

비동기 순차 머신의 비-기본모드에서 발생하는 고장 극복을 위한 교정 제어

Fault-Tolerant Corrective Control for Non-fundamental Mode Faults in Asynchronous Sequential Machines

양 정 민*, 곽 성 우**★

Jung-Min Yang*, Seong Woo Kwak**★

Abstract

Fault tolerant corrective control for asynchronous sequential machines (ASMs) with transient faults is discussed in this paper. The considered ASM is vulnerable to a kind of faults whose manifestation may arise during transient transitions of the ASM, leading to transient faults occurring in non-fundamental mode. To overcome adverse effects caused by these faults, we present a novel corrective control scheme that can detect and tolerate transient faults in non-fundamental mode. The existence condition and design algorithm for an appropriate fault tolerant controller is addressed in the framework of corrective control theory. The applicability of the proposed control methodology is demonstrated in the FPGA experiment.

요 약

본 논문은 과도 고장을 가지는 비동기 순차 머신에 대한 내고장성 교정 제어를 다룬다. 본 논문에서 다루는 고장은 머신이 과도 상태 천이를 거칠 때 발현할 수 있으므로 비-기본 모드(non-fundamental mode)에서 원하지 않는 상태 천이를 일으킨다. 본 논문에서는 비동기 순차 머신의 비-기본 모드에서 발생하는 과도 고장을 탐지하고 극복할 수 있는 새로운 내고장성 제어 시스템을 제안한다. 교정 제어 이론의 틀 안에서 내고장성 제어기의 존재 조건과 설계 과정을 제시한다. 또한 제안된 제어 시스템의 효용성을 검증하기 위해 FPGA 실험을 실시한다.

Key words : Asynchronous sequential machines, Corrective control, Fault tolerance, Latent faults, Fundamental mode, FPGA

* Professor, School of Electronics Engineering, Kyungpook National University

** Professor, Department of Control and Instrumentation Engineering, Pukyong National University

★ Corresponding author

E-mail : ksw@pknu.ac.kr, Tel : +82-51-629-6325

※ This research was supported in part by Kyungpook National University Research Fund, 2020, and in part by Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education (NRF-2016R1D1A1B02012959).

Manuscript received Aug. 18, 2020; revised Sep. 15, 2020; accepted Sep. 16, 2020.

This is an Open-Access article distributed under the terms of the Creative Commons Attribution Non-Commercial License(<http://creativecommons.org/licenses/by-nc/3.0>) which permits unrestricted non-commercial use, distribution, and reproduction in any medium, provided the original work is properly cited.

1. 서론

전역 클럭(clock) 없이 동작하는 비동기 순차 머신(asynchronous sequential machine)은 동기 순차 머신에 비해 천이 속도가 빠르다는 장점을 가진다. 비동기 순차 머신은 디지털 시스템의 핵심 부품과 통신 프로토콜 등에서 여전히 많이 활용되고 있다 [1]-[3]. 하지만 클럭이 없기 때문에 비동기 순차 머신은 동기 머신보다 설계하기가 더 어렵다는 단점도 가진다. 교정 제어(corrective control)는 피드백(feedback) 보상 기법을 이용하여 비동기 순차 머신의 안정 상태(stable state) 동작을 개선하는 제어 이론이다. 교정 제어는 특히 비동기 순차 머신에서 발생하는 설계 오류나[4]-[6] 다양한 고장

을 탐지하고 극복하는 데[7]-[9] 우수한 성과를 보였다.

본 논문에서는 비-기본 모드(non-fundamental mode)에서 발생하는 과도 고장(transient fault)을 탐지하고 극복하는 교정 제어 기법을 제안한다. 클럭 없이 동작하는 비동기 순차 머신이 예측 불가능한 결과를 보이지 않으려면 폐루프(closed-loop) 시스템 내에서 두 변수의 값이 동시에 바뀌는 일이 없어야 한다. 이와 같은 조건이 만족되면 시스템은 '기본 모드'를 준수한다고 한다[10]. 내고장성 교정 제어에 대한 선행 연구[7]-[9]에서는 발생하는 과도 고장 역시 기본 모드를 준수한다고 가정하였다. 즉 제어 대상 비동기 순차 머신과 교정 제어기가 모두 안정 상태에 있을 때만 고장이 발생한다고 가정하였다. 하지만 일반적으로 과도 고장은 비동기 순차 머신의 상태에 상관없이 독립적으로 발생하는 특징을 가지므로 이러한 기존 가정은 현실적이지 못하다.

이번 연구에서는 비동기 순차 머신 내에서 최초 발생 후 그 영향이 발현하기까지 시간이 소요되는 잠복 고장(latent fault)[11]이 존재한다고 설정한다. 또 잠복 고장이 발생하면 비동기 순차 머신이 원하지 않는 상태 천이를 한다고 설정한다. 고장 발생 후 고장의 영향이 일정 시간 동안 지속되는 간헐 고장(intermittent fault)[7], [12] 역시 잠복 고장의 한 종류이다. 비동기 순차 머신의 상태를 구현하는 메모리 비트(bit)의 전선 등이 일시적으로 단절된 후 머신이 해당 상태를 지날 때 올바른 상태 값이 출력되지 않는 경우가 잠복 고장에 의한 상태 천이가 발현되는 대표적인 예이다. 또한 고장 발현 순간 머신이 과도 상태 천이를 하고 있었다면 기본 모드가 준수되지 않기 때문에 잠복 고장의 발현은 '비-기본 모드 과도 고장'이라고도 부를 수 있다. 이러한 비-기본 모드 고장을 탐지하고 극복하기 위해서는 기존 연구[7]-[9]와는 다른 문제 설정과 고장 극복 제어 경로가 필요하다.

