

논문 2010-47SC-4-2

모델 불확실성을 가진 비동기 순차 머신의 모델 정합 포함을 위한 상태 피드백 제어

(State Feedback Control for Model Matching Inclusion of
Asynchronous Sequential Machines with Model Uncertainty)

양 정 민*, 박 용 국**

(Jung-Min Yang and Yong Kuk Park)

요 약

유한 상태 머신으로 표현되는 비동기 머신의 안정 상태 동작은 피드백 제어를 통해서 원하는 목적에 맞게 교정될 수 있다. 본 논문에서는 불확실한 상태 천이를 가지는 입력/상태 비동기 머신을 위한 상태 피드백 제어기를 제안한다. 비동기 머신은 결정적인 동작 특성을 가지고 있으나 모델 불확실성, 내부 고장 등으로 인해서 일부 영역의 상태 천이 함수가 불확실하다. 교정 제어의 목적은 불확실한 상태 천이를 고려하면서 머신의 페루프 동작이 주어진 정상적인 모델 동작의 부분 집합이 되도록 하는 일이다. 또 제어기는 실제 교정 동작을 수행하면서 획득하는 머신의 정확한 상태 천이를 그 다음 제어 동작에 반영한다. 즉 학습을 통해서 페루프 시스템이 이룰 수 있는 모델의 부분 동작의 범위가 더 확대된다. 사례 연구를 제안된 제어기의 설계 과정을 예시한다.

Abstract

Stable-state behaviors of asynchronous sequential machines represented as finite state machines can be corrected by feedback control schemes. In this paper, we propose a state feedback control scheme for input/state asynchronous machines with uncertain transitions. The considered asynchronous machine is deterministic, but its state transition function is partially known due to model uncertainty or inner logic errors. The control objective is to compensate the behavior of the closed-loop system so that it matches a sub-behavior of a prescribed model despite uncertain transitions. Furthermore, during the execution of corrective action, the controller reflects the exact knowledge of transitions into the next step, i.e., the range of the behavior of the closed-loop system can be enlarged through learning. The design procedure for the proposed controller is described in a case study.

Keywords : Asynchronous Sequential Machines, Model Matching Inclusion, Uncertain Transitions,
State Feedback Control

I. 서 론

디지털 순차 회로나 컴퓨터 프로그래밍 등 유한 상태 머신(finite state machine)으로 표현되는 이산 사건 시스템(discrete-event system)의 동작은 입력과 출력 조합으로 나타낼 수 있다. 시스템의 내부 로직(logic)이 한번 설계되면 이러한 입력-출력 조합 명세(specification)는 통상 로직이 재설계되지 않는 한 변하지 않는다. 하

* 정희원, 대구가톨릭대학교 전자공학과
(Department of Electrical Engineering,
Catholic University of Daegu)

** 정희원, 대구가톨릭대학교 기계자동차공학부
(School of Mechanical and Automotive
Engineering, Catholic University of Daegu)

※ 이 논문은 2010년도 대구가톨릭대학교 교내연구비
지원에 의한 것임

접수일자: 2009년8월12일, 수정완료일: 2010년7월8일

지만 전역 클럭(global clock)이 존재하지 않는 비동기 순차 머신(asynchronous sequential machine)은 피드백 제어를 통해서 입력-출력의 특성을 바꿀 수 있다. 외부 입력이 들어오면 제어기가 이 입력 값 대신 원하는 동작을 이끌어내는 제어 입력을 비동기 머신에 넣어줌으로써 페루프 시스템의 동작이 보상된다. 비동기 머신의 과도 응답 시간이 극히 짧기 때문에 외부 사용자에게는 페루프 시스템이 원하는 입력-출력 조합 특성을 보이면서 동작하는 모습만 관측된다.

병렬 연산의 효율을 높이기 위한 비동기식 설계, 비동기 라우팅 칩(routing chip) 제작, 캐시 컨트롤러(cache controller) 등 비동기 방식을 활용한 전용 하드웨어/소프트웨어 설계와 산업계 응용에 관한 연구 결과가 여전히 많이 발표되고 있기 때문에^[1~2] 비동기 머신을 위한 이러한 자동 제어(automatic control) 이론은 연구할만한 가치가 있다고 말할 수 있다^[3~5].

이번 연구의 목적은 불확실한 상태 천이를 가진 입력/상태 비동기 머신을 위한 교정 제어기를 제안하는 일이다. 고려하는 비동기 머신은 결정적인(deterministic) 동작 특성을 가지고 있으나 모델 불확실성, 내부 고장 등으로 인해서 일부 영역의 상태 천이 함수가 불확실하다. 즉 불확실한 상태-입력 조합에서 제어기는 실제 제어 동작을 수행하여 피드백 정보를 관측한 다음에야 정확한 천이를 알 수 있다. 우주 입자나 급격한 전압 변화로 시스템의 메모리 비트 하나가 특정 값으로 고정되어 버리는 Single Event Latchup (SEL)^[6] 고장이 이러한 불확실한 상태 천이를 유발하는 대표적인 사건이다.

본 논문의 목적은 두 가지로 나뉜다. 첫 번째는 머신이 가지는 모델 불확실성에도 불구하고 페루프 시스템이 원하는 모델의 부분 동작(sub-behavior)을 가지도록 하는 교정 제어기를 구현한다. 즉 모델 정합 포함(model matching inclusion) 문제^[7]를 해결한다. 이전 연구^[3~6]와 비교하여 본 연구가 가지는 가장 큰 차이점은 페루프 시스템의 동작을 모델과 정확하게 일치시키는 것이 아니라 모델의 부분 동작과 일치시킨다는 점이다. 두 번째 목적은 실제 교정 동작을 수행하면서 제어기가 획득하는 머신의 정확한 상태 천이를 그 다음 제어 동작에 반영하는 일이다. 즉 학습을 통해서 페루프 시스템이 이룰 수 있는 모델의 부분 동작의 범위를 더 확장시킨다.

