for ASMs. The objective in [12,13] was model matching, namely, refining the stable-state behavior of the closed-loop system to that of a reference model. The present work extends that of [12,13] so as to achieve fault diagnosis and fault tolerance. Next, the condition for designing a static controller is much improved. In [12,13], the flexibility of the static controller is restrictive since the control input is determined only by the external input and state feedback. To alleviate this harshness, we use a diagnoser that detects and isolates every fault occurrence and provides the fault information by switching an indicator signal.

With this additional argument, the static controller is given greater easiness in generating the control input. The reachability condition required for the proposed controller is the same as that for dynamic ones. To facilitate the design of a diagnoser and controller, we utilize the state burst, or the fast sequence of transient states traversed by the machine during transitions.

As a similar subject to the present study, the diagnosability of discrete event systems (DESs) is much investigated in supervisory control of DESs [16–20]. The proposed diagnoser differs from these results in that while the diagnosers in supervisory control receive traces of inputs (events), the proposed diagnoser utilizes stable bursts traversed by the ASM. Moreover, the diagnosers in supervisory control cannot be applied to corrective control since they do not discriminate between stable and transient states, nor do they comply with fundamental mode operations.

Here, transient faults are referred to as short-lived violation of the system's normal behavior, each fault showing no correlation with one another. Typical examples of transient faults include radiation particles, surging voltage, etc. Though not considered in this paper, there are other types of faults—permanent faults and intermittent ones. If the adverse effect of a fault persists indefinitely, it is classified as a permanent fault. Occurrences of permanent faults are mainly attributed to a physical defect or an inadequacy in the design of the system. On the other hand, an intermittent fault is a malfunction of a device or system that occurs at intervals. It may be caused by unstable or marginally stable hardware or inadequacy in the design, for example, by loose wires [21].

The outcome of transient faults caused by the adversarial input bears a strong resemblance to intelligent attacks in cyber–physical systems, where attackers may change the enablement of actuators commanded by a supervisor or sensor readings of the controlled system [22–24]. However, the adversarial input addressed in this study does not come from an intelligent entity; it is regarded as a randomly generated outer disturbance, e.g., single event upset (SEU) faults in radiation environments [14,15], or intrinsic faults occurring to the actuator [25–27]. Still, the considered fault situation is severe since the transient fault may occur at arbitrary moments.

Fault diagnosis and fault tolerant control is an area of active research not only in event-driven systems, but also in continuous-time dynamic systems. Notable among the recent results is [28] that addresses adverse effects on time delay between fault occurrence and fault accommodation in T–S fuzzy systems. Further, references [29,30] present active fault tolerant control for overcoming un-modeled actuator faults while considering time delay attributed to fault diagnosis. Transient faults in our study are conceptually similar to actuator faults, although the results of [25–27,29,30] are not applicable to controlling ASMs.

The remainder of this paper is organized as follows. In Section 2, we first address the mathematical formulation of input/state ASMs with transient faults and the overall structure of static fault tolerant control with a diagnoser. In Section 3, we present the operation of a diagnoser that can detect and isolate unauthorized transitions. Based on the fault indicator signal generated by the diagnoser, we elucidate in Section 4 the necessary and sufficient condition and synthesis procedure for a static corrective controller that realizes immediate fault recovery against every transient fault. Two illustrative examples are provided in Section 5 to demonstrate the design procedure of the proposed static controller. Finally, some conclusions are drawn in Section 6.

2. Preliminaries

For a finite set D, $|D|$ is the cardinality of D, and D^+ is the set of non-empty strings made of characters in D. For $p, q \in D^+$, $|p| \in \mathbb{N}$ is the length of p, p is a *strict prefix* of q if $q = pr$ with $r \in D^+$. $sp(q) \subset D^+$ denotes the set of all strict prefixes of q. We also denote $N_0 := \{0, 1, \ldots, n\}$ for some $n \in \mathbb{N}$. Table 1 summarizes the notations used in this paper.

Table 1. Nomenclature.

Symbol	Definition
Σ, Σ'	Input/state ASM
C	Static corrective controller
G	Diagnoser
A	Input set
$A_n, A_d \subset A$	Normal and adversarial input set
X	State set
$U(x) \subset A$	Set of inputs making stable pairs with $x \in X$
$T(x) \subset A$	Set of inputs making transient pairs with $x \in X$
f	State transition function of Σ
s, s'	Stable recursion function of Σ and Σ'
$\alpha(x, u) \in X^+$	State burst with respect to $x \in X$ and $u \in A$
$z_f, z_l \in X$	First and last element of state burst $z \in X^+$
$W(x) \subset A_d$	Set of adversarial inputs that occur at $x \in X$
$\Gamma(x) \subset X$	Set of states reached as a result of $W(x)$

ASMs are classified as input/state machines in which the current state is given as the output, and input/output ones that generate the output different from the state. In this study, we focus our concern on input/state ASMs. An input/state ASM Σ is modeled by a quadruple as follows:

$$\Sigma = (A, X, \bar{x}, f),$$

where A is the input set, $X = \{x_1, \ldots, x_n\}$ with $|X| = n$ is the state set, $\bar{x} \in X$ is the initial state, and $f : X \times A \to X$ is the state transition function partially defined on $X \times A$. A is divided into $A = A_n \dot{\cup} A_d$ where A_n and A_d are the set of normal and adversarial inputs, respectively. $(x, v) \in X \times A$ is valid if $f(x, v)$ is defined. A valid pair (x, v) is stable if $f(x, v) = x$, and transient if $f(x, v) \neq x$. Let

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

denote the set of inputs that make stable and transient pairs with x, respectively.

Owing to the absence of a synchronizing clock, Σ responds only to the input change. It rests in a stable pair (x^0, a') with $a' \in U(x^0)$ indefinitely as long as a' remains fixed. If a' changes to another value $a \in T(x^0)$, Σ engages in a chain of transient transitions as follows:

$$f(x^0, a) = x^1, f(x^1, a) = x^2, \ldots, \tag{1}$$

while a remains fixed. Provided that Σ possesses no infinite cycles, Σ reaches the next stable state as follows:

$$f(x^k, a) = f(x^{k-1}, a) = x^k \tag{2}$$

after k steps, where $1 \leqslant k \leqslant n - 1$. As transient states are traversed instantaneously, it is convenient to describe this chain of transitions only in terms of stable states, omitting instantaneous transient transitions. To this end, the stable recursion function s [1,9] is defined on every valid pair as follows:

$$s : X \times A \to X, \; s(x, v) := x',$$

where x' is the next stable state of (x, v). If (x, v) is a stable pair, $s(x, v) := x$. The chain of transient transitions characterized by s is termed a *stable transition*. In this study, $s(x, v) = x'$ is alternatively described as follows:

$$(x, v, x') \in s \Leftrightarrow s(x, v) = x'.$$

The domain of s is often extended to $X \times A_n^+$ through the following relation:

$$s(x, v_1 v_2 \cdots v_k) := s(s(x, v_1), v_2 \cdots v_k),\ v_1 v_2 \cdots v_k \in A_n^+.$$

where x' is said to be *stably reachable* from x if $s(x, t) = x'$ for some $t \in A_n^+$ [1,9]. With $|X| = n$, every state of Σ is stably reachable in at most $n - 1$ steps of stable transitions. Thus, the length of t is bounded by $1 \leqslant |t| \leqslant n - 1$.