본 논문의 구성은 다음과 같다. 먼저 II장에서 비동기 순차 머신에 대한 수학적 모델링을 기술하고 비-기본 모드에서 발생하는 과도 고장을 분석한다. III장에서는 교정 제어 이론의 틀 안에서 비-기본 모드 과도 고장을 탐지하고 극복하는 내고장성 제어기의 존재 조건과 설계 과정을 제시한다. 기본 모드 고장의 경우와 비교할 때 비-기본 모드 고장

의 극복을 위해서는 제어 대상 비동기 순차 머신의 도달가능성(reachability)이 더 커져야 함을 입증한다. 제안된 기법의 고장 복구 능력과 응용 가능성을 보이기 위해 IV장에서는 사례 연구 비동기 순차 머신을 설정하고 교정 제어기를 부착한 폐루프 시스템을 FPGA로 구현한 뒤 제어기의 성능을 실험적으로 보인다. 마지막으로 V장에서 본 논문의 결론을 내린다.

II. 비동기 순차 머신 및 고장 사건 모델링

이번 연구에서는 현재 상태가 출력으로 나오는 입력/상태(input/state) 비동기 순차 머신에 대한 교정 제어를 고려한다. 입력/상태 비동기 순차 머신 Σ 를 다음과 같이 정의하자.

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

A는 입력 집합, X는 $|X|=n$ 인 상태 집합, f는 상태 천이 함수이며 $f: X \times A \rightarrow X$ 와 같은 관계를 가진다. A는 다시

$$A=A_n \cup A_d$$

로 분류되는데 A_n 은 정상 입력이며 A_d 는 고장 입력이다.

Σ 의 상태/입력 조합 $(x_0, v) \in X \times A$ ($x_0 \in X, v \in A$)은 f에 따라 안정(stable) 또는 과도(transient) 조합으로 나뉜다. Σ 가 $f(x_0, v)=x_0$ 인 안정 조합을 가지면 입력이 바뀌지 않는 한 현재 상태 x_0 에 계속 머무른다. (x_0, v) 가 $f(x_0, v) \neq x_0$ 인 과도 조합이라면 Σ 는 x_0 에서 시작하여 일련의 과도 상태

$$f(x_0, v)=x_1, f(x_1, v)=x_2, \dots, f(x_{k-1}, v)=x_k$$

를 지나 $f(x_k, v)=f(x_{k-1}, v)=x_k$ 인 다음 안정 상태(next stable state) x_k 에 도달한다($1 \leq k < n-1$). Σ 가 과도 상태를 지나는 시간이 아주 짧기 때문에 보통 과도 상태를 생략하고 안정 상태 간의 동작만으로 Σ 의 동작을 표현한다. 이를 위해 stable recursion 함수 [4] $s: X \times A \rightarrow X$ 를 다음과 같이 정의한다.

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

여기서 x_k 는 (x, v) 의 다음 안정 상태이다. s의 정의구역은 아래와 같이 길이 2 이상의 입력 스트링

(string)으로 확장 가능하다.

$$s(x, v_1 v_2 \dots v_k) = s(s(x, v_1), v_2 \dots v_k), \quad v_1 v_2 \dots v_k \in A_n^+$$

여기서 A_n^+ 는 A_n 의 원소들로 구성된 공집합이 아닌 입력 스트링의 집합이다.

또 Σ 가 과도 조합 (x, v) 에서 시작하여 다음 안정 상태에 도달하기까지 거치는 모든 과도 상태를 $\tau(x, v) \subset X$ 로 표시한다. 앞의 예에서

$$\tau(x_0, v) = \{x_0, x_1, \dots, x_{k-1}\}$$

이다.

그림 1은 비-기본 모드 고장이 존재하는 비동기 순차 머신 Σ 에 대한 교정 제어 시스템을 나타낸 것이다. C 는 교정 제어기이며 $v \in A_n$ 은 외부 입력, $u \in A_n$ 은 C 가 생성하는 제어 입력, $x \in X$ 는 상태 피드백이고 $w \in A_d$ 는 과도 고장 입력이다. Σ 와 C 가 결합된 폐루프 시스템을 Σ_c 라 한다. Σ 에서 일어나는 과도 고장 입력은 주로 현재 상태 x 에 따라서 결정된다. Σ 가 상태 x 를 가질 때 발생할 수 있는 모든 과도 고장 입력을

$$W(x) \subset A_d$$

라고 정의하자. 또 $X' \subset X$ 인 임의의 상태 집합 X' 에 대해

$$W(X') = \bigcup_{x \in X'} W(x)$$

로 정의한다.