이번 논문의 두 가지 목적은 모두 고려하는 비동기 머신이 불확실성을 가지며, 또한 제어기가 피드백 정보

를 받은 후에는 이 불확실성이 제거된다는 특징에 착안하여 설정하였다. 본 연구는 크리티컬 레이스(critical race)^[3]나 비결정적(non-deterministic) 머신^[8] 등 시스템 실행 여부에 관계없이 항상 비결정적 특성을 보이는 머신에 대한 연구와는 다르다. 특히 비결정적 머신에 대해서는 본 연구의 두 번째 목적인 학습을 통한 페루프 시스템 동작 확장 기법이 적용되지 못한다.

본 연구에서는 [8]에서 제안된 방법 바탕으로 불확실한 상태 천이를 가지는 머신의 도달가능성을 표현한다. 또한 사례 연구를 통해서 제안된 제어기의 존재 조건 확인 과정과 설계 과정을 예시한다.

II. 모델 불확실성을 가진 비동기 머신

입력/상태 비동기 순차 머신은 현재의 상태 값이 출력으로 나오는 머신이다. 유한 상태 머신으로 입력/상태 비동기 머신을 표현하면 다음과 같다.

$$\Sigma = (A, X, x_0, f)$$

A는 입력 집합, X는 상태 집합, x_0 는 초기 상태이며 $f: X \times A \rightarrow X$ 는 상태 천이 함수이다. 클럭이 없는 비동기 머신의 특성에 따라 Σ 의 과도 상태 천이 시간은 극히 짧다. 예를 들어 $(x, u) \in X \times A$ 가 안정 상태 조합이라고 하면 $f(x, u) = x$ 이다. 이때 외부 입력이 u' 로 바뀐다고 하고 (x, u') 가 과도 상태 조합이라고 한다면 $f(x, u') = x_1, x_2 = f(x_1, u'), \dots$ 등으로 머신 Σ 가 과도 상태 x_1, x_2, \dots 를 순식간에(이론적으로 0 시간) 거쳐서 다음 안정 상태(next stable state) x' 에 도달한다($f(x', u) = x'$). 외부 사용자에게는 머신이 안정 조합 (x, u) 에서 다음 안정 조합 (x, u') 으로 즉시 이동하는 모습만 관측된다.

Σ 의 안정 상태 동작만을 따로 표현하기 위해서 'stable-state 머신 Σ_s '를 아래와 같이 정의한다^[3~4].

$$\Sigma_s = (A, X, x_0, \mathcal{S}), \quad \mathcal{S}(x, u) := x' \quad (1)$$

위 식에서 f 대신 사용되는 'stable recursion 함수' s는 상태-입력 조합 (x, u) 의 다음 안정 상태 x' 를 반환하는 함수이다. 단위 입력 대신 입력 스트링(string)을 s의 변수로 설정하면 다음과 같이 일반화할 수 있다.

$$s(x, ut) := s(s(x, u), t), \quad x \in X, u \in A, t \in A^+$$

위 식에서 A^+ 는 A에 속한 단위 입력들로 이루어지는 길이 1 이상의 스트링 집합을 말한다. 또 $s(x, t) = x'$ 인

입력 스트링 $t \in A^+$ 가 존재하면 상태 x' 는 상태 x 로부터 '도달가능하다고(stably reachable)'^[3~4] 말한다.

본 논문에서는 불확실한 상태 천이를 가지는 r 개의 상태-입력 조합이 Σ 에 존재한다고 가정한다. 그러한 상태-입력 조합을 $(z_i, v_i) \in X \times A, i=1, \dots, r$ 이라 명명하면 이 조합에서 Σ 의 stable recursion 함수 s 는 아래와 같다.

$$s(z_i, v_i) \in \{z_{i1}, \dots, z_{ip(i)}\}, p(i) \geq 2 \quad (2)$$

즉 교정 제어를 시작하기 전 제어기는 (z_i, v_i) 의 다음 안정 상태가 무엇인지 알지 못하며 $z_{i1}, \dots, z_{ip(i)}$ 중의 하나라는 사실만 알고 있다. 하지만 실제 제어를 수행하여 상태 피드백을 받은 후에는 $s(z_i, v_i)$ 의 정확한 값을 알게 된다. 이것이 항상 비결정적 동작을 보이는 머신을 제어할 때와는^[3, 8] 다른 점이다.

본 논문에서는 [8]에서 제안된 방법을 이용하여 불확실한 상태 천이를 다시 표현한다. (z_i, v_i) 에 속한 각 입력 v_i 에 첨자를 추가한 $p(i)$ 개의 입력 값 $v_{i1}, \dots, v_{ip(i)}$ 를 새로 정의하자. v_{ij} 는 z_{ij} 와 대응하며 stable recursion 함수 s 는 아래와 같이 정의된다.

$$s(z_i, v_{ij}) := z_{ij}, j=1, \dots, p(i) \quad (3)$$

물론 위 $p(i)$ 개의 상태 천이 중 한 개만이 실제로 정의된(하지만 아직 알지 못하는) 관계이다. v_{ij} 를 '불확실한 입력(uncertain input character)'이라고 부르기로 하고 모든 불확실한 입력의 집합을 $A_u := \{v_{ij} | 1 \leq i \leq r, 1 \leq j \leq p(i)\}$ 라고 정의한다. 앞으로 집합 A_u 를 추가하여 stable-state 머신 Σ_{is} 을 $\Sigma_{is} = (A \cup A_u, X, x_0, s)$ 라고 표현한다. s 는 정상적인 상태-입력 조합에서는 (1), 불확실한 상태 천이에서는 (3)과 같이 정의된다.