When Σ goes through a stable transition, it generates a *state burst* [31], or a fast state sequence consisting of underlying transient states and next stable state. If each generated state is separately delivered to the controller, the resultant configuration has the state feedback control mechanism. If, on the other hand, the controller has access to the state burst, the closed-loop system is endowed with burst-feedback control. In this study, we utilize the burst feedback control scheme, as it gives more flexibility of controller synthesis, albeit needing an additional resource to record the state burst.

The state burst of a valid state–input pair is described as the following mapping:

$$\alpha : X \times A \to X^+.$$

$\alpha(x, u) \in X^+$ denotes the state burst with respect to a valid pair $(x, u) \in X \times A$, namely, $\alpha(x, u)$ is generated when Σ takes the stable transition from (x, u). For instance, (x^0, a) addressed in (1) and (2) leads to the following state burst:

$$\alpha(x^0, a) = x^0 x^1 \cdots x^k.$$

For a state burst $z \in X^+$, let $z_f \in X$ and $z_l \in X$ be its first and last element, respectively. In terms of the foregoing notations, $\alpha_f(x^0, a) = x^0$ and $\alpha_l(x^0, a) = x^k$.

Figure 1 shows the proposed static fault tolerant corrective control system, where C is the static corrective controller, G is the diagnoser, $v \in A_n$ is the external input, $u \in A_n$ is the control input provided by C, $w \in A_d$ is the adversarial input, $z \in X^+$ is the state burst, and $x \in X$ and $m \in N_0$ are respectively the state feedback and the fault indicator signal generated by G. Σ_c denotes the closed-loop system consisting of Σ, C, and G. When w enters Σ_c, it overrides u and causes a transient fault, forcing Σ to undergo an unauthorized state transition whenever it is valid with respect to the current state. w represents an external disturbance that infiltrates into Σ_c through the control input channel. As addressed before, a typical instance of w is an SEU fault [14,15] that upsets the logic value of memory bits expressing the control input. Inherent mechanical or electrical faults to the actuator can be also modeled by w.

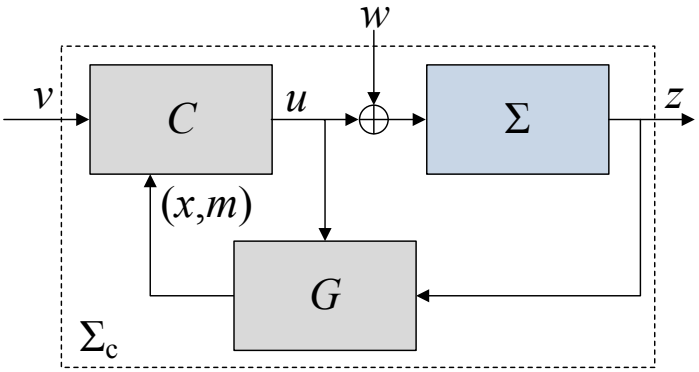

Figure 1. Static corrective control system with a diagnoser.

For $x \in X$, let

$$W(x) := T(x) \cap A_d$$

be the set of adversarial inputs that cause unauthorized transitions to Σ when it stays at the stable state x, and let

$$\Gamma(x) := \{s(x, w) | w \in W(x)\}$$

be the set of states Σ reaches as a result of a transient fault occurring at x. Note that w is unobservable from both C and G, which fits into the characteristics of adversarial entities.

G provides C with the next stable state $x \in X$ and the fault indicator signal $m \in N_0$ based on the state burst z and the control input u. Thus G has the following mapping:

$$G : X^+ \times A_n \rightarrow X \times N_0.$$

With the state notation $X = \{x_1, \ldots, x_n\}$, $m = 0$ indicates that Σ has undergone a nominal stable transition. On the other hand, $m = i \neq 0$ implies that Σ has undergone an unauthorized transition from x_i caused by an unspecified adversarial input $w \in W(x_i)$.

Referring to Figure 1, C receives the input triplet (x, v, m) and generates the control input u as the output. Being a static controller, C is represented by the following function:

$$C : X \times A_n \times N_0 \rightarrow A_n.$$

During the normal behavior of Σ, C just relays the external input to Σ without modification. When a transient fault is diagnosed, C provides appropriate control input sequences. The control objective is to achieve immediate fault recovery, namely, to take Σ from the faulty state to the original state at which the fault occurs before further change in the external input. Since neither C nor G is governed by a synchronizing clock, their operations are also conducted instantaneously under asynchrony. Hence the procedure of fault diagnosis and fault tolerant control can be completed before the external input changes to the next value, rendering Σ_c to show the normal input/state behavior as if no fault has happened.

Σ_c is assumed to preserve fundamental mode operations [32], wherein no two variables change simultaneously. Under the fundamental mode, w is supposed to occur to Σ only when Σ stays at a stable state. This is not a burdensome constraint since the speed of transient transitions is so fast that the possibility of fault occurrences during the transitions is negligible. Hence the stable state at which the fault occurs serves as the goal state for the corresponding fault tolerant control. In a similar sense, it is also supposed that v is not altered during the correction procedure.

Remark 1. *Whereas the present study focuses on transient faults of which influence on the machine vanishes right after its occurrence, there exist other kinds of faults differing in the durability of their influences. If the adverse effect persists for a finite time after initial occurrence, the fault is termed an intermittent fault [33]; if the effect remains indefinitely (or irreversible), it is termed a permanent fault [34]. For input/state ASMs, fault recovery is impossible for either intermittent or permanent faults since immediate return to the original state cannot be implemented. The latter problem can be tackled for input/output ASMs, where the output differs from the present state [35], or switched ASMs that possess redundant states which may substitute faulty states [8].*

Remark 2. *In the field of DESs, the stability and stabilizability of the system under static feedback control means that starting from any arbitrary initial state, the system can (or can be controlled to) go to a "legal state" and stay there after a finite number of transitions [36]. In our problem setting, the original state at which a fault occurs can be regarded as a legal state. With no infinite cycles, further, fault recovery implies that Σ must be controlled to the original state in finite steps of stable transitions. In this sense, the fault tolerant controllability of Σ is equivalent to the stabilizability of*

DESs in supervisory control. Note that a stable *pair* $(x, v) \in X \times A$ *just implies that* x *is a fixed point of* f*, irrelevant to the stability of* Σ.

