

논문 2014-51-8-15

병렬 결합된 비동기 순차 머신을 위한 교정 제어

(Corrective Control of Composite Asynchronous Sequential Machines in Parallel Connection)

양 정 민*

(Jung-Min Yang[©])

요 약

이번 연구에서는 병렬 결합된 두 개의 비동기 순차 머신에 대한 교정 제어 문제를 다룬다. 각 비동기 머신은 동일한 외부 입력을 받아 서로 독립적인 상태 천이 특성을 보인다. 본 논문에서는 한 개의 교정 제어기만을 이용하여 두 개의 비동기 머신의 페루프 시스템 동작을 각각의 기준 모델의 동작과 일치시키도록 하는 제어 방법을 제안한다. 본 방법은 교정 제어기 두 개를 사용해야 하는 기존 방법에 비해 제어기 크기 및 계산량을 줄일 수 있다. 본 논문에서는 대상 비동기 머신이 입력/상태 유형이라고 설정하고 상태 피드백 교정 제어기가 존재할 조건과 설계 과정을 제시한다. 또 예제 시스템을 통하여 제안된 기법의 적용 과정을 기술한다.

Abstract

We address the problem of corrective control for two asynchronous sequential machines in parallel connection. Each asynchronous machine receives the same external input and shows independent state transition characteristics. We propose a novel control scheme in which only one corrective controller is employed so as to make the closed-loop system of each machine match the behavior of the corresponding reference model. Compared with the former method utilizing two corrective controllers, our scheme can reduce the controller size and computational load in controller design. We present the existence condition and design procedure for a state-feedback corrective controller under the assumption that the controlled machines are of input/state type. The design procedure for the proposed controller is described in an illustrative example.

Keywords : Asynchronous Sequential Machines, Corrective Control, State Feedback, Parallel Connection

I. 서 론

교정 제어(corrective control)는 피드백(feedback) 제어 구조를 통해 주어진 비동기 순차 머신(asynchronous sequential machine)의 정상 상태(steady state) 천이 특

성을 바꾼다. 교정 제어는 2000년대 초반 이론적 체계가 정립된 후^[1] 비동기적으로 동작하는 디지털 시스템의 고장 극복 문제를 해결하는 데 주로 적용되었다^[2-4]. 교정 제어에서 사용하는 시스템 모델링은 이산 사건 시스템(discrete event systems)의 관리 제어(supervisory control)^[5]와 비슷하다. 하지만 관리 제어는 형식 언어(formal language) 이론을 기반으로 제어 입력을 생성하는데 반해 교정 제어는 입력과 피드백 값을 비교하여 다음 제어 입력을 결정하는 전통적인 자동 제어(automatic control)의 방법을 사용하므로 두 이론 사이에는 근본적인 차이가 존재한다.

본 논문에서는 병렬 결합(parallel connection)된 두

* 정회원, 경북대학교 전자공학부
(School of Electronics Engineering, Kyungpook National University)

© Corresponding Author (E-mail: jmyang@ee.knu.ac.kr)

※ 이 논문은 2013학년도 경북대학교 신입교수정착연구비에 의하여 연구되었음

접수일자: 2014년04월15일, 수정일자: 2014년07월15일

수정완료: 2014년07월30일

개의 비동기 순차 머신으로 이루어진 복합 시스템에 대한 교정 제어기를 제안한다. 두 개의 비동기 순차 머신은 동일한 외부 입력을 받아서 독립적인 동작을 수행한다. 제어 목적은 각 머신의 폐루프(closed-loop) 시스템 동작을 원하는 기준 모델(reference model)의 동작과 일치시키도록 하는 제어기를 꾸미는 일이다(이러한 문제를 ‘모델 정합(model matching)’이라 부른다). 병렬 결합은 복합 디지털 시스템을 만드는 기본적인 연결 요소 중의 하나이며 병렬 결합되어 비동기적으로 움직이는 복합 시스템도 현재 많이 존재하므로^[6] 본 연구 주제의 응용 가능성은 크다고 말할 수 있다. 참고로 직렬 결합된 복합 시스템에 대한 교정 제어는 저자의 선행 연구^[7]에서 다루었다.

병렬 결합된 복합 이산 사건 시스템을 제어하는 연구 결과는 많지 않다. 현재 교정 제어 분야에서 병렬 결합된 복합 시스템을 다룬 연구는 발표되지 않았다. 이산 사건 시스템 관리 제어 분야에서는 하나의 대상 시스템을 여러 개의 제어기가 지역적(locally)으로 제어하는 decentralized 제어가 주로 관심을 받아 왔다^[8]. 최근 많이 연구되고 있는 이산 사건 시스템 모델의 하나인 불리언 네트워크(Boolean network) 분야에서는 여러 개의 부(sub)-시스템의 특성을 번갈아 가면서 가지는 스위칭 불리언 네트워크(switched Boolean network)에 대한 안정화(stabilization) 연구가 많이 이루어지고 있다^[9]. 하지만 스위칭 시스템은 두 개 이상의 독립된 부-시스템이 동시에 움직이는 병렬 결합 복합 시스템과는 동특성이 다르다.

이번 연구의 핵심은 한 개의 교정 제어기만으로 병렬 결합된 두 개의 비동기 순차 머신이 각각의 모델 정합 목적을 이루게 하는 제어 구조를 제안하는 것이다. 이 문제를 이전 연구^[1]의 방법을 사용하여 푼다면 i) 복합 비동기 머신에 대한 상태 천이 함수를 모두 구한 후 단일 비동기 머신으로 생각해서 교정 제어기를 구하거나, ii) 각 머신에 대한 제어를 담당하는 두 개의 교정 제어기를 따로따로 설계해야 한다. 이번 연구에서는 i)과 ii)의 경우를 모두 피해서 하나의 교정 제어기만을 이용하여 복합 비동기 머신의 상태 천이 함수를 구할 필요 없이 두 비동기 머신의 모델 정합 문제를 해결하도록 한다. 제안된 교정 제어기가 존재할 조건을 해석적으로 규명한 후 제어기 설계 과정을 제시한다. 또한 예제를 통해서 제안된 제어 기법의 적용 과정을 보인다.