비동기 머신의 교정 제어기를 설계하기 위해서는 먼저 머신 Σ 의 상태간 도달가능성을 알아야 한다. 그런데 Σ 는 불확실한 상태 천이를 가지고 있으므로 이전 연구와는 다른 도달가능성 표현법이 필요하다. 먼저 아래와 같은 행렬 $R(\Sigma)$ 를 정의한다.

정의 1. 불확실한 입력 집합 A_u 를 가지는 입력/상태 비동기 순차 머신을 $\Sigma = (A \cup A_u, X, x_0, f)$ 라고 하고 상태 집합을 $X = \{x_1, \dots, x_n\}$ 라 하자. $n \times n$ 행렬 $R(\Sigma)$ 의 (p, q) 번째 원소는 아래와 같이 정의된다.

$$R_{pq}(\Sigma) = \{t \in (A \cup A_u)^+ | s(x_p, t) = x_q, |t| \leq n-1\} \quad (4)$$

$R_{pq}(\Sigma)$ 는 머신 Σ 를 상태 x_p 에서 x_q 로 천이시키고 길

이가 $n-1$ 보다 적거나 같은 모든 입력 스트링을 포함한다. x_p 와 x_q 사이에서 그러한 입력 스트링이 하나도 존재하지 않으면 $R_{pq}(\Sigma) = \emptyset$ 이다. n 개의 상태를 가진 비동기 머신에서는 길이 $n-1$ 이하의 입력 스트링만 체크하면 모든 도달가능성을 확인할 수 있다^[3].

$t \in A^+$ 인 t 가 존재하는 경우는 정의에 의해서 x_p 에서 x_q 까지의 도달가능성이 확실하게 보장된다. 문제는 $R_{pq}(\Sigma)$ 에 $t \in A^+$ 인 t 가 하나도 없으며 $R_{pq}(\Sigma)$ 에 속한 모든 스트링이 불확실한 입력 집합 A_u 의 원소를 적어도 하나 이상 포함하고 있는 경우이다. 교정 제어를 실행하기 전에는 제어기가 임의의 불확실한 상태-입력 조합 (z_i, v_i) 의 정확한 다음 안정 상태를 전혀 알지 못한다. 그러나 만약 v_i 가 유도하는 모든 불확실한 입력 $v_{i1}, \dots, v_{iq(i)}$ 을 각각 포함하는 $q(i)$ 개의 입력 스트링이 $R_{pq}(\Sigma)$ 에 존재한다면 v_i 가 야기하는 불확실성에도 불구하고 x_p 로부터 x_q 까지의 도달가능성이 확실하게 보장된다고 말할 수 있다.

형식 언어(formal language)의 정규표현식(regular expression)에 쓰이는 합집합 연산자(union operator) '+'와 연결(concatenation operator) 연산자 '.'를 이용하여 앞의 조건을 표현할 수 있다. 먼저 $R_{pq}(\Sigma)$ 에 속한 입력 스트링 중 v_i 가 유도하는 불확실한 입력 $v_{i1}, \dots, v_{iq(i)}$ 을 포함하는 원소들을 \cdot 와 + 연산자를 이용하여 결합한다. 또 [8]에서 제안한 방법대로 $v_{i1}, \dots, v_{iq(i)}$ 에 대해서 아래와 같은 연산 규칙을 추가로 설정한다.

$$\beta v_{i1} + \beta v_{i2} + \dots + \beta v_{iq(i)} := \beta v_i, \beta \in (A \cup A_u)^+ \quad (5)$$

현재의 정규표현식이 v_i 에 대한 모든 불확실한 입력을 각각 포함하는 $q(i)$ 개의 스트링을 가지며 각 스트링의 $v_{i1}, v_{i2}, \dots, v_{iq(i)}$ 문자 앞에 연결되는 스트링(prefix)이 동일하면(식 (5)의 좌변) $q(i)$ 개의 불확실한 입력 $v_{i1}, v_{i2}, \dots, v_{iq(i)}$ 는 확실한 한 개의 확실한 입력 v_i 으로 간주할 수 있다(식 (5)의 우변).

결론은 불확실성을 가지는 비동기 머신에 대한 도달가능성을 알기 위해서는 $R_{pq}(\Sigma)$ 의 각 원소에 식 (5)을 포함하는 정규표현식 연산을 적용시켜야 한다. 유도되는 정규표현식이 A_u 에 속한 불확실한 입력 문자를 하나도 가지지 않는다면 이 표현식은 불확실한 상태 천이의 모든 경우의 수에 대한 도달경로를 다 가진다는 뜻이다. 따라서 x_p 에서 x_q 까지의 도달가능성이 보장된다. $R(\Sigma)$ 의 각 entry $R_{pq}(\Sigma)$ 에 속한 스트링으로부터 유도될 수 있는 모든 정규표현식이 추가된 '도달가능성 행렬'을

$R(\Sigma)$ 라고 하자. 상태 x_p 에서 x_q 까지의 도달가능성 여부를 $R(\Sigma)$ 로 나타내면 다음과 같다.