3. Diagnoser

For G to determine whether or not a transient fault occurs, z_f and z_l of the current state burst z are investigated with respect to $u \in A_n$. Assume first that $(z_f, u, z_l) \in s$. This implies that Σ undergoes a nominal stable transition $z_f \xrightarrow{u} z_l$. To signify this, we assign the fault indicator signal $m := 0$. Accordingly, G is set to be $G(z, u) := (z_l, 0)$ if $(z_f, u, z_l) \in s$.

On the other hand, assume that $(z_f, u, z_l) \notin s$. This elucidates that the latest stable transition is caused not by u, but by $w \in W(z_f)$ such that $s(z_f, w) = z_l$ and $z_l \in \Gamma(z_f)$. In this case, we assign $m := i \in N_0 \backslash \{0\}$, where i is the index of the state at which the fault occurs, i.e., $z_f = x_i$. Hence $G(z, u) := (z_l, i)$ if $(z_f, u, z_l) \notin s$ and $z_f = x_i$.

Once a transient fault is diagnosed, Σ is controlled to return to the original state x_i via a chain of stable transitions. This means that after diagnosing a transient fault, G receives a sequence of pairs of state bursts and control inputs characterizing nominal stable transitions. Since the procedure of fault recovery persists until Σ reaches x_i, G must continue to give the same fault indicator signal $m = i$ unless the state feedback x_i is received. Thus m of $G(z, u)$ is set to be unchanged if $(z_f, u, z_l) \in s$, $m_o = i \neq 0$, and $z_l \neq x_i$, where m_o is the previous value of m.

Finally, assume that fault recovery is accomplished as Σ reaches x_i. Upon receiving the state feedback x_i, G must signify the end of the recovery procedure. Hence G is designed to generate $m = 0$ at this phase, or $G(z, u) := (z_l, 0)$ if $(z_f, u, z_l) \in s$, $m_o = i$, and $z_l = x_i$.

Combining the above discussions, we encapsulate the algorithm of fault detection and isolation by G.

Proposition 1. *Algorithm 1 is correct, i.e., fault detection and isolation is characterized by* m.

Algorithm 1: Fault detection and isolation by G with $X = \{x_1, \ldots, x_n\}$ and $m_o \in N_0$ (the previous value of m):

1. Receive z and u, and extract z_f and z_l from z.
2. If $m_o = 0$, check (z_f, u, z_l):
   a. If $(z_f, u, z_l) \notin s$, set $m := i$ where $z_f = x_i$.
   b. Else if $(z_f, u, z_l) \in s$, maintain $m = 0$.
3. Else if $m_o = i \neq 0$, check z_l:
   a. If $z_l \neq x_i$, maintain $m = i$.
   b. Else if $z_l = x_i$, set $m := 0$.
4. If $m = i$, a transient fault has occurred at x_i; else if $m = 0$, no fault has occurred yet or the normal input/state behavior of Σ is recovered.

Proof. In Algorithm 1, the value of m is 0 at either Step 2.b or Step 3.b. The algorithm goes to Step 2.b when $(z_f, u, z_l) \in s$. As this indicates the execution of the nominal stable transition $z_f \xrightarrow{u} z_l$, no transient fault occurs. Step 3.b implies that after a transient fault at x_i ($m_o = i$), Σ recovers the normal behavior by reaching x_i. Hence $m = 0$ corresponds to no fault occurrence or fault recovery. On the other hand, m is assigned a non-zero index at either Step 2.a or Step 3.a. Since both steps represent an unauthorized transition from x_i or failure of returning to x_i, $m = i$ serves as the indicator for fault detection and isolation. □

A formal definition of G is constructed from the above algorithm as follows:

$$G(z,u) := \begin{cases} (z_l, i) & (z_f, u, z_l) \notin s,\ u \in U(z_f), \text{ and } z_f = x_i \\ (z_l, \text{-}) & (z_f, u, z_l) \in s,\ m_o = i, \text{ and } z_l \neq x_i \\ (z_l, 0) & \text{otherwise} \end{cases} \tag{3}$$

where '-' in the second line implies 'unchanged'.

To comply with fundamental mode, G should provide C with (x, m) only when Σ stays at a stable state. To this end, the end of every stable transition must be identified. In the nominal transitions, it is easily done by referring to z and u since the current stable transition will end at $z_l = s(z_f, u)$. In the case of unauthorized transitions, however, one must determine it only by referring to z, as w is unobservable. The latter property is termed *fault detectability* [31]. The condition for fault detectability with respect to the state burst is that any state burst generated during the unauthorized transition is not a strict prefix of another one (Theorem 3.7 of [31]). The underlying reason is obvious as elicited in the following.

Proposition 2. *For $x \in X$ with $W(x) \neq \varnothing$, Σ is fault detectable at x if and only if $\alpha(x,w) \notin sp(\alpha(x,w'))$, $\forall w, w' \in W(x)$ with $\alpha(x,w) \neq \alpha(x,w')$.*

Proof. (Only if) Assume that Σ is fault detectable at x but $\alpha(x,w_1) \in sp(\alpha(x,w_2))$ for some $\{w_1, w_2\} \subset W(x)$. Assume further that while u remains unchanged, G receives the changed state burst $\alpha(x,w_1)$. Then, one cannot determine whether w_1 occurs so that Σ reaches the faulty state $\alpha_l(x,w_1) \in \Gamma(x)$, or w_2 occurs so that Σ is on its way to the corresponding faulty state $\alpha_l(x,w_2) \in \Gamma(x)$, passing through the intermediate transient state $\alpha_l(x,w_1)$. This contradicts the assumption of fault detectability at x.

(If) Suppose that G receives $z = \alpha(x,w)$ (w is unknown) such that $(z_f, u, z_l) \notin s$. Since $\alpha(x,w) \notin sp(\alpha(x,w'))$ for all $w' \in W(x)$ with $\alpha(x,w) \neq \alpha(x,w')$, one can identify the end of the unauthorized transition merely by referring to z. Hence Σ is fault detectable at x. □

4. Controller Synthesis

A necessary condition for taking Σ from a beginning state to a goal state via corrective control is that the goal state is stably reachable from the beginning state, namely, an input string exists with which Σ stably reaches the goal state [9]. Such input strings are utilized by the controller in building feedback paths. For static controllers, another significant condition must be satisfied with respect to the set of utilized input sequences. Slightly relaxing the notion presented in [12,13], we define the *implementability* of static corrective controllers.

Definition 1. *A static corrective controller is said to be* implementable *with respect to a set of control input sequences $S \subset A^+$ if by utilizing S, the controller's output is uniquely determined by each input combination of the controller.*

Implementability is necessary for the integrity of the static controller since having no states, the static controller must always generate a unique output with respect to each input combination.