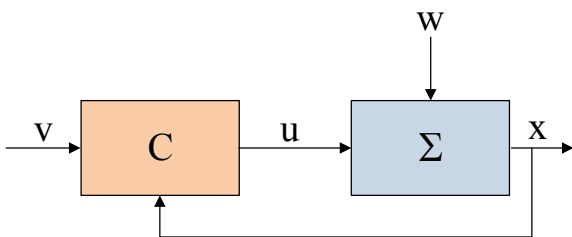


Fig. 1. Corrective control system for an input/state asynchronous sequential machine with non-fundamental mode faults.
그림 1. 비-기본 모드 고장이 존재하는 입력/상태 비동기 순차 머신에 대한 교정 제어 시스템

기본 모드에서 발생하는 과도 고장은 선행 연구 [7-9]에서 충분히 다루었기 때문에 이번 연구에서는 모든 과도 고장이 비-기본 모드에서만 일어난다고 설정한다. 기본 모드를 준수하는 과도 고장은 Σ 가 안정 상태에 있을 때만 발생한다. 반면 비-기

본 모드 고장은 고장 발생 시 Σ 가 과도 상태 천이를 하고 있어야 성립된다. 따라서 고장 발생 후의 상황을 기술하기 위해 앞에서 든 예와 같이 Σ 가 과도 조합 (x_0, v) 에서 시작한 과도 상태 천이를 하고 있다고 가정하자. 이때 Σ 가 다음 안정 상태 $s(x_0, v)$ 에 도달할 때까지 거치는 중간 과도 상태 집합은 $\tau(x_0, v) = \{x_0, x_1, \dots, x_{k-1}\}$ 이므로 비-기본 모드 고장의 발생은 $W(\tau(x_0, v))$ 에 속한 임의의 고장 입력 w 가 들어왔음을 의미한다. 여기서 $x_i \in \tau(x_0, v)$ 이고 $(0 \leq i \leq k-1)$ $w \in W(x_i)$ 인 w 가 발생했다고 하자. Σ 는 원래 $s(x_0, v)$ 에 도달해야 하지만 w 가 발생하면 대신 원하지 않는 상태 $s(x_i, w)$ 로 강제 천이한다.

그림 1에서 알 수 있듯이 C 는 v 와 x 를 입력 받아 u 를 생성한다. 따라서 C 는 아래와 같은 입력/출력 (input/output) 비동기 순차 머신으로 모델링된다.

$$C = (A_n \times X, A_n, \varepsilon, \xi_0, \phi, \eta)$$

여기서 $A_n \times X$ 는 입력 집합, A_n 는 출력 집합이며 ε 는 상태 집합, $\xi_0 \in \varepsilon$ 는 초기 상태, $\phi: \varepsilon \times A_n \times X \rightarrow \varepsilon$ 와 $\eta: \varepsilon \rightarrow A_n$ 는 각각 상태 천이 함수와 출력 함수이다.

C 의 역할은 과도 고장을 탐지하고 Σ 를 정상 동작으로 즉시 복귀시키는 일이다. Σ 가 고장 없이 정상 동작한다면 C 는 외부 입력 v 를 그대로 제어 입력 $u = v$ 로 전달한다. 만약 비-기본 모드 과도 고장이 발생하면 C 는 고장 발생을 탐지하고 적절한 제어 입력 시퀀스(sequence)를 생성하여 Σ 를 정상 상태로 즉시 복귀시킨다. C 와 Σ 모두 비동기 순차 머신이므로 고장 탐지와 복구 과정은 비동기적으로 구현될 수 있다. 따라서 고장 극복 교정 제어가 성공된다면 외부 사용자들에게는 Σ_c 가 고장 없이 정상 운용되는 것처럼 관측된다.

III. 내고장성 교정 제어기

교정 제어기 C 에 의한 고장 탐지를 먼저 설명하기 위해 앞에서와 마찬가지로 Σ 가 (x_0, v) 에서 출발하여 $\tau(x_0, v) = \{x_0, x_1, \dots, x_{k-1}\}$ 를 거치면서 다음 안정 상태 $s(x_0, v) = x_k$ 에 도달하는 상태 천이를 하고 있다고 가정하자. 그림 1에서 Σ 가 (x_0, v) 에서 상태 천이를 시작한 후 C 는 상태 피드백 x 의 변화를 관측하면서 Σ 의 동작을 추정한다. Σ 가 고장 발생 없이 정상 천이를 했다면 x 의 값은 x_1, x_2, \dots 를 거쳐 x_k 까지 변한다. 상태 피드백 x_k 를 받는 순간 C 는 Σ 가

(x_0, v)에서 (x_k, v)까지의 상태 천이를 완료했음을 인지한다.