II. 문제 설정

1. 병렬 결합된 복합 비동기 머신

병렬 결합된 두 개의 비동기 순차 머신을 M_1 과 M_2 라 하자. M_1 과 M_2 는 현재 상태 값이 출력으로 나오는 입력/상태 형태(input/state type)라고 설정한다. 유한 상태 머신(finite-state machine)으로 M_1 과 M_2 를 모델링하면 다음과 같다.

$$\begin{aligned} M_1 &= (A, X, x_0, f_1) \\ M_2 &= (A, Y, y_0, f_2) \end{aligned} \quad (1)$$

A 는 두 머신에 공통으로 들어가는 입력 집합이다. 식 (1)의 M_1 과 M_2 가 병렬 결합을 이루어 항상 동일한 입력 신호를 받기 때문에 입력 집합 A 를 공유한다. X 와 Y 는 두 머신의 상태 집합, $x_0 \in X$ 와 $y_0 \in Y$ 초기 상태(initial state)이며, $f_1: X \times A \rightarrow X$ 와 $f_2: Y \times A \rightarrow Y$ 는 M_1 과 M_2 의 상태 천이 함수(state transition function)이다.

전역 클럭(clock) 없이 동작하는 비동기 머신은 안정 상태(stable state)에 머무르다 입력의 변화에 따라 다른 안정 상태로 천이한다. 일반적으로 머신은 두 안정 상태 사이를 천이할 때 여러 개의 과도 상태(transient state)를 거친다. 머신이 과도 상태에 머무르는 시간이 극히 짧기 때문에 외부 사용자는 머신이 안정 상태에서 다음 안정 상태로 천이하는 운동만 관측한다. M_1 과 M_2 의 안정 상태 동작만을 표현하기 위해서 ‘stable recursion 함수’ $s_1: X \times A \rightarrow X$ 와 $s_2: Y \times A \rightarrow Y$ 를 도입한다^[10]. 예를 들어 $(x, u) \in X \times A$ 가 M_1 의 과도 상태/입력 조합이고 $f_1(x, u) = x_1$, $f_1(x_1, u) = x_2, \dots$, $f_1(x_{k-1}, u) = x_k$, $f_1(x_k, u) = x_k$ 의 천이 과정을 가진다면

$$s_1(x, u) = x_k \quad (2)$$

로 정의된다. 즉 M_1 은 상태 x 에서 입력 u 를 받아 ‘다음 안정 상태(next stable state)’ x_k 로 즉시 천이한 후 입력이 다시 바뀔 때까지 x_k 에 머무른다. s_2 도 식 (2)의 s_1 과 동일하게 정의된다. s_1 과 s_2 를 길이가 2 이상의 입력 스트링(string)에 대해서 다음과 같이 확장한다.

$$\begin{aligned} s_1(y, u_1 u_2 \cdots u_k) &:= s_1(s_1(x, u_1), u_2 \cdots u_k) \\ s_2(y, u_1 u_2 \cdots u_k) &:= s_2(s_2(y, u_1), u_2 \cdots u_k), \quad u_1 u_2 \cdots u_k \in A^+ \end{aligned} \quad (3)$$

2. 모델 정합

제어 목적은 식 (1)의 M_1 과 M_2 의 폐루프 시스템 동작을 아래와 같이 주어진 기준 모델 R_1 과 R_2 와 일치시키도록 하는 일이다.

$$\begin{aligned} R_1 &= (A, X, x_0, \sigma_1) \\ R_2 &= (A, Y, y_0, \sigma_2) \end{aligned} \quad (4)$$

위 식에서 σ_1 과 σ_2 는 stable recursion 함수이다. R_1 과 R_2 는 M_1 과 M_2 와 각각 동일한 입력 및 상태 집합을 가지나 정상 상태 동작은 서로 다르다. 즉 어떤 상태/입력 조합 $(x,u) \in X \times A$ 와 $(y,v) \in Y \times A$ 에 대해서

$$\begin{aligned} \sigma_1(x,u) &\neq s_1(x,u) \\ \sigma_2(y,v) &\neq s_2(y,v) \end{aligned} \quad (5)$$

이다. M_1 의 예를 든다면 안정 상태 x 에서 입력 u 가 들어오면 머신 M_1 은 다음 안정 상태 $s_1(x,u)$ 로 천이하지만 기준 모델 R_1 은 $s_1(x,u)$ 와 다른 상태인 $\sigma_1(x,u)$ 로 천이하는 모델 부정합(不整合)이 존재한다. M_1 과 R_1 , 그리고 M_2 와 R_2 사이의 모델 부정합을 아래와 같은 집합 D_1 과 D_2 로 다시 표현하자.

$$\begin{aligned} D_1 &= \{(x,u,\acute{x}) \in X \times A \times X \mid s_1(x,u) \neq \acute{x}, \sigma_1(x,u) = \acute{x}\} \\ D_2 &= \{(y,u,\acute{y}) \in Y \times A \times Y \mid s_2(y,u) \neq \acute{y}, \sigma_2(y,u) = \acute{y}\} \end{aligned} \quad (6)$$

또 모델 부정합 D_1 과 D_2 를 일으키는 머신의 상태/입력 조합의 집합 E_1 과 E_2 , D_1 과 D_2 를 일으키는 머신의 상태 집합 F_1 과 F_2 를 아래와 같이 따로 정의한다.