The constraint on implementability makes it impossible to apply the previous static controller [12,13] to fault recovery. To validate the latter assertion, assume that Σ undergoes an unauthorized transition from x to x' by $w \in W(x)$ ($s(x,w) = x'$) during which both external and control inputs remain $a \in A_n$. Assume further that x is stably reachable from x', e.g., $s(x', bcd) = x$ for $bcd \in A_n^+$. If we use the previous static controller

$$C_p : X \times A_n \to A_n \tag{4}$$

with no indicator signal as its argument, we must assign the output b to the input combination (x', a) to start the correction procedure, that is, $C_p(x', a) = b$. However, (x', a) may be a valid pair of Σ, e.g., $a \in T(x')$. Then, in the normal behavior of Σ, C_p must provide a as the control input without modification, i.e., one must design C_p such that $C_p(x', a) = a$ to maintain the nominal stable transition from (x', a). Since this conflicts with the foregoing assignment $C_p(x', a) = b$, the previous static corrective controller cannot be applied to the present problem.

The use of G and the addition of the related argument m to C resolve the aforementioned predicament. m is retained as i throughout the recovery procedure for the unauthorized transition from x_i. Further, G gives C only the next stable state, discarding the underlying transient states. Hence the proposed static controller achieving fault recovery against $W(x_i)$ can be designed if and only if x_i is stably reachable from every $x_j \in \Gamma(x_i)$, which equals the following condition for the existence of a corresponding dynamic controller.

Lemma 1 ([31]). *Given Σ with $W(x_i) \neq \varnothing$ for $x_i \in X$, a dynamic corrective controller exists that achieves fault recovery against $W(x_i)$ if and only if the following holds:*

$$\forall x_j \in \Gamma(x_i), \exists t_{j,i} \in A_n^+ : \; s(x_j, t_{j,i}) = x_i. \tag{5}$$

Provided that the above condition is valid, we design the proposed static controller $C : X \times A_n \times N_0 \to A_n$. First, if $m = 0$, C just relays the external input to the control input channel, as no transient fault occurs. For this purpose, we set the following:

$$C(x, v, 0) := v, \; \forall (x, v) \in X \times A_n. \tag{6}$$

To design the operation of fault recovery, take an arbitrary faulty state $x_j \in \Gamma(x_i)$ and denote a proper input string by $t_{j,i} := t = u_1 \cdots u_{|t|}$ with $s(x_j, t) = x_i$. Denote further by $x^1, \ldots, x^{|t|} \in X$ the intermediate stable states Σ traverses when it undertakes the chain of stable transitions in response to t, i.e., the following:

$$\begin{aligned} s(x_j, u_1 \cdots u_h) &= x^h \\ s(x^{h-1}, u_h) &= x^h, \; h = 1, \ldots, |t|, \end{aligned}$$

where $x^0 = x_j$ and $x^{|t|} = x_i$. Suppose that Σ is staying at a stable pair (x_i, a) when $w \in W(x_i)$ infiltrates into Σ such that $s(x_i, w) = x_j$. The input pair of G changes to $(\alpha(x_i, w), a)$ at this time, where $\alpha_f(x_i, w) = x_i$ and $\alpha_l(x_i, w) = x_j$. Since $(x_i, a, x_j) \notin s$, G generates $m = i$ according to (3). Receiving (x_j, i) at the instant of the fault occurrence, C commences the correction procedure by generating u_1 until Σ reaches $s(x_j, u_1) = x^1$. Transient states underlying between x_j and x^1 need not be considered since they are discarded by G. Hence we set C as follows:

$$C(x_j, a, i) := u_1. \tag{7}$$

As soon as the state feedback changes to x^1, C provides u_2, which takes Σ toward $s(x_j, u_1 u_2) = x^2$, and so on. The following assignment materializes these recursive operations of C.

$$C(x^{h-1}, a, i) := u_h, \; h = 2, \ldots, |t|. \tag{8}$$

Fault recovery is accomplished when Σ reaches x_i in response to $u_{|t|}$. m is reset to 0 at this time. Since $m = 0$, C generates a again as the control input according to (6).

By Definition 1, the control input sequences should be selected in such a way that C is implementable. We assert that if the reachability condition for a dynamic corrective

controller is valid for every x_i with $W(x_i) \neq \varnothing$, one can find a set of input sequences for which C is implementable.

Proposition 3. *If Lemma 1 is valid for every x_i with $W(x_i) \neq \varnothing$, a set of control input sequences exists such that C designed according to (6)–(8) is implementable.*

Proof. If $|W(x_i)| = 1$, the implementability of C with respect to x_i is ensured trivially since only one input string is used as the control input sequence for the transient fault. Else if $|W(x_i)| > 1$, consider $w, w' \in W(x_i)$ with $s(x_i, w) := x_j \neq s(x_i, w') := x_k$. By assumption, $t := u_1 \cdots u_{|t|}$ and $r := u'_1 \cdots u'_{|r|}$ exist such that $s(x_j, t) = s(x_k, r) = x_i$. Let $X_{j,i}$ and $X_{k,i}$ be the set of intermediate stable states Σ traverses when Σ undertakes the chain of stable transitions from x_j to x_i and from x_k to x_i, respectively. Then, we have the following:

$$\begin{aligned} X_{j,i} &= \{x_j, s(x_j, u_1), \ldots, s(x_j, u_1 \cdots u_{|t|-1})\} \\ X_{k,i} &= \{x_k, s(x_k, u'_1), \ldots, s(x_k, u'_1 \cdots u'_{|r|-1})\}. \end{aligned} \tag{9}$$

If $X_{j,i} \cap X_{k,i} = \varnothing$, the use of t and r satisfies the implementability of C since each feedback path contains no common state. Otherwise, Σ passes through a common state $x_l \in X_{j,i} \cap X_{k,i}$ when it is driven by C along two state trajectories $x_j \to x_i$ and $x_k \to x_i$. Specifically, assume the following:

$$s(x_j, u_1 \cdots u_p) = s(x_k, u'_1 \cdots u'_q) = x_l$$

where $0 \leqslant p < |t|$ and $0 \leqslant q < |r|$ ($s(x_j, u_0) := x_j$ and $s(x_k, u'_0) := x_k$). If $u_{p+1} = u'_{q+1}$, the implementability of C is still valid with respect to x_l since C will generate the same output u_{p+1} ($= u'_{q+1}$) in response to x_l. Else if $u_{p+1} \neq u'_{q+1}$, we adjust one of t and r as follows to satisfy the implementability. First, the suffix lengths $|t| - p$ and $|r| - q$ are compared. Suppose that $|t| - p > |r| - q$. Next, the suffix of t is substituted by the corresponding part of r, i.e., we induce an alternative input string t' from t and r as follows:

$$t' := u_1 \cdots u_p u'_{q+1} \cdots u'_{|r|}.$$

Since $s(x_j, t') = x_i$ by the definition of t and r, t' can be used instead of t for fault recovery from x_j to x_i. It is clear that C is implementable with respect to the derived input sequence. Since m is distinctive with respect to x_i, the latter implies the implementability of C. □

In the above proof, the suffix of the previous input sequence with a longer length ($|t| - p$) is replaced by the shorter one ($|r| - q$). This is intended to reduce the computational load of the controller by taking a shorter feedback path.