한편 Σ 의 상태 천이 도중 $W(\tau(x_0, v))$ 에 속한 임의의 w 가 들어와 비-기본 모드 과도 고장이 발생하여 Σ 가 원하지 않는 상태 천이를 한다고 하자. $x_i \in \tau(x_0, v)$ 이고 $w \in W(x_i)$ 라면 w 의 발생으로 Σ 가 도달하는 (원하지 않는) 안정 상태는 $s(x_i, w)$ 이다. C 는 정상적인 다음 안정 상태 x_k 가 아니라 $s(x_i, w)$ 를 상태 피드백으로 받는 순간 비-기본 모드 과도 고장이 발생했음을 인지한다. 그런데 $s(x_i, w)$ 가 $\tau(x_0, v)$ 에 속한다면 고장 탐지 과정에서 문제가 생긴다. 위에서 Σ 가 (x_0, v)에서 (x_k, v)까지 정상적인 상태 천이를 할 때 $\tau(x_0, v)$ 에 속하는 과도 상태를 거친다고 하였다. 그렇다면 고장에 의해서 생기는 상태 피드백 $s(x_i, w)$ 를 C 가 받는 순간 아래 두 가지 상황 중 어떤 일이 일어났는지를 구분하지 못한다.

- (i) Σ 가 (x_0, v)에서 시작하여 정상적인 상태 천이를 하고 있으며 중간 과도 상태 $s(x_i, w)$ 를 거치는 중이다.
- (ii) Σ 가 (x_0, v)에서 상태 천이를 하는 도중 비-기본 모드 과도 고장이 발생하여 원하지 않는 다음 안정 상태 $s(x_i, w)$ 에 도달하였다.

따라서 C 가 비-기본 모드 과도 고장을 정확하게 탐지하기 위해서는 고장 발생으로 인해 Σ 가 도달하는 안정 상태가 Σ 가 정상적인 상태 천이 과정에서 거치는 중간 과도 상태 집합에 속하지 말아야 한다. 이 조건을 표현하기 위해 (x_0, v)에서 발생하는 비-기본 모드 과도 고장에 의해서 Σ 가 도달하는 원하지 않는 안정 상태 집합을 $\Gamma(x_0, v)$ 라고 하자. $\Gamma(x_0, v)$ 는 다음과 같이 정의된다.

$$\Gamma(x_0, v) = \{s(x_i, w) \mid x_i \in \tau(x_0, v), w \in W(x_0)\}$$

정리 1. Σ 가 $W(\tau(x_0, v)) \neq \emptyset$ 인 과도 조합 (x_0, v)에서 상태 천이를 할 때 비-기본 모드 과도 고장이 발생한다고 하자. 그림 1의 교정 제어기 C 가 (x_0, v)에서 발생하는 임의의 비-기본 모드 과도 고장을 탐지하기 위한 조건은 다음과 같다.

$$\Gamma(x_0, v) \cap \tau(x_0, v) = \emptyset \tag{1}$$

(x_0, v)에서 정리 1의 조건이 만족된다고 가정하고 비-기본 모드 과도 고장에 대한 내고장성 교정 제어기 존재 조건을 기술한다. 고장이 발생하면 위에

서 정의한 대로 Σ 는 $x' \in \Gamma(x_0, v)$ 인 임의의 상태 x' 로 원하지 않는 천이를 한다. 그런데 Σ 는 정상 동작이 계속되었다면 다음 안정 상태 $s(x_0, v)$ 에 도달해야 한다. 따라서 고장 복구가 되려면 고장이 탐지되는 즉시 Σ 를 x' 에서 $s(x_0, v)$ 로 보내야 한다. 선행 연구[4]-[9]에서 Σ 를 x' 에서 $s(x_0, v)$ 로 보내는 동적 교정 제어기가 존재할 조건은 Σ 가 x' 에서 $s(x_0, v)$ 까지 안정적인 도달가능성(stable reachability)을 가져야 한다고 기술되었다. 본 논문에서도 이 선행 조건을 그대로 활용하여 아래와 같이 표현한다.

$$\exists t_{x'} \in A^+_n : s(x', t_{x'}) = s(x_0, v)$$

위 조건을 만족시키는 입력 스트링(string) $t_{x'}$ 을 이용하여 교정 제어기 C 를 설계한다. 아래 정리는 본 논문의 주요 연구 결과이다.

정리 2. Σ 가 $W(\tau(x_0, a)) \neq \emptyset$ 인 과도 조합 (x_0, a)에서 상태 천이를 할 때 발생하는 비-기본 모드 과도 고장을 극복하는 그림 1의 교정 제어기 C 가 존재할 필요충분조건은 다음과 같다.

$$\forall x' \in \Gamma(x_0, a), \exists t_{x'} \in A^+_n : s(x', t_{x'}) = s(x_0, a) \tag{2}$$

증명. (\Leftarrow) 교정 제어기 C 가 초기 상태 ξ_0 에 있을 때 Σ 가 안정 상태 x_0 에 진입하면 C 는 transition 상태[4],[6] ξ_r 로 천이한다. 이때 외부 입력이 a 로 바뀐다고 하자. 아직 고장 발현이 되지 않았기 때문에 C 는 a 를 제어 입력 $u=a$ 로 전달한다. a 를 받은 Σ 는 (x_0, a)에서 상태 천이를 시작한다. 만약 비-기본 모드 과도 고장이 발생하지 않은 채 천이가 완료된다면 상태 피드백 $s(x_0, a)$ 가 C 에 전달된다. $s(x_0, a)$ 를 받은 C 는 초기 상태 ξ_0 로 복귀한다.