$$x_q \text{가 } x_p \text{로부터 도달가능} \Leftrightarrow R_{pq}(\Sigma) \text{에 } t \in A^+ \text{인 } t \text{가 존재하거나 } A \text{의 원소로만 구성된 정규표현식이 존재} \quad (6)$$

$n \times n$ skeleton 행렬 $K(\Sigma)$ 는 Σ 의 상태간 도달가능성을 간략하게 표현한다^[3-5]. 상태 x_q 가 x_p 로부터 도달가능하면 $K_{pq}(\Sigma)=1$ 이고 도달가능하지 않으면 $K_{pq}(\Sigma)=0$ 이다. 행렬 $R(\Sigma)$ 의 정의와 명제 (6)을 적용하면 불확실한 상태 천이를 가진 Σ 의 $K(\Sigma)$ 는 다음과 같이 유도된다.

$$K_{pq}(\Sigma)=1: R_{pq}(\Sigma) \text{에 } t \in A^+ \text{인 } t \text{가 존재하거나 } A \text{의 원소로만 구성된 정규표현식이 존재}$$

$$K_{pq}(\Sigma)=0: \text{otherwise} \quad (7)$$

III. 모델 정합 포함 문제

그림 1은 입력/상태 비동기 순차 머신을 위한 피드백 제어 시스템이다^[3-5]. Σ 는 제어 대상 비동기 머신이며 C 는 역시 비동기 머신으로 구현되는 피드백 제어기이다. $v \in A$ 는 외부 입력, $u \in A$ 는 제어기가 만드는 제어 입력이며, $x \in X$ 는 Σ 의 출력, 즉 상태 피드백 값이다. 또 Σ_c 는 C 와 Σ 가 결합된 페루프 시스템을 가리킨다.

모델 정합(model matching) 문제는 그림 1의 페루프 시스템 Σ_c 의 안정 상태 동작을 주어진 모델

$$\Lambda = (A, X, x_0, s')$$

와 일치시키는 제어기 C 를 설계하는 문제이다^[3-5]. Λ 는 Σ 와 공통된 A, X, x_0 를 가지나 stable recursion 함수 s' 가 Σ 의 s 와 다르다. 한편 본 논문의 목적인 모델 정합 포함(model matching inclusion) 문제는 Σ_c 가 Λ 의

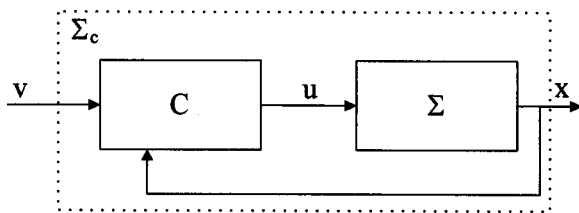


그림 1. 입력/상태 비동기 순차 머신을 위한 상태 피드백 제어 시스템

Fig. 1. State feedback control system for input/state asynchronous sequential machines.

가능한 한 가장 큰 부분 동작이 되게 하는 문제이다^[7,9]. 즉 임의의 상태-입력 조합 $(x,u) \in X \times A$ 에서 Σ_c 가 $s_c(x,u)=x'$ 이면(s_c 는 Σ_c 의 천이 함수) Λ 도 $s'(x,u)=x'$ 이어야 한다. 하지만 그 역(逆)이 반드시 성립하지는 않는다(역까지 성립하면 모델 정합 문제). 또 Σ_c 는 앞의 관계를 만족시키면서 가능한 한 가장 큰 동작(maximally permissive)을 보여야 한다.

기존 연구^[3-5]에서 페루프 시스템 Σ_c 가 모델 Λ 와 정확한 모델 정합을 이루도록 하는 제어기 C 가 존재할 조건은 아래와 같은 skeleton 행렬 사이의 부등식으로 표현되었다.

$$K(\Sigma) \geq K(\Lambda) \quad (8)$$

위 식을 다시 말하면 임의의 상태들에서 머신 Σ 의 도달가능성이 모델 Λ 의 도달가능성보다 항상 커야 한다는 것이다. 즉 Λ 에서 상태 x_q 가 다른 상태 x_p 로부터 도달가능하면 Σ 에서도 동일한 조건이 만족되어야 한다($K_{pq}(\Sigma) \geq K_{pq}(\Lambda), \forall p,q \in \{1, \dots, n\}$). 식 (8)이 성립하지 않으면 모델 정합 문제는 풀리지 않는다.

모델 정합 포함 문제는 식 (8)의 성립여부에 관계없이 페루프 시스템 Σ_c 가 모델 Λ 의 가능한 한 가장 큰 부분 동작을 취하게 만드는 일이다. 관점을 바꾸어 생각하면 이 문제는 주어진 제어 대상 머신 Σ 에 대해서 정확한 모델 정합 문제가 성립하도록 하는 'Λ의 부분 동작'을 찾는 문제와 동일하다. 본 논문에서는 그러한 부분 동작을 보이는 비동기 머신을 Λ^0 이라 하고 Λ^0 을 찾는 알고리즘을 아래와 같이 제안한다.