Theorem 1. *Given the configuration of Figure 1, assume that $W(x_i) \neq \varnothing$ for all $x_i \in \hat{X} \subset X$ of Σ. Then, C achieving fault recovery against transient faults by A_d exists if and only if every unauthorized transition is fault detectable and $\forall x_i \in \hat{X}$, $\forall x_j \in \Gamma(x_i)$, x_i is stably reachable from x_j.*

Proof. (If) Since every unauthorized transition is fault detectable, we can design G by applying (3). In addition, since x_i is stably reachable from every state of $\Gamma(x_i)$, by Proposition 3 we can find a set of control input sequences with which C is implementable and can design C by referring to (6)–(8). According to the foregoing discussions, Σ_c is immune against any unauthorized transitions by A_d.

(Only if) In view of Figure 1, the existence of C implies that the next stable state of every stable transition is identified solely by observing z. Hence every unauthorized transition is fault detectable by Proposition 2. Since C has access to Σ only by changing u, fault recovery ensures that for all x_i with $W(x_i) \neq \varnothing$ and for all $x_j \in \Gamma(x_i)$, $t_{j,i} \in A_n^+$ exists which takes Σ from x_j to x_i via a chain of stable transitions. □

Figure 2 illustrates the flowchart elucidating the execution of the proposed static corrective controller for achieving fault tolerance. Compared with dynamic controllers [31], the improvement of computational load in controller synthesis is obvious. The dynamic controller needs to define maximum $n+1$ states for the correction procedure of each unauthorized transition. Since there may be maximally $n(n-1)$ unauthorized transitions, the size (or number of states) of the overall controller has the complexity of $O(n^3)$. On the other hand, the proposed static controller is much more efficient, as it needs no states. Of course, G requires memory elements as the state burst and the previous output m_o must be recorded. Since the maximum length of the state burst is $n-1$, the construction of G is computed in $O(n)$, which is a mild degradation of resource usage. Note that a symbolic computation algorithm for inducing feasible control input sequences addressed in (5) is presented in the prior work [1,2]. Further, numerical algorithms avoiding tedious symbolic computations are found in the recent results [37,38].

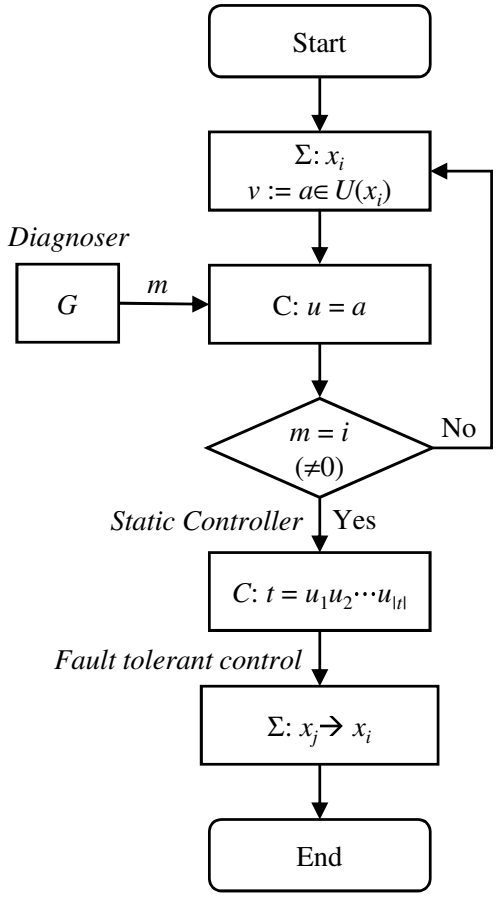

Figure 2. Flowchart of the proposed static corrective control scheme.

5. Illustrative Examples

5.1. Home Security System

Consider $\Sigma_1 = (A, X, \bar{x}, f)$ whose state flow diagram is shown in Figure 3, where $A_n = \{a, b, c, d\}$, $A_d = \{w_1, w_2\}$, and $\bar{x} = x_1$. Σ_1 represents a home security system [12], where x_1 is the initial state and x_2–x_4 are three alarm states that are reached by break-in events d, a, and b, respectively. c is the reset signal that is activated only at x_3.

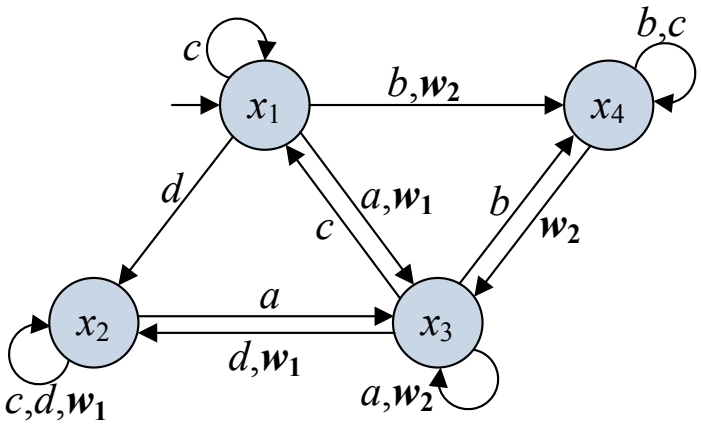

Figure 3. State flow diagram of a home security system Σ_1.

As an example instance, let us take the case of x_1 with $U(x_1) = \{c\}$ and $W(x_1) = \{w_1, w_2\}$. To check the fault detectability, derive the state bursts caused by the elements of $W(x_1)$ as $\alpha(x_1, w_1) = x_1x_3x_2$ and $\alpha(x_1, w_2) = x_1x_4x_3$. Since $\alpha(x_1, w_1) \notin sp(\alpha(x_1, w_2))$ and vice versa, Σ_1 is fault detectable at x_1 by Proposition 2.

Following (3), we design G as follows:

$$\begin{aligned} G(x_1x_3x_2, c) &= (x_2, 1) \\ G(x_1x_4x_3, c) &= (x_3, 1) \\ G(z, u) &= (z_l, \text{-}),\ (z_f, u, z_l) \in s,\ m_o = 1,\ z_l \neq x_1 \\ G(z, u) &= (z_l, 0),\ \text{otherwise.} \end{aligned}$$

If G receives $x_1x_3x_2$ or $x_1x_4x_3$ while u is equal to c, the unauthorized transition by w_1 or w_2 is diagnosed with certainty since $(x_1, c, x_2) \notin s$ and $(x_1, c, x_3) \notin s$.