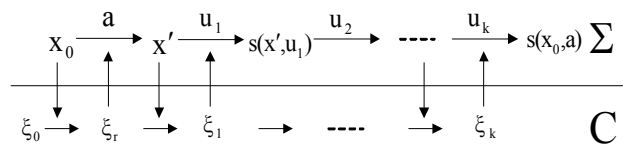


Fig. 2. Correction procedure between C and Σ .
그림 2. C 와 Σ 간의 교정 제어 과정

그런데 상태 천이를 하는 도중 비-기본 모드 과도 고장이 발생하여 Σ 가 정상적인 다음 안정 상태 $s(x_0, a)$ 대신 $x' \in \Gamma(x_0, v)$ 인 임의의 상태 x' 로 천이한다고 하자. 정리 1의 고장 탐지 조건이 만족된다

고 가정했기 때문에 C가 상태 피드백 x' 을 받는 순간 비-기본 모드 과도 고장 발생을 인지한다. 또한 가정에 의해서 x' 에서 $s(x_0,a)$ 까지의 도달가능성이 보장되므로 $s(x',t_{x'})=s(x_0,a)$ 인 입력 스트링

$$t_{x'}=u_1u_2\cdots u_k \in A_n^+$$

가 존재한다. C는 $t_{x'}$ 를 이용하여 x' 에서 $s(x_0,a)$ 까지의 제어 경로를 구성한다. 이를 위해 먼저 k개의 보조 상태(auxiliary state) $\xi_1, \dots, \xi_k \in \Sigma$ 를 정의한다. C는 상태 피드백 x' 을 받고 ξ_1 로 천이한 다음 첫 번째 제어 입력 u_1 을 생성한다. u_1 을 받은 Σ 는 $s(x',u_1)$ 로 상태 천이한다. $s(x',u_1)$ 을 상태 피드백으로 받은 C는 ξ_1 에서 ξ_2 로 천이한 다음 두 번째 제어 입력 u_2 를 생성한다. u_2 를 받은 Σ 는 $s(x',u_1u_2)$ 로 상태 천이한다. 이러한 C와 Σ 의 상호 동작이 k번 반복되면 Σ 는 목표 상태 $s(x_0,a)$ 로 천이하고 고장 복구가 완료된다. C와 Σ 모두 비동기 순차 머신이므로 상기한 제어 과정은 순식간에 진행되며 결과적으로 페루프 시스템 Σ_c 는 고장 없는 정상 동작을 수행하는 것처럼 관측된다. 그림 2는 C와 Σ 간의 교정 제어 과정을 도시한 그림이다.

(\Rightarrow) 식 (2)가 내고장성 교정 제어기 C가 존재할 충분조건이라는 사실은 선행 연구[4]-[6]에서 나와 있으므로 증명을 생략한다.

위 정리에 나온 제어기 존재 조건 (2)를 보면 기본 모드 고장과 비교하여 비-기본 모드 고장을 극복하기 위해서 더 큰 도달가능성이 필요하다는 주장이 입증된다. 이전 연구[7]-[9]에서 논한 기본 모드 고장은 Σ 가 모두 안정 상태에 있을 때만 발생하므로 고장이 발생하는 순간 Σ 에 가해진 입력이 어떤 값인지는 중요하지 않았다. 하지만 (2)에서 알 수 있듯이 비-기본 모드 고장은 고장이 발생하는 상태 x_0 뿐만 아니라 입력 a에 의해서도 목적 상태 $s(x_0,a)$ 가 결정된다. 다시 말하면 a의 값이 바뀌면 목적 상태도 바뀐다. 따라서 x_0 과 과도 조합을 이루는 모든 입력 a에 대해서 고장 탐지 및 극복이 가능해야 하므로 기본 모드 고장에 비해 더 까다로운 제어기 존재 조건이 필요한 것이다.

IV. FPGA 실험

제한한 내고장성 교정 제어기의 응용 가능성을

입증하기 위해 FPGA 실험을 실시한다. 그림 3은 실험 연구에서 사용된 비동기 순차 머신 $\Sigma=(A,X,f)$ 이다. Σ 는 다음과 같은 입력 및 상태 집합을 가진다.