$$\begin{aligned} E_1 &= \{(x,u) \in X \times A \mid \exists \acute{x} \in X \text{ s.t. } (x,u,\acute{x}) \in D_1\} \\ E_2 &= \{(y,u) \in Y \times A \mid \exists \acute{y} \in Y \text{ s.t. } (y,u,\acute{y}) \in D_2\} \\ F_1 &= \{x \in X \mid \exists u \in A \text{ s.t. } (x,u) \in E_1\} \\ F_2 &= \{y \in Y \mid \exists u \in A \text{ s.t. } (y,u) \in E_2\} \end{aligned} \quad (7)$$

그림 1은 병렬 결합된 복합 비동기 순차 머신에 대한 교정 제어 시스템이다. $v \in A$ 는 공통 외부 입력이며 C 는 교정 제어기이다. $u_1, u_2 \in A$ 는 C 가 생성하여 M_1 과 M_2 에 전달하는 제어 입력이고 $x \in X$ 와 $y \in Y$ 는 두 머신으로부터 나오는 상태 피드백이다. 일반적으로 병렬 결합 복합 머신은 각 머신의 출력 두 개를 합하여 최종 출력으로 사용하나 본 논문에서는 x 와 y 가 합쳐진 최종 출력은 고려하지 않는다.

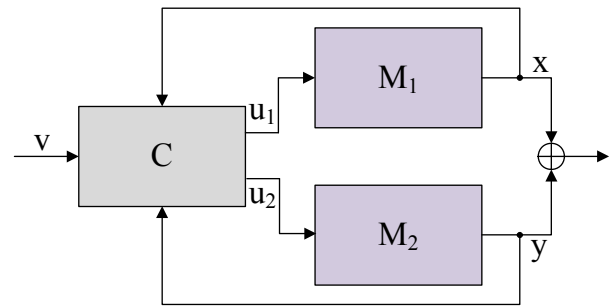


그림 1. 병렬 결합 비동기 순차 머신에 대한 교정 제어 시스템

Fig. 1. Corrective control system for a composite asynchronous machine in parallel connection.

C 는 v 와 x, y 값을 기반으로 제어 입력 u_1 과 u_2 를 생성한다. 현재의 상태/입력 조합이 M_1 과 M_2 에 모델 부정합을 야기하지 않는다면 C 는 외부 입력 v 를 그대로 u_1 과 u_2 의 값으로 한다. 즉 M_1 과 M_2 가 정상(normal) 동작을 한다면

$$u_1 = u_2 = v \quad (8)$$

이다. M_1 과 M_2 중 적어도 하나의 머신에 모델 부정합을 일으키는 외부 입력이 들어온다면 C 는 제어 입력 스트링을 생성하여 M_1 과 M_2 의 폐루프 시스템이 원하는 동작을 가지게 하도록 ‘교정(correction)’한다.

본 연구에서 그림 1의 폐루프 시스템은 기본 모드 원리(the principle of fundamental mode operation)^[10]를 만족시킨다고 가정한다. 기본 모드에서는 비동기 머신이 안정 상태에 있을 때에만 임출력 변수가 바뀌므로 클럭에 의한 동기화에 안 됨으로써 생길 수 있는 예측 불가능한 상태 천이를 방지한다.

III. 제어기 설계

1. 제어기 존재 조건

단일 비동기 순차 머신에 대한 교정 제어 이론을 적용하면 M_1 과 R_1 사이의 모델 부정합 D_1 을 해결하는 교정 제어기가 존재할 필요충분조건은 다음과 같다^[1].

$$\forall (x,u,\acute{x}) \in D_1, \exists t \in A^+ \text{ such that } s_1(x,t) = \acute{x} \quad (9)$$

위 조건을 다시 설명하면 모델 부정합을 일으키는 상태/입력 조합 (x,u) 에서 M_1 을 원하는 상태 \acute{x} 로 천이시키는 입력 스트링 t 가 존재해야 한다는 것이다. 교정 제

여기는 M_1 이 안정 상태 x 에 있을 때 외부 입력이 u 로 바뀌면 u 대신 스트링 t 의 원소를 차례로 M_1 에 전달하여 페루프 시스템이 $s_1(x,u)$ 대신 \dot{x} 으로 비동기적으로 천이하게 만든다. M_2 와 R_2 사이에서 만족되어야 할 조건은 (9)와 유사하게 다음과 같이 표현된다.

$$\forall (y,u,\dot{y}) \in D_2, \exists w \in A^+ \text{ such that } s_2(y,w) = \dot{y} \quad (10)$$

조건 (9)와 (10)이 모두 성립한다면 식 (6)의 모델 부정합 D_1 과 D_2 를 해결하는 교정 제어기가 존재한다. 그런데 이전 연구 결과^[1]를 적용할 때의 문제점은 M_1 과 M_2 에 대한 제어기를 각각 따로 설계해야 한다는 사실이다. 즉 병렬 결합된 복합 비동기 머신에 대해서 독립적으로 동작하는 교정 제어기 두 개가 필요하다. 이번 연구에서는 그림 1에서 도시한 대로 한 개의 제어기만을 이용하여 M_1 과 M_2 에 대한 모델 정합 문제를 동시에 풀 수 있는 교정 제어 기법을 제안한다.

