Automatica 50 (2014) 989–993

Contents lists available at ScienceDirect

Automatica

journal homepage: www.elsevier.com/locate/automatica

Technical communique

Fault-tolerant control of a class of asynchronous sequential machines with permanent faults[☆]

Jung-Min Yang[1]

School of Electronics Engineering, Kyungpook National University, 80 Daehakro, Bukgu, Daegu, 702-701, Republic of Korea

ARTICLE INFO

Article history:
Received 2 April 2013
Received in revised form
22 August 2013
Accepted 9 December 2013
Available online 3 February 2014

Keywords:
Asynchronous sequential machines
Corrective control
Fault tolerance
Permanent faults
State feedback

ABSTRACT

In this note, we address a fault-tolerant control scheme for asynchronous sequential machines with permanent faults. The considered asynchronous machine is influenced by faults that change perpetually a portion of its state transition logic. If the asynchronous machine has appropriate analytic redundancy in its reachability, we can design a corrective controller so that the stable-state behavior of the closed-loop system can match that of a reference model despite occurrences of permanent faults. It is assumed that the controller is always fault-free. The existence condition and design procedure for an appropriate controller are presented based on the corrective control scheme. We also provide a controller synthesis example for validating the proposed scheme.

1. Introduction

Due to the unique feature of clockless operation, asynchronous sequential machines are still being employed in many applications notwithstanding the inherent difficulty in their design (Sparsø & Furber, 2001). To guarantee robust and reliable operation, sequential systems should be endowed with the capability of fault diagnosis and tolerance (Hadjicostis, 2004; Paoli, Sartini, & Lafortune, 2011), from which asynchronous sequential machines are not an exception.

This paper presents a fault-tolerant control scheme for a class of asynchronous machines. Our study is based on corrective control that has been developed as a novel and efficient method for controlling asynchronous machines (Murphy, Geng, & Hammer, 2003; Yang, 2011; Yang & Hammer, 2008). The control objective is fulfilled when the input/output behavior of the closed-loop system matches that of a reference model. In the framework of corrective control, only stable states are considered in evaluating model matching between the machine and the reference model, in contrast with the case of synchronous machines (Di Benedetto, Sangiovanni-Vincentelli, & Villa, 2001). In our discussion, we assume that the controller is immune to faults, i.e., fault occurrence is confined to the controlled machine.

The considered asynchronous machine is supposed to have permanent state transition faults, namely a kind of faults that cause the corruption of a state transition at a particular time step and maintain the characteristic of the faulty transition indefinitely. A common instance of such a fault is observed in digital systems working in a space environment, where strong radiation changes the state transition characteristics of the system permanently (Smith & Mostert, 2007). The control objective is to compensate the asynchronous machine so that the stable-state behavior of the closed-loop system matches that of a given reference model, while all the permanent state transition faults are tolerated, meaning that the closed-loop system behaves normally as if no faults occurred. This fault tolerance is made possible if the considered asynchronous machine has potential analytic redundancy in its reachability. We address the existence condition and design procedure for an appropriate fault-tolerant controller based on the corrective control scheme. Compared to critical races (Murphy et al., 2003) and transient faults (Yang & Hammer, 2008), the influence of permanent faults is more serious because once a transition is corrupted by fault, it cannot be used as a segment of any correction trajectory. In other words, besides being the adversarial entity that must be tolerated, a permanent fault also restrains the reachability of the machine, which impedes fault tolerance against subsequent fault occurrences.

The rest of this paper is organized as follows. In Section 2, we introduce a mathematical model of asynchronous machines

[☆] This work was supported in part by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIP) (No. NRF-2012R1A2A2A01003419), and in part by Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education (No. NRF-2010-0007271). The material in this paper was not presented at any conference. This paper was recommended for publication in revised form by Associate Editor Jan Komenda under the direction of Editor André L. Tits.

E-mail address: jmyang@ee.knu.ac.kr.