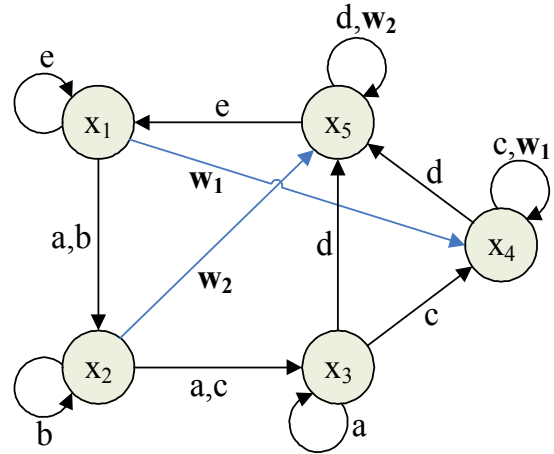


Fig. 3. Input/state asynchronous sequential machine Σ .
그림 3. 입력/출력 비동기 순차 머신 Σ

$$A_n=\{a, b, c, d, e\},$$

$$A_d=\{w_1, w_2\}$$

$$X=\{x_1, x_2, x_3, x_4, x_5\}$$

상태 x_1 과 x_2 에서 고장 입력 w_1 과 w_2 가 각각 발생할 수 있으므로

$$W(x_1)=\{w_1\}, W(x_2)=\{w_2\},$$

$$W(x_i)=\emptyset, \forall i=3, \dots, 5$$

이다. 따라서 x_1 과 x_2 가 이루는 모든 과도 조합에서 비-기본 모드 고장이 발생 가능하다. 그러한 과도 조합을 그림 3에서 찾으면

$$\{(x_1,a), (x_1,b), (x_2,a), (x_2,c)\}$$

이다.

위의 네 개의 과도 조합 중 사례 연구로서 (x_1,a) 에 대해 고장 탐지 가능성과 내고장성 제어기 존재 가능성을 조사한다. 먼저 Σ 가 (x_1,a) 에서 출발하는 상태 천이를 할 때 거치는 중간 과도 상태는

$$\tau(x_1,a)=\{x_1, x_2\}$$

이고

$$W(\tau(x_1,a))=\{w_1, w_2\}$$

이다. 또 그림 3에서 $s(x_1,w_1)=x_4, s(x_2,w_2)=x_5$ 이므로

$$\Gamma(x_1,a) = \{x_4, x_5\}$$

이다. 그런데

$$\Gamma(x_1,a) \cap \tau(x_1,a) = \{x_4, x_5\} \cap \{x_1, x_2\} = \emptyset$$

이므로 조건 (1)이 만족된다. 따라서 정리 1에 의해서 (x_1,a) 에서 발생하는 모든 비-기본 모드 과도 고장은 그림 1의 교정 제어 시스템에서 탐지 가능하다.

다음으로 내고장성 교정 제어기의 존재 조건을 조사한다. $s(x_1,a) = x_3$ 이므로 Σ 가 $\Gamma(x_1,a) = \{x_4, x_5\}$ 에 속한 각 상태에서부터 x_3 까지 안정적으로 도달가능한지를 보면 된다. 그림 3의 상태 흐름도를 보면

$$s(x_4,dea) = x_3$$

$$s(x_5,ea) = x_3$$

이므로 조건 (2)가 만족된다. 따라서 정리 2에 의해 (x_1,a) 에서 발생하는 비-기본 모드 과도 고장을 극복하는 교정 제어기가 설계 가능하다. 또 위의 입력 스트링 dea와 ea는 고장 복구를 위한 피드백 경로 구축에 활용될 수 있다. (x_1,a) 의 경우와 유사한 방법으로 조사하면 나머지 과도 조합에서도 비-기본 모드 과도 고장에 대한 탐지 및 극복이 모두 가능하다는 사실을 입증할 수 있다.

그림 3의 Σ 와 교정 제어기 C가 결합된 페루프 시스템 Σ_c 를 VHDL로 구현한 후 FPGA에서 실험을 실시하였다. 실험에 사용된 소프트웨어는 Quartus® II (ver. 9.1)이며, 타겟(target) FPGA는 EP1C6Q240C8이다.

그림 4와 그림 5는 그림 3의 사례 연구 시스템에서 비-기본 모드에서 발생하는 과도 고장을 교정 제어기 C를 이용하여 복구하는 실험 결과이다. 그림 4는 (x_1,a) 에서 시작한 상태 천이 도중 발생하는 w_1 에 의한 비-기본 모드 과도 고장을 극복하는 교정 제어 과정을 보여준다. Σ 의 각 상태는 3비트 이진수 $b_2b_1b_0$ 로 나타내었다. x_1 은 $b_2b_1b_0=000$, x_2 는 $b_2b_1b_0=001$, x_3 은 $b_2b_1b_0=011$, x_4 는 $b_2b_1b_0=111$, x_5 는 $b_2b_1b_0=101$ 로 할당된다.

그림 4에서 Σ 가 초기 상태 $x_1(000)$ 에 머물러 있을 때 시간 $t_1=80ns$ 에 외부 입력 a가 입력되면 $x_2(001)$ 를 거쳐 $x_3(011)$ 에 도달하기 위한 상태 천이를 시작한다.

하지만 상태 천이를 시작하자마자 비-기본 모드 고장 w_1 에 의해 머신은 $t_2=89ns$ 에서 $x_4(111)$ 로 강제 천이하였다. $s(x_1,w_1) = x_4$ 이므로 교정 제어기 C는 w_1