2. 제어기 설계 과정

가. 교정 제어기 구조

그림 1의 교정 제어기 C 는 아래와 같은 입력/출력 (input/output) 형태의 비동기 순차 머신으로 표현된다.

$$C = (X \times Y \times A, A \times A, \Xi, \xi_0, \Phi, \eta) \quad (11)$$

위 식에서 $X \times Y \times A$ 는 입력 집합이며(M_1 과 M_2 의 상태 피드백 x, y 와 외부 입력 v) $A \times A$ 는 제어 입력 집합이다(u_1 과 u_2). 또 Ξ 는 C 의 상태 집합, $\xi_0 \in \Xi$ 는 초기 상태이며 $\phi: \Xi \times X \times Y \times A \rightarrow \Xi$ 와 $\eta: \Xi \times X \times Y \times A \rightarrow A \times A$ 는 C 의 상태 천이 함수와 출력 함수이다.

나. 상태 ξ_0 와 ξ_t 에서의 제어기 동작

교정 제어기를 설계하기 전에 주어진 머신 M_1 과 M_2 와 식 (4)의 모델 R_1 과 R_2 사이에서 조건 (9)와 (10)이 모두 만족된다고 가정한다. 최초 구동 시 교정 제어기 C 는 초기 상태 ξ_0 에 머무르며 외부 입력 v 를 그대로 제어 입력으로 전달한다. 따라서

$$\eta(\xi_0, x, y, v) = (v,v) \quad \forall (x,y) \in X \times Y \quad (12)$$

이다. M_1 과 M_2 가 모델 부정합을 일으키는 상태 천이를 가지려면 두 머신 중 적어도 한 개가 식 (7)의 F_1 이나

F_2 에 속하는 상태와 안정 조합을 이루어야 한다. M_1 이나 M_2 가 이러한 안정 상태에 진입할 때 C 는 다음과 같이 ξ_0 에서 'transition 상태' ξ_t 로 이동한다.

$$\Phi(\xi_0, x, y, v) = \xi_t \quad (13)$$

$$[v \in S(x) \text{ and } v \in S(y)] \text{ and } [x \in F_1 \text{ or } y \in F_2]$$

위 식에서 $S(x) \subset A$ 와 $S(y) \subset A$ 는 상태 x, y 와 안정 조합을 이루는 입력 집합을 각각 가리킨다. 위 식의 뜻은 M_1 과 M_2 가 안정 상태에 진입하고 그 안정 상태가 다음 입력에 따라서 모델 부정합을 야기할 수 있다면 C 를 ξ_t 로 옮긴 후 모델 부정합에 대한 제어를 준비해야 한다는 것이다.

ξ_t 는 일종의 stand-by 상태로서 다음 외부 입력의 변화에 따라 C 의 행동이 결정된다. 먼저 외부 입력이 현재 상태와 안정 조합을 이루지 않고 동시에 모델 부정합을 일으키지 않는 값으로 바뀌었다면 C 는 ξ_0 로 복귀한다. 이 동작을 구현하기 위해서 ϕ 를 아래와 같이 설정한다.

$$\phi(\xi_t, x, y, v) = \xi_0 [v \notin S(x) \text{ and } v \notin S(y)] \text{ and } [(x,v) \notin E_1 \text{ and } (y,v) \notin E_2] \quad (14)$$

C 는 ξ_t 에서도 구체적인 제어 동작을 아직 실행하지 않으므로 이때의 제어 입력은 여전히 외부 입력과 동일하다. 따라서

$$\eta(\xi_t, x, y, v) = (v,v) \quad \forall (x,y) \in X \times Y \quad (15)$$

이다.

앞의 경우와 대조적으로 바뀐 외부 입력이 현재 상태와 결합하여 모델 부정합을 일으킨다고 하자. 이때 M_1 과 M_2 동시에 모델 부정합이 발생하거나 M_1 과 M_2 중 하나의 머신에만 모델 부정합이 생길 수 있다.

다. M_1 에만 모델 부정합이 발생한 경우

먼저 M_1 과 M_2 중 하나의 머신에만 모델 부정합이 생기는 경우를 논한다. 편의상 M_1 에 모델 부정합이 발생한다고 하자. 이때 C 는 외부 입력 대신 제어 입력 스트링을 M_1 에 전달하여 모델 정합을 구현한다. 조건 (9)에 의해서 $s_1(x,t) = \dot{x}$ 인 제어 입력 스트링 $t = u_1 \dots u_k$ 가 존재한다. $|t| = k$ 이므로 t 를 이용한 교정 동작을 위해 k 개의 보조 상태(auxiliary state)가 필요하다^[1]. 이 보조 상태를 $\xi_1, \dots, \xi_k \in \Xi$ 라고 정의하자. C 가 ξ_t 에 있을 때, 즉 M_1 과

M_2 가 안정 상태 x 와 y 에 있을 때 외부 입력이 $(x, v) \in E_1$ 이고 $(y, v) \notin E_2$ 인 v 로 바뀐다고 하자. M_2 는 v 를 받고 정상 동작을 계속하기 때문에 C 는 M_2 의 제어 입력 u_2 를 v 로 설정한다(M_1 을 위한 교정 제어 중에도 u_2 값은 v 로 일정하게 유지된다).