Referring to Figure 3, x_1 is stably reachable from every state of $\Gamma(x_1) = \{x_2, x_3\}$, e.g., $s(x_2, ac) = x_1$ and $s(x_3, c) = x_1$. By Theorem 1, C exists which achieves fault recovery against $W(x_1)$. To check the implementability of C associated with ac and c, $X_{2,1}$ and $X_{3,1}$ defined in (9) are derived as $X_{2,1} = \{x_2, x_3\}$ and $X_{3,1} = \{x_3\}$. Although $X_{2,1} \cap X_{3,1} = \{x_3\}$ is non-empty, both ac and c have the common input c for x_3. Hence they can be used as the control input sequences while guaranteeing the implementability of C.

The design procedure of C is straightforward. Since $s(x_2, a) = x_3$ and $s(x_3, c) = x_1$, the control inputs with respect to $m = 1$ are assigned in line with (7) and (8) as follows:

$$\begin{aligned} C(x_2, c, 1) &= a \\ C(x_3, c, 1) &= c. \end{aligned} \tag{10}$$

When an unauthorized transition occurs so that m changes to 1 and the state feedback to x_2 or x_3, C activates the recovery procedure by performing the above operations. When Σ_1 reaches x_1, m is reset to 0, and upon receiving $(x_1, 0)$ from G, C terminates the recovery procedure. To this end, set C as follows:

$$C(x_1, c, 0) = c.$$

Since the fault detectability condition in Proposition 2 and the reachability condition (5) are valid for the other states as well, the design of G and C for the rest of states and inputs can be similarly conducted.

As a comparative study, let us try to achieve the above fault tolerant control by the previous static controller C_p receiving no fault indicator signal (see (4)). Assume that w_2 occurs when Σ_1 has been staying at (x_1, c) so that Σ_1 is forced to reach x_2. In a similar way to (10), C_p must generate a to initiate the correction procedure, namely, $C_p(x_2, c) = a$. However, since $s(x_2, c) = c$ in Figure 3, C_p already has the assignment $C_p(x_2, c) = c$ for

ensuring the nominal transition. As this contradicts the foregoing operation, one cannot design C_p that accomplishes fault tolerant control for Σ_1.

5.2. Asynchronous Error Counter

As the second example, we apply the proposed scheme to fault tolerant control for asynchronous error counters embedded in the satellite computers [39]. Since SEU faults caused by cosmic rays in space corrupt logic values of memory elements in the computers, periodic memory scrubbing is needed based on the amount of accumulated errors. Error counters play the role of detecting and recording the error occurrences by transferring to specific states that characterize the degree of error occurrences.

Consider an asynchronous 6-error counter $\Sigma_2 = (A, X, \bar{x}, f)$ whose state flow diagram is shown in Figure 4, where $A_n = \{a_i, b_i | 1 \leqslant i \leqslant 6\}$, $A_d = \{n_j, f_j | 0 \leqslant j \leqslant 2\}$, and $\bar{x} = x_1$. Σ_2 receives two kinds of error signals: a_i with one-step resolution and b_i with two-step resolution. Typical examples of a_i and b_i are 1-bit and 2-bit errors that frequently occur in space-born digital systems [14,15]. It is assumed that every character of A_n can be generated for control purposes. In accordance with the meaning of each signal, Σ_2 advances one state in response to a_i and two states in response to b_i as depicted in Figure 4.

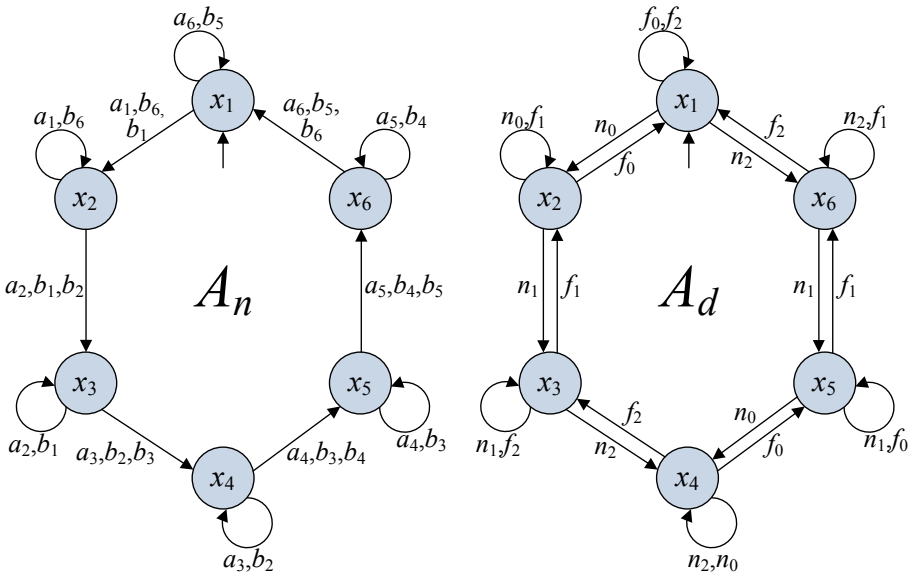

Figure 4. State flow diagram of an asynchronous 6-error counter Σ_2: state transitions with respect to A_n are drawn on the left, and those with respect to A_d are on the right.

Σ_2 counts maximally six occurrences of a_i's and three occurrences of b_i's, after which Σ_2 is reset to x_1. Supposing that Σ_2 is implemented as a digital circuit, we assign a three-bit binary number $c_2c_1c_0$ to each state as follows: $x_1 = 000$, $x_2 = 001$, $x_3 = 011$, $x_4 = 111$, $x_5 = 110$, and $x_6 = 100$.

Working in space, all the memory elements corresponding to $c_2c_1c_0$ are also exposed to SEU faults. In A_d, n_j and f_j represent an SEU fault that upsets the logic value of c_j from 0 to 1 and from 1 to 0, respectively, $j = 0, 1, 2$. The control goal is to design G and C, if any, that accomplish fault diagnosis and fault recovery against any adversarial input of A_d.

By observing Figure 4, we can derive each $W(x_i)$ and $\Gamma(x_i)$, e.g., for x_1,

$$W(x_1) = \{n_0, n_2\},\ \Gamma(x_1) = \{x_2, x_6\}.$$

It is found that $W(x_i) \neq \varnothing$, $\forall x_i \in X$. Let us investigate fault detectability for the existence of G. For instance, consider the case of x_1. The state bursts produced by $W(x_1)$ are $\alpha(x_1, n_0) = x_1x_2$ and $\alpha(x_1, n_2) = x_1x_6$. Since $\alpha(x_1, n_0) \notin sp(\alpha(x_1, n_2))$ and vice versa, any unauthorized transitions occurring at x_1 are fault detectable by Proposition 2. In fact, fault detectability is ensured for the rest of the states and thus G can be designed.