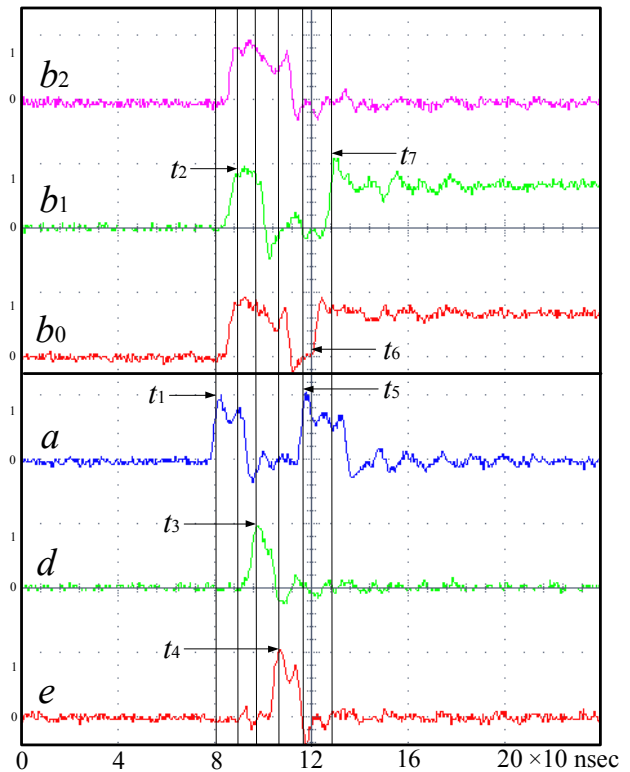


Fig. 4. Experimental result of corrective control tolerating non-fundamental mode fault by w_1 that occurs in the stable transition from (x_1,a) .

그림 4. (x_1,a) 에서 시작한 상태 천이 도중 발생하는 w_1 에 의한 비-기본 모드 과도 고장 극복하는 교정 제어 실험 결과

이 발생하였음을 탐지하고 입력 스트링 $t_{x4}=dea$ 를 이용하여 고장 복구를 실시한다. 먼저 C는 시간 $t_3=97ns$ 에서 첫 번째 제어 입력 d를 발생시킨다. d를 받은 Σ 는 $x_5(101)$ 로 천이하며, 시간 $t_4=106ns$ 에서 C는 두 번째 제어 입력 e를 발생시킨다. e에 의해 Σ 가 x_1 로 천이하면 세 번째 입력 a가 전달된다(시간 $t_5=117ns$). Σ 는 상태 x_2 (시간 t_6)를 거쳐 시간 $t_7=129ns$ 에서 목표 상태 x_3 에 도달하고 고장 복구 과정은 종료된다. 고장 탐지에서 복구에 걸리는 총 시간은 $t_7-t_2=40ns$ 로 매우 짧기 때문에 비동기적 교정 제어가 성공했음을 알 수 있다.

그림 5는 (x_1,a) 에서 시작한 상태 천이 도중 발생하는 w_2 에 의한 비-기본 모드 과도 고장을 극복하는 교정 제어 과정이다. Σ 가 초기 상태 x_1 에 머물러 있을 때 시간 $t_8=80ns$ 에서 외부 입력 a가 입력되면 앞서와 같이 x_2 를 거쳐 x_3 에 도달하기 위한 상태 천이를 시작한다. 하지만 중간 상태 x_2 (시간 $t_9=87ns$)에서 비-기본 모드 고장 w_2 에 의해 머신은

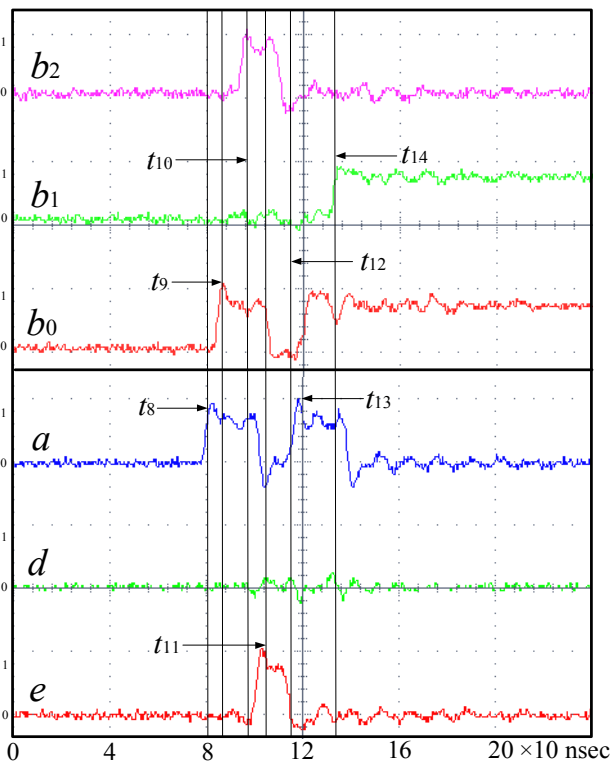


Fig. 5. Experimental result of corrective control tolerating non-fundamental mode fault by w_2 that occurs in the stable transition from (x_1, a) .