M_1 을 위한 교정 제어는 다음과 같이 이루어진다. v 를 입력 받은 C 는 첫번째 보조 상태 ξ_1 로 천이한다.

$$\phi(\xi_1, x, y, v) = \xi_1 \quad (x, v) \in E_1 \text{ and } (y, v) \notin E_2 \quad (16)$$

그런 다음 t 의 첫번째 입력 u_1 을 생성하여 M_1 에 전달한다. u_1 을 받은 M_1 은 x 에서 $x_1 := s_1(x, u_1)$ 로 상태 천이한다. 또 상태 피드백이 x_1 로 바뀌는 것을 확인한 C 는(그림 1 참조) 두번째 보조 상태 ξ_2 로 천이한다. 이전 단계와 마찬가지로 C 는 t 의 두번째 입력 u_2 를 제어 입력으로 M_1 에 보내며 M_1 은 $x_2 := s_1(x_1, u_2)$ 로 상태 천이하고, 상태 피드백 x_2 를 받은 C 는 다음 단계 동작을 실행한다. 이러한 일련의 상호 동작을 구현하기 위해서 ϕ 와 η 를 아래와 같이 설정한다(v, y, u_2 값은 일정하다).

$$\phi(\xi_i, x_i, y, v) = \xi_{i+1}, \quad i=1, \dots, k$$

$$\eta(\xi_i, x_i, y, v) = (u_i, v), \quad x_i = s_1(x_{i-1}, u_i), \quad x_0 = x \quad (17)$$

결국 마지막 보조 상태 ξ_k 에서 C 가 제어 입력 u_k 를 생성하면 M_1 은 목적 상태 $\dot{x} = s_1(x_{k-1}, u_k)$ 에 도달한다. 상태 피드백 \dot{x} 를 받은 C 가 초기 상태 ξ_0 으로 복귀하면서 교정 동작을 완료한다. 비동기적으로 실행되는 (16)과 (17)의 과정에서 외부 사용자는 M_1 이 상태 x 에서 외부 입력 v 를 받은 후 목적 상태 \dot{x} 로 즉시 천이하는 것만을 관측하므로 모델 정합이 실현되었다고 말할 수 있다.

정리하면 v 를 받은 C 는 t 를 이용하여 M_1 을 위한 교정 제어를 실행하고 M_2 에게는 v 를 그대로 전달한다. 또 이 동작들은 외부 입력이 다시 바뀌기 전에 모두 이루어지므로 복합 비동기 머신의 모델 정합 특성을 유지시킨다.

라. M_1, M_2 동시에 모델 부정합이 발생한 경우

다음으로 바뀐 외부 입력 v 가 M_1 과 M_2 모두에게 모델 부정합을 야기하는 경우를 생각하자. 즉 v 는 현재 상태 x 와 y 에 대해서 $(x, v) \in E_1$ 이며 $(y, v) \in E_2$ 이다. 앞서와 마찬가지로 조건 (9)에 의해서 $s_1(x, t) = \dot{x}$ 인 제어 입력 스트링 $t = u_1 \dots u_k$ 가 존재하고 또 조건 (10)에 의해

$s_2(y, w) = \dot{y}$ 인 제어 입력 스트링 $w = v_1 \dots v_m$ 이 존재한다. $|t|=k, |w|=m$ 이므로 앞에서 기술한 설계 과정을 그대로 적용한다면 M_1 의 제어를 위해서 C 에게 k 개의 보조 상태가 필요하며, M_2 의 제어를 위해서 m 개의 보조 상태가 필요하다. 그런데 C 가 M_1 과 M_2 의 교정 제어 궤적(trajjectory)을 공유하면 필요한 보조 상태의 수를 줄일 수 있다.

편의상 $k \geq m$ 라 하자. 입력 스트링의 길이가 긴 M_1 의 교정 동작을 먼저 구현한다. 앞의 경우와 동일하게 C 는 k 개의 보조 상태 $\xi_1, \dots, \xi_k \in E$ 를 정의한다. 표기를 명확하게 하기 위하여 v 로 바뀌기 전의 외부 입력을 v_{old} 라고 하자. 즉 M_1 과 M_2 는 안정 조합 (x, v_{old}) 와 (y, v_{old}) 에 머물러 있었다.

한 개의 교정 제어기 C 를 이용하여 기본 모드 원리를 만족시키면서 M_1 과 M_2 의 안정 상태를 모두 바꾸려면 제어 입력 u_1 과 u_2 를 동시에 변경해서는 안 된다. u_1 과 u_2 가 동시에 변한다면 M_1 과 M_2 가 모두 과도 상태로 진입하게 된다. 다시 말하면 두 머신이 비록 병렬 결합되었지만 교정 제어는 순서대로(sequentially) 실행해야 한다. 그런데 하나의 교정 동작을 수행하는 시간이 극히 짧으므로 순서대로 각 교정 동작을 완료할 때까지 외부 입력은 변하지 않고 일정하게 유지될 것이다.

C 가 M_1 의 교정 제어를 먼저 수행한다고 하자. (16)과 마찬가지로 ξ_1 에 있던 C 는 외부 입력 v 를 받고 첫번째 보조 상태 ξ_1 로 천이한다. 그런 다음 C 는 (17)에서 명기된 바대로 ξ_1 에서 ξ_k 까지 일련의 교정 동작을 연쇄적으로 수행한다. 이때 M_2 는 안정 상태 y 를 유지해야 하므로 제어 입력으로 $u_2 = v_{old}$ 를 받아야 한다. 이상과 같은 요구 조건을 실현하는 ϕ 와 η 의 정의는 아래와 같다(x_i 는 식 (17)에서와 마찬가지로 M_1 이 거치는 안정 상태를 의미한다).

$$\begin{aligned} \phi(\xi_i, x_i, y, v) &= \xi_{i+1} \\ \eta(\xi_i, x, y, v) &= (u_i, v_{old}) \quad \forall (x, y) \in X \times Y \end{aligned} \quad (18)$$

M_1 의 교정 제어를 완료한 후 C 는 M_2 를 위한 교정 동작을 바로 시작해야 한다. 그런데 $k \geq m$ 이므로 w 를 이용한 M_2 에서의 교정 궤적은 이미 정의한 k 개의 보조 상태 ξ_1, \dots, ξ_k 를 이용하여 구현 가능하다. 마지막 보조 상태 ξ_k 에서 C 는 M_1 에 t 의 마지막 입력 character u_k 를 전달하고, M_1 은 안정 조합 (\dot{x}, u_k) 를 가진다고 하였다. 앞에서는 C 가 상태 피드백 \dot{x} 를 받은 후 초기 상태 ξ_0 으로