알고리즘 1: 머신 Σ 와 모델 Λ 가 주어졌을 때 모델 정합 포함 문제를 풀 수 있는 최대 부분 동작 모델 Λ^0 찾기 (Σ 와 Λ 의 상태 집합은 $X=\{x_1, \dots, x_n\}$ 라고 설정)

$\Lambda^0 = \Lambda$ 로 초기 설정

for $p=1:n$

for $q=1:n$

if $K_{pq}(\Sigma) \geq K_{pq}(\Lambda^0)$: 다음 과정을 수행

Else: Λ^0 에서 모든 $x_p \rightarrow x_q$ 상태 천이를 제거

알고리즘 1은 조건 (8)을 만족시키지 못하는 모든 상태 천이를 모델 Λ 에서 제거하는 과정이다. 알고리즘 1에 의해서 부분 동작 모델 Λ^0 과 머신 Σ 사이에는 아래 관계가 성립한다.

$$K(\Sigma) \geq K(\Lambda^0) \quad (9)$$

따라서 페루프 시스템 Σ_c 의 안정 상태 동작을 Λ^0 과 일치시키는 상태 피드백 제어기 C가 존재한다. 알고리즘 1은 Σ 의 도달가능성이 Λ 보다 더 큰 상태 천이는 Λ^0 에 그대로 남겨두므로 Λ^0 의 동작은 (9)를 만족시키는 Λ 의 부분 동작 중 가장 크다고 말할 수 있다.

IV. 페루프 시스템 부분 동작 확장

앞서 기술했듯이 교정 제어가 실제 수행될 때 Σ 가 불확실한 상태 천이 (z_i, v_i) 를 통과한다면 그림 1의 제어기 C는 Σ 가 출력하는 상태 피드백 x 를 보고 정확한 다음 안정 상태 $s(z_i, v_i)$ 를 파악한다. 즉 (z_i, v_i) 에 대한 불확실성이 사라진다. 따라서 $q(i)$ 개의 불확실한 입력 $v_{i1}, v_{i2}, \dots, v_{iq(i)}$ 중 확실한 상태 천이를 나타내는 한 개의 값만 유지되고 나머지 값들은 도달가능성을 체크할 때 빼야 더 정확한 계산이 이루어질 것이다.

본 논문에서는 불확실성을 제거하면서 제어기가 얻는 정보를 이용하여 페루프 시스템의 부분 동작을 확장하는 기법을 제안한다. 제어기가 얻는 시스템의 정확한 상태 천이 정보는 도달가능성 행렬 $R(\Sigma)$ 에 반영한다. $R(\Sigma)$ 은 모델 불확실성을 최대한 고려하여 계산한 결과이기 때문에(Σ 는 모든 불확실성 경우의 수에 대해서 도달가능성을 하나하나 다 가져야 한다) 제어기가 얻는 정보를 $R(\Sigma)$ 에 반영하면 행렬의 도달가능성은 더 커질 것이다.

먼저 교정 제어를 시작하기 전에 모델 불확실성을 고려하여 얻은 도달가능성 행렬과 skeleton 행렬을 각각 $R^0(\Sigma)$, $K^0(\Sigma)$ 라고 다시 표기하자. 또 $K^0(\Sigma)$ 을 알고리즘 1에 적용시켜 얻은 부분 동작 모델을 Λ^0 이라 하자(알고리즘 1에 기술한 표기와 동일). 마지막으로 페루프 시스템을 Λ^0 과 모델 정합시키는 제어기를 C^0 이라 하자. $K^0(\Sigma)$, Λ^0 , C^0 은 아래 관계를 만족시킨다.

$$K^0(\Sigma) \geq \Lambda^0(\Sigma), \quad \Sigma_{ctrls} = \Lambda^0 \quad (10)$$

다음으로 제어기 C^0 을 설계하여 $\Sigma_{ctrls} = \Lambda^0$ 를 구현하는 그림 1의 교정 제어 시스템이 작동하기 시작한 후 Σ 가 불확실한 상태 천이 $(z_1, v_1), \dots, (z_r, v_r)$ 중 하나를 거친다고 가정하자. 상태 피드백을 받아서 정확한 다음 안정 상태를 관측한 C^0 은 이 정보를 가지고 도달가능성 행렬 $R^0(\Sigma)$ 을 갱신한다. 갱신한 도달가능성 행렬을 $R^1(\Sigma)$ 이

라 하고 $R^1(\Sigma)$ 로부터 유도된 skeleton 행렬을 $K^1(\Sigma)$ 라 표기하자. 불확실한 상태 천이가 하나 제거되었으므로 당연히 $K^0(\Sigma) \leq K^1(\Sigma)$ 이다.

이제 새로운 skeleton 행렬 $K^1(\Sigma)$ 을 가지고 알고리즘 1을 다시 수행하여 부분 동작 모델을 구하고, 구한 모델을 Λ^1 이라 명명한다. $K^0(\Sigma) \leq K^1(\Sigma)$ 이므로 Λ^1 을 구할 때 알고리즘 1에 의해서 제거되는 원래 모델 Λ 의 상태 천이의 개수는 Λ^0 일 때보다 더 적다. 다시 말하면 Λ^0 과 비교하여 Λ^1 이 더 넓은 동작 범위를 가진다. 페루프 시스템을 Λ^1 과 일치시키는 상태 피드백 제어기를 C^1 이라 하고 그 다음 제어 과정에서부터 도달가능성 행렬이 다시 갱신되기 전까지는 C^1 을 제어기로 사용한다. 전술한 과정을 반복하면 페루프 시스템은 현재 제어기가 알고 있는 상태 천이 정보를 최대한 활용하여 얻은 부분 동작을 가질 것이다. 식 (10)과 반복 과정에서 유도되는 관계를 일반화하면 아래와 같다.

$$\begin{aligned} K^0(\Sigma) &= K(\Sigma), \quad K^k(\Sigma) \leq K^{k+1}(\Sigma) \\ K^k(\Sigma) &\geq K^k(\Lambda^k), \quad \Sigma_{ctrls} = \Lambda^k, \quad k=0,1,2,\dots \end{aligned} \quad (11)$$

알고리즘 1의 실행과 실제 교정 제어기의 설계, 페루프 시스템 부분 동작 확장을 위한 제어기 추가 등의 과정은 다음 장 예제에서 설명한다.