[1] Tel.: +82 53 950 7235; fax: +82 53 950 5505.

0005-1098/$ – see front matter © 2013 Elsevier Ltd. All rights reserved.
http://dx.doi.org/10.1016/j.automatica.2013.12.034

with permanent faults and the basic configuration of fault-tolerant control. In Section 3, we present the necessary and sufficient condition for the existence of a corrective controller that achieves model matching as well as fault tolerance against permanent state transition faults. In Section 4, we demonstrate the design algorithm of the proposed controller in a case study. Finally, Section 5 concludes the paper.

2. Preliminaries

We focus on a class of asynchronous sequential machines where the present state is given as the output value, namely input/state machines. An input/state asynchronous machine Σ is represented by a four-tuple $\Sigma := (A, X, x_0, f)$, where A is the input set, X is the state set, $x_0 \in X$ is the initial state, and $f : X \times A \to X$ is the state transition function partially defined on $X \times A$. Σ operates according to a recursion of the form

$$x_{k+1} = f(x_k, u_k), \quad k = 0, 1, 2, \ldots,$$

where the current state x_k goes to the next state x_{k+1} asynchronously in response to a switch of the input character to u_k.

A valid state–input pair $(x, v) \in X \times A$ is called a *stable combination* if $f(x, v) = x$; otherwise, it is termed as a *transient combination*. A transient combination (x, v) induces a chain of transient transitions, e.g., $f(x, v) = x_1, f(x_1, v) = x_2, \ldots,$ until it reaches the *next stable state* $x_s = f(x_s, v)$. Due to the lack of a synchronizing clock, these transitions are executed instantaneously. As a result, from an outer user's viewpoint, Σ seems to move directly from (x, v) to (x_s, v). The *stable recursion function* $s : X \times A \to X$ (Kohavi & Jha, 2010) epitomizes this feature of asynchronous machines:

$$s(x, v) := x_s.$$

A chain of transitions from one stable combination to another, as described by s, is called a *stable transition*. s is often extended from input characters to sequences recursively: for $x \in X$, $v \in A$ and $w \in A^+$,

$$s(x, vw) := s(s(x, v), w).$$

A state x' is said to be *stably reachable* from another state x if there exists an input sequence $t \in A^+$ such that $x' = s(x, t)$.

Definition 1. Associated with a valid pair of state and input sequence $(x, t) \in X \times A^+$, define $\gamma(x, t) \subset X \times A$ as the set of all the state–input pairs generated while the input sequence t is applied to Σ at the stable state x.

Denote $t = u_1 u_2 \cdots u_k$ and let $p_1 = s(x, u_1), p_2 = s(p_1, u_2), \ldots, p_k = s(p_{k-1}, u_k) = s(x, t)$ be the intermediate stable states traced by Σ on the way from x. Between p_i and p_{i+1}, Σ may pass through a number of transient combinations, where $i = 0, 1, \ldots, k - 1$ and $p_0 := x$. Assuming that $n(i)$ transient states exist between p_i and p_{i+1}, denote by $p_i^1, p_i^2, \ldots, p_i^{n(i)}$ the corresponding transient states, that is, $f(p_i, u_{i+1}) = p_i^1, f(p_i^1, u_{i+1}) = p_i^2, \ldots, f(p_i^{n(i)}, u_{i+1}) = p_{i+1}$. By definition,

$$\gamma(x, t) = \bigcup_{i=0,\ldots,k-1} \{(p_i, u_{i+1}), (p_i^1, u_{i+1}), \ldots, (p_i^{n(i)}, u_{i+1}), (p_{i+1}, u_{i+1})\}.$$