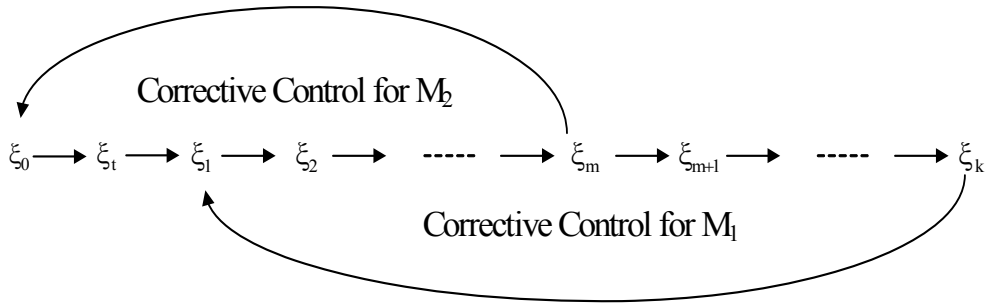


그림 2. 교정 궤적을 공유하는 제어 과정
Fig. 2. Control procedure sharing the correction trajectory.

복귀한다고 했으나 이번에는 ξ_0 대신 다시 첫번째 보조 상태 ξ_1 로 이동하도록 설정한다. 그런 다음 w 의 입력 character v_1, v_2, \dots, v_m 을 차례대로 생성하여 제어 입력 u_2 로 전달한다. 중요한 사실은 이때 M_1 의 제어 입력으로 u_k 를 계속 넣어주어야 한다는 것이다. 즉 M_1 을 목적 상태 x 에 머물러 있게 하면서 M_2 의 교정 제어를 수행한다. 이상을 고려하여 ϕ 와 η 를 설정하면 아래와 같다.

$$\Phi(\xi_k, x_k, y, v) = \xi_1$$

(M_1 에 대한 교정 제어 완료 이후 ξ_k 에서 ξ_1 로 천이)

$$\Phi(\xi_j, x_k, y_j, v) = \xi_{j+1}, \quad j=1, \dots, m,$$

$$\eta(\xi_j, x_k, y_j, v) = (u_k, v_j), \quad y_j = S_2(y_{j-1}, v_j), \quad y_0 = y$$

위 식에서 y_j 는 ($j=1, \dots, m$) M_2 가 입력 스트링 w 를 받고 y 에서 y 까지 천이할 때 거치는 안정 상태들을 가리킨다. 보조 상태 ξ_m 에서 M_2 에 대한 교정 동작을 완료한 C 는 초기 상태 ξ_0 으로 복귀한다. 이상과 같은 C 의 교정 동작을 도시하면 그림 2와 같다.

마. 선행 연구와의 비교

(12)~(19)가 병렬 결합된 두 비동기 머신 M_1 과 M_2 를 위한 모델 정합을 구현하는 단일 교정 제어기 C 에 대한 설계 과정이다. 교정 제어기 한 개만을 설계하여 모델 정합을 이룩하기 때문에 제안된 방법이 교정 제어기 두 개를 사용해야 하는 기존 방법보다 더 효과적임은 자명하다. 제안된 방법의 효율성은 제어기의 보조 상태 개수를 비교함으로써 더 드러난다. M_1 과 M_2 동시에 모델 부정합을 일으키는 외부 입력 v 를 다시 생각해 보자. 기존 방법대로 M_1 과 M_2 각각에 대해서 교정 제어기를 따로 설계한다면 필요한 보조 상태 수는 $k+m$ 이다(k 와 m 은 사용하는 제어 입력 스트링의 길이). 그

러나 교정 궤적을 공유하는 본 논문의 기법을 적용할 때의 필요한 보조 상태 개수는 $\max(k,m)$ 이므로 제어기의 크기(=상태 수)를 확실히 줄일 수 있다.

IV. 예 제

1. 사례 연구 비동기 머신

제안한 방법의 적용과정을 예시하기 위하여 그림 3

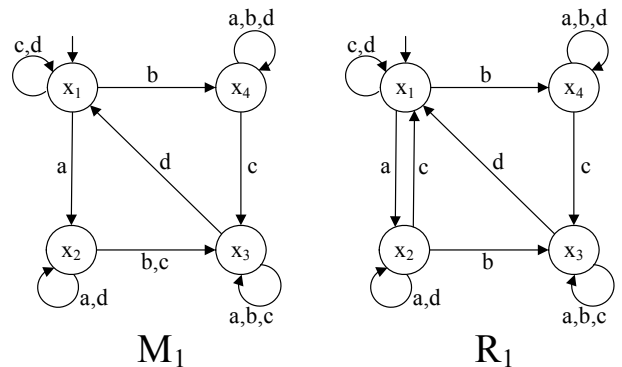


그림 3. 비동기 머신 M_1 과 기준 모델 R_1
Fig. 3. Asynchronous machine M_1 and reference model R_1 .

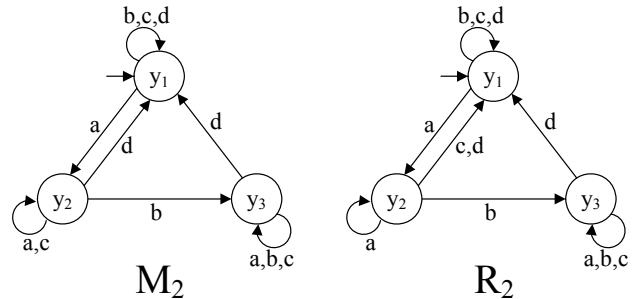


그림 4. 비동기 머신 M_2 와 기준 모델 R_2
Fig. 4. Asynchronous machine M_2 and reference model R_2 .