V. 제어기 설계 예제

1. 모델 정합 포함

그림 2와 같은 상태 흐름도를 가지는 비동기 머신 Σ

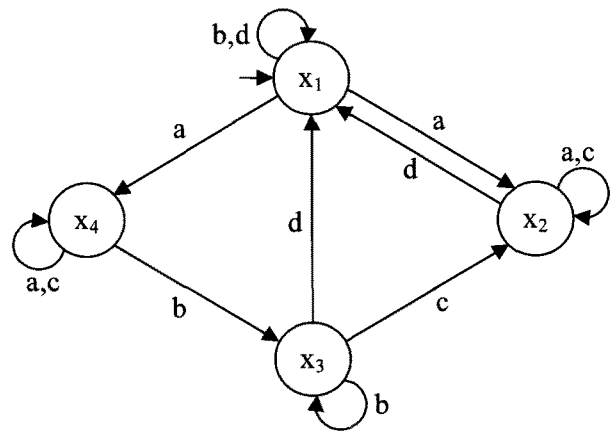


그림 2. 불확실한 상태 천이 (x_1, a) 를 가진 제어 대상 비동기 머신 Σ .

Fig. 2. Controlled asynchronous machine Σ with the uncertain transition (x_1, a) .

가 있다($x_0 = x_1$). 그림에서 볼 수 있듯이 Σ 에는 한 개의 불확실한 상태 천이 (x_1, a)가 존재하며($r=1$), (x_1, a)의 다음 안정 상태는 x_2 와 x_4 중 하나이다(식 (2)에서 $p(1) = 2$). 식 (3)을 이용하여 (x_1, a)의 불확실한 상태 천이를 나타내면 아래와 같다.

$$s(x_1, a_1) = x_2, s(x_1, a_2) = x_4, A_u = \{a_1, a_2\}$$

그림 2로부터 Σ 의 skeleton 행렬 $K^0(\Sigma)$ 를 유도하면 다음과 같다(도달가능성 행렬 $R(\Sigma)$ 의 완전한 기술은 지면 관계상 생략한다).

$$K^0(\Sigma) = \begin{pmatrix} 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 \end{pmatrix} \quad (12)$$

위 행렬에서 주목해야 할 원소는 $K_{12}^0(\Sigma)$ 와 $K_{14}^0(\Sigma)$ 이다. 먼저 $K_{12}^0(\Sigma)$ 를 이끌어내는 도달가능성 행렬 $R(\Sigma)$ 의 해당 원소 $R_{12}(\Sigma)$ 는

$$R_{12}(\Sigma) = \{a_1, a_2bc\}$$

이다. 식 (5)를 이용하여 $R_{12}(\Sigma)$ 로부터 정규표현식을 유도하면 $a_1 + a_2bc = a(\varepsilon + bc)$ 이다(ε 는 빈(empty) 스트링). $R_{12}(\Sigma)$ 에 A 의 원소로만 구성된 정규표현식이 존재하므로 조건 (6)에 따라서 x_2 는 모델 불확실성에도 불구하고 x_1 로부터 도달가능하다. 하지만 $R_{14}(\Sigma) = \{a_2\}$ 이므로 x_1 에서 x_4 까지의 도달가능성은 아직 보장하지 못한다. 따라서 $K_{14}^0(\Sigma) = 0$ 이다($K_{13}^0(\Sigma)$ 도 자동적으로 0이다).

그림 3은 주어진 기준 모델 Λ 이다. Λ 의 도달가능성 행렬을 구하고 skeleton 행렬을 유도하면

$$K(\Lambda) = \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \end{pmatrix} \quad (13)$$

이다. (12)와 (13)을 비교하면 $K^0(\Sigma) \neq K(\Lambda)$ 로서 모델 정합 조건 (8)을 만족시키지 못한다. 그러므로 모델 정합 대신 모델 정합 포함 문제를 구현하는 제어를 설계해야 한다. Σ 와 Λ 를 알고리즘 1에 적용시켜 부분 동작 모델 Λ^0 을 구하면 그림 4와 같다. 그림 3과 4를 비교하면 Λ^0 는 Λ 에서 $x_1 \rightarrow x_4$ 의 상태 천이를 제거한 것과 같다. 이 상태 천이를 제거하면 $x_1 \rightarrow x_4$ 상태 천이는 자동적으로 제거된다.

$K^0(\Sigma) \geq K(\Lambda^0)$ 이 자명하므로 페루프 시스템 Σ_c 를 Λ^0 과 일치하도록 하는 상태 피드백 제어를 설계할 수

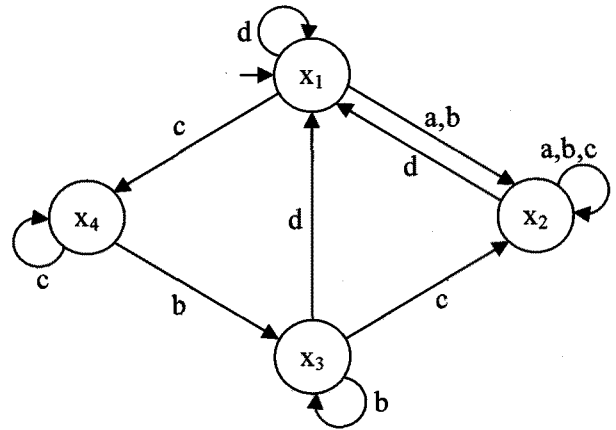


그림 3. 기준 모델 Λ
Fig. 3. Reference model Λ .