To construct G, we continue to concern the case of x_1. A fault occurrence at x_1 is diagnosed when the state burst is observed to change to either x_1x_2 or x_1x_6, while the control input remains unchanged. According to the first line of (3), G is set to be $(U(x_1) \cap A_n = \{a_6, b_5\})$ as follows:

$$G(x_1x_2, u) = (x_2, 1),\ \forall u \in \{a_6, b_5\}$$
$$G(x_1x_6, u) = (x_6, 1),\ \forall u \in \{a_6, b_5\}.$$

When Σ_2 is under the procedure of fault recovery, G must not change m as defined in the second line of (3). For the case of x_1, the latter operation is materialized by setting the following:

$$G(z, u) = (z_l, \text{-}),\ (z_f, u, z_l) \in s,\ m_o = 1,\ z_l \neq x_1.$$

Finally, when Σ_2 reaches the original state x_1, m is reset to 0 as defined in the third line of (3). The operation of G at the other states is designed in a similar manner.

To determine the existence of C, we now investigate stable reachability between x_i and $\Gamma(x_i)$. An examination of Figure 4 shows that for all $i = 1, \ldots, 6$, x_i is stably reachable from every state of $\Gamma(x_i)$. For instance, x_1 is stably reachable from x_2 via an input string $b_2b_4a_6$ $(s(x_2, b_2b_4a_6) = x_1)$, and from x_6 via an input string a_6 $(s(x_6, a_6) = x_1)$. By Theorem 1, therefore, C exists which achieves fault recovery against transient faults by A_d.

C is constructed in line with (6)–(8). Let us keep focusing on the fault recovery to x_1. As addressed above, we employ $b_2b_4a_6$ and a_6 in activating the correction procedure from x_2 and x_6, respectively. Upon receiving $b_2b_4a_6$, Σ_2 traverses two intermediate stable states $s(x_1, b_2) = x_4$ and $s(x_4, b_4) = x_6$. With m fixed to 1, C is designed with respect to x_2 and $b_2b_4a_6$ as follows:

$$\begin{aligned} C(x_2, u, 1) &= b_2,\ \forall u \in \{a_6, b_5\} \\ C(x_4, u, 1) &= b_4,\ \forall u \in \{a_6, b_5\} \\ C(x_6, u, 1) &= a_6,\ \forall u \in \{a_6, b_5\}. \end{aligned} \tag{11}$$

On the other hand, Σ_2 transfers from x_6 directly to x_1 in response to a_6, so C is designed as $C(x_6, u, 1) = a_6$, $\forall u \in \{a_6, b_5\}$. As (11) already contains this operation, it serves as the correction procedure realizing fault recovery to x_1.

When Σ_2 reaches x_1, m is reset to 0. Upon receiving $(x_1, 0)$ from G, C terminates the recovery procedure. To this end, set C as follows:

$$C(x_1, u, 0) = u,\ \forall u \in \{a_6, b_5\}. \tag{12}$$

It is clear from (11) and (12) that the selected control input sequences preserve the implementability. The operation of C for the other states is attained by adopting (11) and (12). Since all the interactions between C, G, and Σ_2 are conducted in an asynchronous mechanism, the correction procedure can be accomplished instantaneously before further change of the external input.

6. Conclusions and Challenges

We have shown that static corrective controllers can solve the problem of fault diagnosis and fault tolerant control for input/state ASMs subject to transient faults. The state burst is used as feedback to design a diagnoser that detects and isolates any transient fault. Since the static controller receives only the stable state and fault indicator signal from the diagnoser, the reachability condition for designing the controller is greatly enhanced compared with the previous result. We have addressed the existence conditions for the diagnoser and static controller and formal algorithms for their synthesis in the framework of corrective control. The case studies on the home security system and the asynchronous error counter validate the applicability of the proposed control scheme.

Since the closed-loop system with the proposed static corrective controller preserves fundamental mode operations, we can code the system in very high speed integrated circuit hardware description language (VHDL) so as to implement it on configurable semiconductor devices such as field-programmable gate arrays (FPGAs); refer to [6–8] for the relevant prior work. We expect that the implementation of the closed-loop system will take significantly fewer resources than the case of dynamic corrective controllers, albeit the addition of the diagnoser G. The design and implementation of the proposed static corrective control scheme on digital systems will be conducted as a further study.

While the ASM in this study has the form of an input/state machine, many practical ASMs are modeled by input/output machines. Hence establishing a static corrective control scheme for input/output ASMs is an important future research topic. Further, although only transients faults were considered in this paper, fault tolerant control for other types of faults, e.g., permanent faults and intermittent ones, may be solved in input/output ASMs as addressed in Remark 1. Hence applying the proposed static control methodology to overcoming such faults in input/output ASMs is also an interesting future research topic.

Author Contributions: Conceptualization, J.-M.Y.; funding acquisition, J.-M.Y., S.-J.P., and S.W.K.; supervision, J.-M.Y.; methodology, S.-J.P. and S.W.K.; formal analysis, J.-M.Y.; software and hardware, S.W.K.; writing—original draft preparation, J.-M.Y.; writing—reviewing and editing, S.-J.P. and S.W.K. All authors have read and agreed to the published version of the manuscript.

Funding: This research was supported in part by Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education (no. NRF-2021R1I1A3040696), in part by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT) (no. NRF-2018R1A5A1025137), and in part by Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education (no. NRF-2016R1D1A1B02012959).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Geng, X.; Hammer, J. Input/output control of asynchronous sequential machines. *IEEE Trans. Autom. Control* **2005**, *50*, 1956–1970. [CrossRef]
2. Venkatraman, N.; Hammer, J. On the control of asynchronous sequential machines with infinite cycles. *Int. J. Control* **2006**, *79*, 764–785. [CrossRef]
3. Xu, X.; Hong, Y. Matrix approach and model matching of asynchronous sequential machines. *IEEE Trans. Autom. Control* **2013**, *58*, 2974–2979. [CrossRef]
4. Wang, J.; Han, X.; Chen, Z.; Zhang, Q. Model matching of input/output asynchronous sequential machines based on the semi-tensor product of matrices. *Future Gener. Comput. Syst.* **2018**, *83*, 468–475. [CrossRef]
5. Wang, B.; Feng, J.E.; Meng, M. Model matching of switched asynchronous sequential machines via matrix approach. *Int. J. Control* **2019**, *92*, 2430–2440. [CrossRef]
6. Yang, J.-M.; Kwak, S.W. Realizing fault-tolerant asynchronous sequential machines using corrective control. *IEEE Trans. Control Syst. Technol.* **2010**, *18*, 1457–1463. [CrossRef]
7. Yang, J.-M.; Kwak, S.W. Output feedback control of asynchronous sequential machines with disturbance inputs. *Inf. Sci.* **2014**, *259*, 87–99. [CrossRef]
8. Yang, J.-M.; Kwak, S.W. Fault tolerance in switched ASMs with intermittent faults. *IET Control Theory Appl.* **2017**, *11*, 1443–1449. [CrossRef]
9. Murphy, T.E.; Geng, X.; Hammer, J. On the control of asynchronous machines with races. *IEEE Trans. Autom. Control* **2003**, *48*, 1073–1081. [CrossRef]
10. Peng, J.; Hammer, J. Input/output control of asynchronous sequential machines with races. *Int. J. Control* **2010**, *83*, 125–144. [CrossRef]
11. Yang, J.-M. A simple fault tolerant control for input/output asynchronous sequential machines. *Automatica* **2015**, *52*, 76–82. [CrossRef]