과 그림 4에 도시한 비동기 순차 머신 $M_1=(A,X,x_0,f_1)$ 과 $M_2=(A,Y,y_0,f_2)$, 그리고 기준 모델 $R_1=(A,X,x_0,o_1)$ 과 $R_2=(A,Y,y_0,o_2)$ 사이의 모델 정합 문제를 생각하자. 먼저 입력과 상태 집합은 다음과 같다.

$$A = \{a, b, c, d\}$$

$$X = \{x_1, x_2, x_3, x_4\}, x_0 := x_1$$

$$Y = \{y_1, y_2, y_3\}, y_0 := y_1$$

편의상 $f_1(x,u)=s_1(x,u) \quad \forall (x,u) \in X \times A, f_2(y,u)=s_2(y,u) \quad \forall (y,u) \in Y \times A$ 라고 설정하였다. 즉 M_1 과 M_2 는 정상 상태(steady state) 동작만 가진다. 따라서 본 모델 정합 문제는 정상 상태 동작이 M_1 과 M_2 와 일치하는 모든 비동기 머신에게 동일하게 적용된다.

먼저 그림 3의 M_1 과 R_1 을 비교해보면 상태/입력 조합 (x_2,c) 에서의 천이 특성이 서로 다름을 알 수 있다. M_1 은 안정 상태 x_2 에서 입력 c 를 받으면 다음 안정 상태 x_3 으로 이동하나 R_1 은 x_3 대신 x_1 로 천이한다. 다시 말하면 $s_1(x_2,c)=x_3$ 이나 $o_1(x_2,c)=x_1$ 이다. 이를 모델 부정합 집합으로 표현하면(식 (6) 참조)

$$D_1=\{(x_2,c,x_1)\}, E_1=\{(x_2,c)\}, F_1=\{x_2\}$$

이다. 다음으로 그림 4의 M_2 와 R_2 를 비교하면 (y_2,c) 에서 상태 천이 차이를 발견할 수 있다. M_2 이 안정 상태 y_2 에 있을 때 외부 입력이 c 로 바뀌면 상태 천이를 하지 않고 현재 안정 상태를 그대로 유지하지만($s_2(y_2,c)=y_2$), 기준 모델 R_2 는 다음 안정 상태 y_1 로 천이한다($o_2(y_2,c)=y_1$). 앞서와 마찬가지로 이 차이를 모델 부정합 집합으로 나타내면

$$D_2=\{(y_2,c,y_1)\}, E_2=\{(y_2,c)\}, F_2=\{y_2\}$$

이다.

M_1 과 M_2 가 병렬 결합되었다고 하고 모델 정합을 위한 교정 제어기의 존재조건 및 제어기 설계를 기술한다. 먼저 그림 3을 보면 M_1 을 x_2 에서 목적 상태 x_1 로 상태 천이시키는 입력 스트링 $t:=bd$ 가 존재한다($s_1(x_2,bd)=x_1$). 또 그림 4를 보면 M_2 를 y_2 에서 목적 상태 y_1 로 이동시키는 입력 스트링 $w:=d$ 가 존재한다($s_2(y_2,d)=y_1$). 조건 (9)와 (10)이 동시에 만족되므로 두 모델 정합 문제를 해결하는 그림 1의 상태 피드백 교정 제어기 C 가 존재한다.

2. 제어기 설계 결과

그림 5는 주어진 모델 정합 문제를 해결하는 교정 제어기 C 를 설계한 것이며 표 1은 C 의 출력 함수이다. $|t|=2$ 이며 $|w|=1$ 이므로 C 는 그림 5에 나와 있듯이 두 개의 보조 상태 ξ_1 과 ξ_2 를 필요로 한다. C 의 동작 과정을 설명하기 위해서 그림 3과 그림 4를 다시 보자. M_1 과 M_2 가 모델 부정합을 야기할 수 있는 상태 x_2, y_2 와 각각 안정 조합을 이루고 있다고 한다면 이때의 외부 입력은 a 이다. 따라서 C 는 입력 (x_2,y_2,a) 를 받으면 초기 상태 ξ_0 에서 transition 상태 ξ_t 로 이동한다. 다음 외부 입력의 변화를 보고 C 는 차기 동작을 결정한다. 만약 모델 부정합을 일으키지 않는 외부 입력 v 가 들어온다면(즉 v 가 b, d 중 하나로 바뀌었다면) C 는 ξ_0 으로 복귀한다. 반면 외부 입력 v 가 c 로 바뀌었다면 C 는 모델 정합을 위한 교정 제어 동작을 수행한다.

논문에서 제안된 기법대로 C 는 제어 입력 스트링의 길이가 더 긴 M_1 을 위한 교정 동작을 먼저 실행한다. 입력 (x_2,y_2,c) 에 반응하여 C 는 ξ_t 에서 첫번째 보조 상태 ξ_1 로 천이한 후 t 의 첫번째 입력 character b 를 생성하여 M_1 에 전달한다($u_1=b$). 이때 M_2 는 현재 상태 y_2 를 그

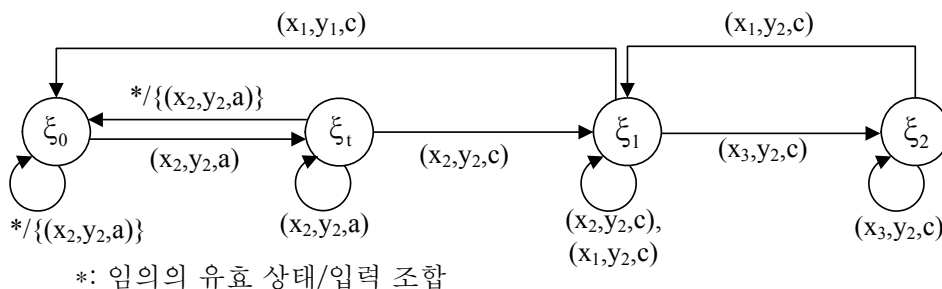


그림 5. 교정 제어기 C
Fig. 5. Corrective controller C.