그림 5. (x_1, a) 에서 시작한 상태 천이 도중 발생하는 w_2 에 의한 비-기본 모드 과도 고장 극복하는 교정 제어 실험 결과

상태 x_5 로 강제 천이한다(시간 $t_{10}=96\text{ns}$). $s(x_2, w_2)=x_5$ 이므로 상태 피드백 x_5 를 받은 교정 제어기 C는 w_2 가 발생하였음을 탐지하고 입력 스트링 $t_{x_5}=ea$ 를 이용하여 고장 복구를 시작한다. 시간 t_{11} 과 t_{13} 에서 제어 입력 e 와 a 를 각각 발생시키면 Σ 는 x_0 를 거쳐 목표 상태 x_3 에 도달한다(시간 $t_{14}=133\text{ns}$). 이번에도 고장 복구에 걸리는 시간은 $t_{14}-t_{10}=37\text{ns}$ 로 매우 짧게 걸렸다. 그림 4와 그림 5의 실험 결과는 본 논문에서 제안한 교정 제어기가 비-기본 모드에서 발생하는 과도 고장을 매우 짧은 시간 내에서 비동기적으로 복구할 수 있음을 보여준다.

V. 결론

본 논문에서는 입력/상태 비동기 순차 머신에서 발생하는 비-기본 모드 과도 고장을 탐지하고 극복하는 방법을 제안하였다. 기본 모드 고장에 비해 비-기본 모드 과도 고장은 머신이 고장 후 도달하는 상태가 다르며, 고장 극복을 위해서도 기본 모

드 고장보다 더 큰 도달가능성이 요구되었다. 이번 연구에서는 상태 피드백을 이용하여 비-기본 모드 과도 고장을 탐지하고 극복하는 방법을 교정 제어 이론의 틀에서 제안하였다. 또 VHDL과 FPGA 상에서 구현된 사례 연구 시스템의 내고장성 제어 문제에 제안된 기법을 적용하였다. 실험 결과는 제안된 교정 제어기를 이용하여 비-기본 모드에서 발생하는 과도 고장을 효과적으로 극복할 수 있음을 입증한다.

References

[1] J. Sparsø and S. Furber, *Principles of Asynchronous Circuit Design—A Systems Perspective*, Boston, MA: Kluwer Academic Publishers, 2002.

[2] I. H. Park, H. K. Lee, and S. J. Kang, “An energy-efficient asynchronous sensor MAC protocol design for wireless sensor networks,” *Journal of IKEEE*, vol.16, no.2, pp.86–94, 2012. DOI: 10.7471/ikeee.2012.16.2.086

[3] E. J. Youn and Y. C. Jang, “A 10-bit 20-MS/s asynchronous SAR ADC using self-calibrating CDAC,” *Journal of IKEEE*, vol.23, no.1, pp.35–43, 2019. DOI: 10.7471/ikeee.2019.23.1.35

[4] T. E. Murphy, X. Geng, and J. Hammer, “On the control of asynchronous machines with races,” *IEEE Transactions on Automatic Control*, vol.48, no.6, pp.1073–1081, 2003. DOI: 10.1109/TAC.2003.812814

[5] N. Venkatraman and J. Hammer, “On the control of asynchronous sequential machines with infinite cycles,” *International Journal of Control*, vol.79, no.7, pp.764–785, 2006. DOI: 10.1080/00207170600665022

[6] J. Peng and J. Hammer, “Input/output control of asynchronous sequential machines with races,” *International Journal of Control*, vol.83, no.1, pp. 125–144, 2010. DOI: 10.1080/00207170903104182

[7] J.-M. Yang and S. W. Kwak, “Fault tolerance in switched ASMs with intermittent faults,” *IET Control Theory & Applications*, vol.11, no.9, pp. 1443–1449, 2017. DOI: 10.1049/iet-cta.2016.1327

[8] J.-M. Yang and S. W. Kwak, “Model matching

and fault-tolerant control of switched asynchronous sequential machines with transient faults," *IET Control Theory & Applications*, vol.13, no.12, pp. 1882-1890, 2019. DOI: 10.1049/iet-cta.2018.6046

[9] J.-M. Yang, "Robust corrective control of asynchronous sequential machines with control input and feedback faults," *Automatica*, vol.107, pp.605-609, 2019.

DOI: 10.1016/j.automatica.2019.06.026

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

[11] S. H. Zad, R. H. Kwong, and W. M. Wonham, "Fault diagnosis in discrete event systems: framework and model reduction," *IEEE Transactions on Automatic Control*, vol.48, no.7, pp.1199-1212, 2003.

DOI: 10.1109/TAC.2003.814099

[12] A. A. Ismaeel and R. Bhatnagar, "Test for detection & location of intermittent faults in combinational circuits," *IEEE Transactions on Reliability*, vol.46, no.2, pp.269-274, 1997.

DOI: 10.1109/24.589956

BIOGRAPHY

Jung-Min Yang (Member)



1993 : BS degree in Electrical Engineering, KAIST.

1995 : MS degree in Electrical Engineering, KAIST.

1999 : PhD degree in Electrical Engineering, KAIST.

2013~present : Professor, School of Electronics

Engineering, Kyungpook National University

Seong Woo Kwak (Member)



1993 : BS degree in Electrical Engineering, KAIST.

1995 : MS degree in Electrical Engineering, KAIST.

2000 : PhD degree in Electrical Engineering, KAIST.

2003~2020 : Professor, Department of Electronics Engineering, Keimyung University

2020~present : Professor, Department of Control and Instrumentation Engineering, Pukyong National University