For describing permanent state transition faults occurring to Σ, we define a set of state–input–state triplets $F \subset X \times A \times X$ as follows:

$$F := \{(z_i, v_i, z_i') | 1 \le i \le r\}, \tag{1}$$

where r denotes the number of elements of F. Under the normal behavior, Σ at a state z_i moves to the next stable state $s(z_i, v_i)$ in response to the input character v_i. However, once a permanent fault happens to Σ and its inner logic governing the transition of (z_i, v_i) degenerates, Σ would be forced to reach an incorrect state $z_i' \neq s(z_i, v_i)$. We call each triplet $(z_i, v_i, z_i') \in F$ a *faulty transition*. We specify that the set F is identified a priori, but the exact moment that each faulty transition occurs is unknown. For later usage, we define

$$E := \{(z_i, v_i) | 1 \le i \le r\} \tag{2}$$

as the set of state–input pairs that can be corrupted by permanent faults.

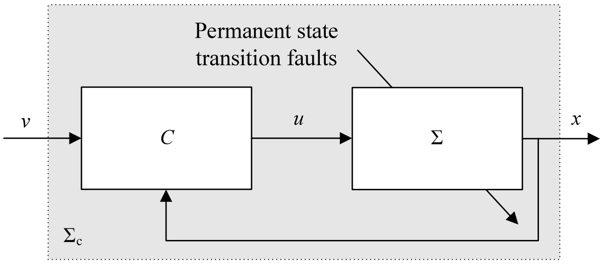

Fig. 1. Basic configuration of fault-tolerant control.

Fig. 1 shows the basic configuration of the fault-tolerant control system used in this note. The considered machine Σ has the permanent fault F. C is the corrective controller also constructed in the form of an asynchronous machine. $v \in A$ is the external input, $u \in A$ is the control input made by C, and $x \in X$ is the current state transmitted to C as the feedback value. Σ_c denotes the asynchronous machine represented by the closed loop. The reference model is given as a stable-state input/state machine $\Sigma' := (A, X, x_0, s')$. Σ' has the same input and state set as those of Σ, and all the states of Σ' are supposed to be fault-free.

We describe model mismatch between Σ and Σ' by a set $D \subset X \times P(A) \times X$, where $P(A)$ is the power set of A:

$$D := \{(y_i, A_i, y_i^m) | 1 \le i \le q\}. \tag{3}$$

Here, q denotes the number of combinations of a state and an input set that incur model mismatch. $(y_i, A_i, y_i^m) \in D$ implies that Σ' has the stable-state behavior $s'(y_i, a) = y_i^m$ for all $a \in A_i$, whereas Σ does not (for some $a \in A_i$, (y_i, a) may be undefined in Σ). y_i and y_j are not necessarily different with each other; if $y_i = y_j, A_i \cap A_j = \varnothing$.

Usually, the model Σ' is specified after Σ is designed. It represents a desirable behavior that should have been included in the original design of Σ, or an improved behavior that Σ must show to carry out a given task. As it depends on the task, Σ' is not unique in general. If Σ has the perfect behavior with respect to nominal transitions, we set $\Sigma' = \Sigma$ except for removing all the permanent faults F from Σ' (in that case, $D = \varnothing$).

The main objective is to propose the existence condition and design procedure for a controller C for which the stable-state behavior of the closed-loop system Σ_c matches that of Σ' despite occurrences of permanent state transition faults. Here *matching behavior* means that Σ_c and Σ' show the identical input/state functioning in terms of stable states (Murphy et al., 2003; Yang, 2011). For excluding unpredictable behavior, we assume that Σ_c abides by the principle of fundamental mode operation (Kohavi & Jha, 2010).

3. Fault tolerance against permanent faults

A corrective controller for model matching exists if and only if for each model mismatch $(y_i, A_i, y_i^m) \in D$, we can find an input sequence w_i that drives Σ from the state y_i to the desired state y_i^m (Murphy et al., 2003), i.e.,

$$\forall i = 1, \ldots, q, \exists w_i \in A^+ \quad \text{s.t.}\ y_i^m = s(y_i, w_i). \tag{4}$$