12. Yang, J.-M.; Hammer, J. Static state feedback control of asynchronous sequential machines. *Int. J. Gen. Syst.* **2016**, *45*, 830–863. [CrossRef]
13. Wang, B.; Feng, J.E. A matrix approach for the static correction problem of asynchronous sequential machines. *Int. J. Control Autom. Syst.* **2020**, *18*, 477–485. [CrossRef]
14. Caron, P.; Inguimbert, C.; Artola, L.; Ecoffet, R.; Bezerra, F. Physical mechanisms of proton-induced single-event upset in integrated memory devices. *IEEE Trans. Nucl. Sci.* **2019**, *66*, 1404–1409. [CrossRef]
15. He, G.; Zheng, S.; Jing, N. A hierarchical scrubbing technique for SEU mitigation on SRAM-based FPGAs. *IEEE Trans. Very Large Scale Integr. VLSI Syst.* **2020**, *28*, 2134–2145. [CrossRef]
16. Sampath, M.; Sengupta, R.; Lafortune, S.; Sinnamohideen, K.; Teneketzis, D. Diagnosability of discrete-event systems. *IEEE Trans. Autom. Control* **1995**, *40*, 1555–1575. [CrossRef]
17. Zad, S.H.; Kwong, R.H.; Wonham, W.M. Fault diagnosis in discrete-event systems: Framework and model reduction. *IEEE Trans. Autom. Control* **2003**, *48*, 1199–1212. [CrossRef]
18. Zaytoon, J.; Lafortune, S. Overview of fault diagnosis methods for discrete event systems. *Annu. Rev. Control* **2013**, *37*, 308–320. [CrossRef]
19. Santoro, L.P.; Moreira, M.V.; Basilio, J.C. Computation of minimal diagnosis bases of discrete-event systems using verifiers. *Automatica* **2017**, *77*, 93–102. [CrossRef]
20. Zhu, G.; Li, Z.; Wu, N.; Al-Ahmari, A. Fault identification of discrete event systems modeled by Petri nets with unobservable transitions. *IEEE Trans. Syst. Man, Cybern. Syst* **2019**, *49*, 333–345. [CrossRef]
21. Siewiorek, D.P.; Swarz, R.S. *Reliable Computer Systems*, 2nd ed.; Digital Press: Bedford, MA, USA, 1992.
22. Carvalho, L.K.; Wu, Y.C.; Kwong, R.; Lafortune, S. Detection and mitigation of classes of attacks in supervisory control systems. *Automatica* **2018**, *97*, 121–133. [CrossRef]
23. Humayed, A.; Lin, J.; Li, F.; Luo, B. Cyber-physical systems security—A survey. *IEEE Internet Things J.* **2017**, *4*, 1802–1831. [CrossRef]
24. Su, R. Supervisor synthesis to thwart cyber attack with bounded sensor reading alterations. *Automatica* **2018**, *94*, 35–44. [CrossRef]
25. Li, Y.X.; Yang, G.H. Adaptive fuzzy decentralized control for a class of large-scale nonlinear systems with actuator faults and unknown dead zones. *IEEE Trans. Syst. Man, Cybern. Syst* **2017**, *47*, 729–740. [CrossRef]
26. Jin, X.; Qin, J.; Shi, Y.; Zheng, W.X. Auxiliary fault tolerant control with actuator amplitude saturation and limited rate. *IEEE Trans. Syst. Man, Cybern. Syst* **2018**, *48*, 1816–1825. [CrossRef]
27. Huang, C.; Naghdy, F.; Du, H. Delta operator-based model predictive control with fault compensation for steer-by-wire systems. *IEEE Trans. Syst. Man, Cybern. Syst* **2020**, *50*, 2257–2272. [CrossRef]
28. Shen, Q.; Jiang, B.; Shi, P. Adaptive fault diagnosis for T–S fuzzy systems with sensor faults and system performance analysis. *IEEE Trans. Fuzzy Syst.* **2014**, *22*, 274–285. [CrossRef]
29. Shen, Q.; Jiang, B.; Shi, P. Active fault-tolerant control against actuator fault and performance analysis of the effect of time delay due to fault diagnosis. *Int. J. Control Autom. Syst.* **2017**, *15*, 537–546. [CrossRef]
30. Shen, Q.; Jiang, B.; Shi, P. Adaptive fault tolerant control against actuator faults. *Int. J. Adapt. Control Signal Process.* **2017**, *31*, 147–162. [CrossRef]
31. Yang, J.-M.; Hammer, J. Asynchronous sequential machines with adversarial intervention: The use of bursts. *Int. J. Control* **2010**, *83*, 956–969. [CrossRef]
32. Kohavi, Z.; Jha, Z. *Switching and Finite Automata Theory*, 3rd ed.; Cambridge University Press: Cambridge, UK, 2010.
33. Xie, G.; Yang, J.; Yang, Y. An improved sparse autoencoder and multilevel denoising strategy for diagnosing early multiple intermittent faults. *IEEE Trans. Syst. Man Cybern. Syst.* **2021**. [CrossRef]
34. Oreifej, R.S.; Al-Haddad, R.; Zand, R.; Ashraf, R.A.; DeMara, R.F. Survivability modeling and resource planning for self-repairing reconfigurable device fabrics. *IEEE Trans. Cybern.* **2017**, *48*, 780–792. [CrossRef] [PubMed]
35. Yang, J.-M.; Kwak, S.W. Fault diagnosis and fault-tolerant control of input/output asynchronous sequential machines. *IET Control Theory Appl.* **2012**, *6*, 1682–1689. [CrossRef]
36. Özveren, C.M.; Willsky, A.S.; Antsaklis, P.J. Stability and stabilizability of discrete event dynamic systems. *J. ACM* **1991**, *38*, 729–751. [CrossRef]
37. Wang, B.; Feng, J.E.; Meng, M. Matrix approach to model matching of composite asynchronous sequential machines. *IET Control Theory Appl.* **2017**, *11*, 2122–2130. [CrossRef]
38. Wang, J.; Han, X.; Chen, Z.; Zhang, Q. Calculating skeleton matrix of asynchronous sequential machines based on the semi-tensor product of matrices. *IET Control Theory Appl.* **2017**, *11*, 2131–2139. [CrossRef]
39. Gao, Z.; Zhu, J.; Han, R.; Xu, Z.; Ullah, A.; Reviriego, P. Design and implementation of configuration memory SEU-tolerant viterbi decoders in SRAM-based FPGAs. *IEEE Trans. Nanotechnol.* **2019**, *18*, 691–699. [CrossRef]