표 1. 교정 제어기 C의 출력 함수
Table 1. Output function of the corrective controller C.

	$n(\xi, x, y, v)$
ξ_0	(v, v)
ξ_t	(v, v)
ξ_1	(b, a) for (x_2, y_2, c)
	(d, d) for (x_1, y_2, c)
ξ_2	(d, a)

대로 유지해야 하므로 제어 입력 $u_2=a$ 를 받는다(표 1 참조). b 를 받은 M_1 은 x_2 에서 다음 안정 상태 x_3 으로 천이한다. 상태 피드백 변화를 통해 이 정보를 획득한 C는 두번째 보조 상태 ξ_2 로 천이한 다음 t 의 두번째 입력 d 를 생성한다. d 를 받은 M_1 은 결국 원하는 상태 x_1 에 도달하며 첫번째 모델 정합이 완성된다.

M_1 을 위한 교정 제어가 끝났다는 사실은 M_1 의 상태 피드백이 x_1 로 바뀌므로써 드러난다. M_2 에 대한 교정 제어를 하기 위해서 C는 ξ_2 에서 입력 (x_1, y_2, c) 를 받은 후 다시 보조 상태 ξ_1 로 이동한다. 여기서 M_2 를 위한 입력 스트링 $w=d$ 를 제어 입력 u_2 로 전달하면 M_2 는 목적 상태 y_1 으로 천이하며 최종 모델 정합이 완료된다. 모델 정합이 끝난 M_1 을 마지막 상태 x_1 에 머무르게 하기 위해서 두번째 제어 단계에서 u_1 을 t 의 마지막 입력 character d 로 교정시켜야 한다(표 1 참조).

이상과 같이 설계한 C의 교정 동작은 기본 모드 원리를 만족시키며 주어진 모델 정합 문제를 해결한다. 또한 앞에서 기술한 대로 C의 설계 과정에서 교정 동작을 구현하는 데 필요한 보조 상태의 개수를 최소한으로 줄였음을 알 수 있다.

V. 결 론

이번 논문에서는 두 입력/상태 비동기 머신이 병렬 결합된 복합 비동기 순차 머신에 대한 모델 정합 문제를 다루었다. 본 연구의 핵심은 한 개의 교정 제어기만으로 각 머신에 대한 모델 정합 문제를 해결하는 제어 기법을 제안하였다는 사실이다. 교정 제어기가 존재할 필요충분조건은 제어기를 두 개 사용해야 하는 경우와 동일하나 제어기의 크기, 즉 제어기 상태 개수를 최소한으로 줄일 수 있었다. 사례 연구를 통해서 제안된 제어기의 설계 과정을 예시하였다. 본 연구는 병렬 결합된 두 개의 비동기 머신에 대한 모델 정합 문제를 다루

었으나 비동기 머신 n 개($n \geq 2$)가 결합된 일반적인 복합 비동기 머신에 대해서도 적용 가능하다.

REFERENCES

- [1] 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.
- [2] J.-M. Yang, "Corrective control of asynchronous sequential machines for tolerating permanent faults", Journal of The Institute of Electronics Engineers of Korea, vol. 47, no. 5, pp. 9-17, 2010.
- [3] J.-M. Yang and S. W. Kwak, "Fault diagnosis and fault-tolerant control of input/output asynchronous sequential machines," IET Control Theory and Applications, vol. 6, no. 11, pp. 1682-1689, 2012.
- [4] J. Peng and J. Hammer, "Bursts and output feedback control of non-deterministic asynchronous sequential machines," European Journal of Control, vol. 18, no. 3, pp. 286-300, 2012.
- [5] S.-J. Park, "Power-based supervisory control of discrete event systems: political economy analysis", Journal of The Institute of Electronics Engineers of Korea, vol. 50, no. 7 pp. 244-252, 2013.
- [6] E. A. Lee and P. Varaiya, Structure and Interpretation of Signals and Systems, 2nd ed., LeeVaraiya.org, 2011.
- [7] J.-M. Yang, "Model matching for composite asynchronous sequential machines in cascade connection," Journal of The Institute of Electronics Engineers of Korea, vol. 50, no. 5, pp. 253-261, 2013.
- [8] K. Rudie and W. M. Wonham, "Think globally, act locally: decentralized supervisory control," IEEE Transactions on Automatic Control, vol. 37, no. 11, pp. 1692-1708, 1992.
- [9] H. Li and Y. Wang, "Consistent stabilizability of switched boolean networks," Neural Networks, vol. 46, pp. 183-189, 2013.
- [10] Z. Kohavi and N. K. Jha, Switching and Finite Automata Theory, 3rd ed., Cambridge University Press: Cambridge, UK, 2010.

저 자 소 개



양 정 민(정회원)

1993년 한국과학기술원 전기 및
전자공학과 학사 졸업

1995년 한국과학기술원 전기 및
전자공학과 석사 졸업

1999년 한국과학기술원 전기 및
전자공학과 박사 졸업

1999년~2001년 한국전자통신연구원 선임연구원

2001년~2013년 대구가톨릭대학교 전자공학과
교수

2013년~현재 경북대학교 전자공학부 부교수

<주관심분야 : 비동기 머신 제어, 보행 로봇 시스
템 등>