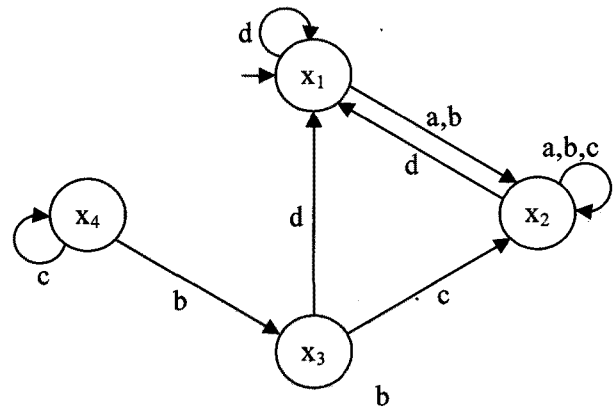


그림 4. 부분 동작 모델 Λ^0
Fig. 4. Reference model Λ^0 .

있다. 이 제어를 앞 장의 명명법을 따라 C^0 이라고 정의한다.

그림 2의 Σ 와 그림 4의 Λ^0 는 (x_1, b)에서 모델 부정합을 보인다. Σ 는 안정 상태 x_1 에서 입력 b 가 들어오면 상태 x_1 에 머무른다. 반면 Λ^0 는 다음 안정 상태 x_2 로 천이한다. 즉 제어기 C^0 의 목적은 (x_1, b)에 대한 페루프 시스템의 안정 상태를 x_1 대신 x_2 로 바꾸는 일이다. C^0 을 아래와 같은 입력/출력 비동기 머신으로 정의하자.

$$C^0 = (A \times X, A, \Xi, \xi_0, \phi, \eta)$$

$A \times X$ 와 A 는 각각 C^0 의 입력과 출력 집합이며(그림 1 참조) Ξ 는 C^0 의 상태 집합, ξ_0 는 초기 상태, ϕ 는 상태 천이 함수, 그리고 η 는 출력 함수이다.

앞에서 도달가능성 행렬의 원소 $R_{12}(\Sigma)$ 는 정규표현식 $a(\varepsilon + bc)$ 를 가진다고 하였다. $a(\varepsilon + bc)$ 는 a 와 abc 두 개의 스트링을 포함하므로 C^0 는 두 경우가 모두 참이라고 가정하고 제어를 설계해야 한다. 본 예제에서는 abc , 즉

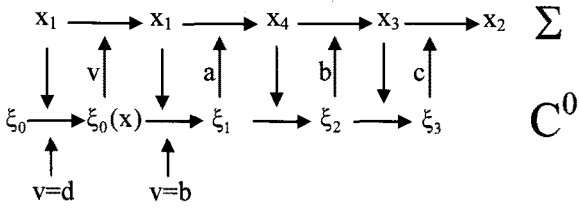


그림 5. $(x_1, a) \rightarrow x_2$ 까지의 교정 동작
Fig. 5. Corrective action for $(x_1, a) \rightarrow x_2$.

$x_1 \rightarrow x_4 \rightarrow x_3 \rightarrow x_2$ 경로에 대한 교정 제어 모듈을 설계하기로 한다. (a에 대한 교정 제어 모듈은 설계될 모듈 내에 포함시킬 수 있다.)

초기 상태 ξ_0 에 있던 C^0 은 Σ 가 x_1 과 안정 조합을 이룰 때, 즉 (x_1, d) 일 때 transition 상태 $\xi_0(x)$ 로 천이한 후 외부 입력이 b로 바뀌기를 기다린다. b 대신 다른 입력이 들어오면 C^0 은 다시 초기 상태 ξ_0 으로 되돌아간다.

외부 입력 b가 들어오는 순간 제어기 C^0 은 교정 동작을 실행한다. 교정 동작에 쓰이는 입력 스트링의 길이가 3이므로 세 개의 보조 상태 ξ_1, ξ_2, ξ_3 이 추가로 필요하다. b를 받은 C^0 은 먼저 상태 ξ_1 로 이동한 후 입력 스트링의 첫 번째 문자 a를 Σ 에 넣어준다. 그런데 만약 $s(x_1, a) = x_2$ 인 상태 천이가 참이라면 Σ 는 원하는 상태인 x_2 에 바로 도착하게 되고 (C^0 이 상태 피드백으로 x_2 를 받음으로써 알게 된다) 교정 동작은 여기서 끝난다. 그렇지 않다면 Σ 는 다음 상태 x_4 로 이동한다. 상태 피드백 x_4 를 받은 C^0 은 두 번째 보조 상태 ξ_2 로 이동한 후 다시 입력 스트링의 두 번째 문자 b를 Σ 에 전달한다. 이런 식으로 계속하면 Σ 는 원하는 상태 x_2 에 도달한다. 이 모든 교정 동작이 클럭 없이 순식간에 진행되므로 외부 사용자에게는 페루프 시스템이 안정 상태 x_1 에 있다가 입력 b에 반응하여 다음 안정 상태 x_2 로 즉시 천이하는 모습만이 관측될 것이다. 그림 5는 이러한 교정 동작을 도식화한 모습이다.

2. 페루프 시스템 부분 동작 확장

앞 절에서 제어기 C^0 을 설계하여 실제로 구동시켰다면 머신 Σ 는 불확실한 상태 천이 (x_1, a) 를 거친다. C^0 은 상태 피드백을 통해서 (x_1, a) 에 대한 정확한 다음 안정 상태를 관측하므로 본 논문에서 제안한 바대로 이 정보를 다음 교정 동작에 활용할 수 있다. (x_1, a) 의 실제 다음 안정 상태가 x_4 라고 하자. 그렇다면 그림 2에서 알 수 있듯이 머신 Σ 는 x_1 에서 x_4 까지 도달가능성을 보유하게 되고 결과적으로 임의의 두 상태간 도달가능성이 보장

된다. 즉 갱신된 도달가능성 행렬로부터 유도되는 두 번째 skeleton 행렬 $K^1(\Sigma)$ 는

$$K^1(\Lambda) = \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \end{pmatrix} \quad (14)$$

과 같이 모든 원소를 1로 가진다. (13)과 (14)로부터 $K^1(\Sigma) \geq K(\Lambda)$ 이므로 페루프 시스템이 모델 Λ 를 정확하게 모사하게 하는 제어기 C^1 이 존재한다. 물론 $K^1(\Sigma)$ 를 알고리즘 1에 적용시켜 나오는 부분 동작 모델의 결과도 $\Lambda^1 = \Lambda$ 이다. 그림 2와 그림 3을 비교하면 Σ 와 Λ 는 (x_1, b) 를 포함하여 다음과 같은 모델 부정합을 보인다.

$$\begin{aligned} s(x_1, a) = x_4 &:: s'(x_1, a) = x_2 \\ s(x_1, b) = x_1 &:: s'(x_1, b) = x_2 \\ s(x_1, c) = \emptyset &:: s'(x_1, c) = x_4 \end{aligned}$$

종합하면 C^1 은 C^0 과 비교하여 페루프 시스템의 부분 동작을 확장시킨다는 사실이 확인되었다. C^1 을 설계하는 방법은 앞 절에서 설명한 C^0 의 방법과 유사하다.

VI. 결 론

본 논문에서는 불확실한 상태 천이를 가진 입력/상태 비동기 순차 머신의 모델 정합 포함 문제를 위한 교정 제어 시스템을 제안하였다. 이번 논문의 주요 목적은 시스템의 부분 고장이나 내부 메모리 비트가 고정되는 사건 등으로 인해서 상태 천이가 불확실해진 비동기 머신을 다루는 것이다. 이전 연구들과 비교하여 본 논문이 가지는 차별성은 i) 모델 정합 문제가 아닌 모델 정합 포함 문제를 해결하는 제어기의 존재 조건 및 설계 과정을 보였다는 사실과, ii) 상태 피드백을 받음으로써 점점 불확실성이 제거되는 머신의 도달가능성을 최대한 활용하여 페루프 시스템이 가질 수 있는 부분 동작을 확장하는 기법을 제안하였다는 점이다.

머신의 출력이 현재 상태와 다른 값으로 나오는 입력/출력 비동기 머신에 대해서 본 논문의 이론을 적용하는 일이 추후 과제로 남아 있다.

참 고 문 헌

[1] N. Nishimura, "Efficient asynchronous simulation of a class of synchronous parallel algorithms,"

- Journal of Computer and System Sciences, vol. 50, no. 1, pp. 98-113, 1995.
- [2] S. M. Nowick, M. E. Dean, D. L. Dill, and M. Horowitz, "The design of a high-performance cache controller: a case study in asynchronous synthesis," in Proceedings of the 26th Hawaii International Conference on System Sciences, pp. 419-427, 1993.
- [3] 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.
- [4] 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.
- [5] J.-M. Yang and J. Hammer, "State feedback control of asynchronous sequential machines with adversarial inputs," International Journal of Control, vol. 81, no. 12, pp. 1910-1929, 2008.
- [6] R. G. Useinov, "Analytical model of radiation induced or single event latchup in CMOS integrated circuits," IEEE Transactions on Nuclear Science, vol. 53, no. 4, pp. 1834-1838, 2006.
- [7] M. D. Dibenedetto, A. Sangiovanni-Vincentelli and T. Villa, "Model matching for finite-state machines," IEEE Transactions on Automatic Control, vol. 46, no. 11, pp. 1726-1743, 2001.
- [8] 양정민, "불확실한 상태 천이를 가진 입력/상태 비동기 머신을 위한 견실 제어," 전자공학회논문지, 제46권 SC 제4호, pp. 39-48, 2009.
- [9] G. Barrett and S. Lafortune, "Bisimulation, the supervisory control problem, and strong model matching for finite state machines," Discrete Event Dynamic Systems: Theory and Applications, vol. 8, no. 4, pp. 377-429, 1998.

저자 소개



양정민(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~현재 대구가톨릭대학교 전자공학과

부교수

<주관심분야 : 비동기 머신 제어, 실시간 시스템 고장 진단 등>



박용국(정회원)

1987년 서울대학교 공과대학 금속공학과 학사 졸업

1988년 미국 미시건대학교 산업공학과 석사 졸업

1995년 미국 오하이오주립대학교 기계생산공학과 박사 졸업

1996년~1998년 삼성자동차 생산기술 선임연구원

1998년~현재 대구가톨릭대학교 기계자동차

공학부 정교수

<주관심분야 : 자동제어, 강건 설계, 생산시스템 설계 등>