논문 2009-46SC-4-6

불확실한 상태 천이를 가진 입력/상태 비동기 머신을 위한 견실 제어

(Robust Control of Input/State Asynchronous Machines with Uncertain State Transitions)

양 정 민*

(Jung-Min Yang)

요 약

전역 클럭 없이 동작하는 비동기 순차 머신은 동기 순차 머신에 비해서 속도나 에너지 소비 면에서 장점을 지닌다. 본 논문에서는 불확실한 상태 천이를 가지는 입력/상태 비동기 머신을 위한 견실 제어기를 제안한다. 논문에서 고려하는 비동기 머신은 모델 불확실성, 내부 고장 등으로 인해서 일부 영역의 상태 천이 함수가 불확실하다. 이번 연구에서는 이러한 비동기 머신을 표현하는 유한 상태 머신 식을 제안한 후 일반화된 도달가능성 행렬을 이용하여 머신의 폐루프 동작이 주어진 정상적인 모델의 동작과 일치하도록 하는 비동기 제어기가 존재할 조건을 규명한다. 또한 기존 연구 결과를 바탕으로 비동기 제어기의 설계 과정을 기술하고 폐루프 시스템의 안정 상태 동작을 분석한다.

Abstract

Asynchronous sequential machines, or clockless logic circuits, have several advantages over synchronous machines such as fast operation speed, low power consumption, etc. In this paper, we propose a novel robust controller for input/output asynchronous sequential machines with uncertain state transitions. Due to model uncertainties or inner failures, the state transition function of the considered asynchronous machine is not completely known. In this study, we present a formulation to model this kind of asynchronous machines and, using generalized reachability matrices, we address the condition for the existence of an appropriate controller such that the closed-loop behavior matches that of a prescribed model. Based on the previous research results, we sketch design procedure of the proposed controller and analyze the stable-state operation of the closed-loop system.

Keywords : Asynchronous Sequential Machines, Model Matching, Uncertain Transitions, Robust Control

Ⅰ. 서 론

이산 사건 시스템(Discrete-Event System)은 외부에서 들어오는 이산적인 신호, 즉 '사건(event)'에 의해서 상태가 바뀌는 시스템이다. 시간의 흐름에 따라서 상태가 결정되는 연속(또는 이산) 시간 시스템(continuous (or discrete) time system)과 대비되는 특징을 지닌 이산 사건 시스템을 제어하는 기법으로는 1980년대부터 연구되어 온 관리 제어(Supervisory Control)가 대표적이다[1].

본 논문에서 다루는 비동기 순차 머신(Asynchronous Sequential Machine)은 넓은 의미로는 이산 사건 시스템에 속한다. 하지만 전역 클럭(clock) 없이 외부 입력의 변화에 따라 즉시 움직이는 비동기 머신의 동작 과정은 일반적인 유한 상태 머신(Finite State Machine)으로는 표현할 수 없는 고유의 특징이다. 또 이러한 특징을 가지는 비동기 머신은 관리 제어나 동기 순차 머신(Synchronous Sequential Machine)을 위한 모델 정합

* 정회원, 대구가톨릭대학교
(Department of Electrical Engineering, Catholic University of Daegu)
접수일자: 2008년10월14일, 수정완료일: 2009년7월2일

(model matching)[2] 등의 방법으로는 제어될 수 없다[3].

본 연구의 기본 개념은 비동기 순차 머신을 설계 대상이 아니라 제어 대상으로 본다는 점이다. 즉 전통적인 피드백 제어(feedback)의 개념을 도입하여 제어 대상 비동기 머신 앞에 제어기를 연결한 후 비동기 머신의 폐루프(closed-loop) 동작을 원하는 목적에 맞게 바꾼다. 즉 제어의 목적은 폐루프 시스템과 모델과의 모델 정합이다. 이러한 기법은 Hammer가 순차 머신에 대해서 처음 제안하였다[4]. 또 비동기 머신 내에 존재하는 크리티컬 레이스(critical race)나 무한 순환(infinite cycle)을 없애거나[2, 5] 비동기 머신으로 침투하는 입력 외란의 영향을 최소화시키는 연구[6], 비결정 모델을 가지는 비동기 머신을 위한 모델 정합 연구[7] 등이 발표되었다.

이번 연구의 목적은 불확실한 상태 천이를 가지는 비동기 머신을 위한 모델 정합을 구현하는 견실 제어기를 제안하는 일이다. 비동기 머신이 장착된 시스템은 우주선(cosmic ray)이나 방사능 등 외란에 의해서 고장을 일으켜[8],[9] 불확실성이 생길 수 있으며, 비동기 머신의 명세를 사용자가 잘 모르는 경우 명확한 상태 천이 함수를 처음부터 정의하지 못하는 경우도 있다.

중요한 것은 본 논문에서 다루는 비동기 머신은 비결정적 동작(nondeterministic behavior)을 가지지만 머신이 추종해야 하는 이상적인 모델은 결정적인 동작(deterministic behavior)을 가진다는 사실이다. 이런 점에서 본 연구는 정반대의 문제, 즉 결정적 비동기 머신이 비결정 모델을 추종해야 하는 문제에 대한 기존 연구[7]나 다른 연구 결과[2, 5~6]와 대비되는 차이점을 지닌다.

이번 연구에서 다루는 비동기 머신은 현재의 상태가 출력으로 나오는 입력/상태 머신(input/state machine)이다. 머신의 출력이 상태와 다른 입력/출력 머신(input/output machine)에 대한 제어기 설계는 추후 연구에서 다루어진다.

본 논문의 구성은 다음과 같다. Ⅱ장에서는 기존 연구 결과를 바탕으로 비동기 머신에 대한 피드백 제어 시스템의 개념과 작동 원리를 기술한다. Ⅲ장에서는 불확실한 상태 천이를 가지는 비동기 머신을 위한 새로운 모델링을 제안한다. Ⅳ장에서는 비동기 머신의 도달가능성을 정량화하고 계산한다. 비동기 머신의 모델 정합 문제를 푸는 제어기가 존재할 조건은 머신의 도달가능성 행렬을 계산함으로써 구해진다[3, 10]. 본 연구에서는 불확실한 상태 천이를 가지는 머신의 도달가능성을 표현하는 방법이 제안된다. Ⅴ장에서는 Ⅳ장의 결과를 바탕으로 비동기 머신을 위한 견실 제어기를 제안한다. Ⅵ장에서는 예제를 통해서 제안된 제어기의 설계 과정을 보여준다. 마지막으로 Ⅶ장에서는 본 논문의 결론을 내린다.

Ⅱ. 비동기 머신 제어 구조

그림 1은 기존 연구[3, 5, 10]에서 제안된 비동기 순차 머신을 위한 피드백 제어 시스템이다. Σ는 제어 대상 비동기 머신이며 C는 역시 비동기 머신으로 구현되는 피드백 제어기이다. v는 외부 입력, u는 제어기가 만드는 제어 입력이며, y는 머신 Σ의 출력이다. Σ가 입력/상태 머신이면 y는 머신의 현재 상태 값 x를 가진다.

비동기 순차 머신에 대한 피드백 제어의 기본 목적은 제어 대상 머신 Σ의 폐루프 시스템 동작을 기준 모델(reference model)의 동작과 일치시키는 일이다. 두 비동기 머신의 동작이 일치한다는 것은 안정 상태(stable state)에서 동일한 입력이 각 머신에 들어갈 때 동일한 출력이 나와야 한다는 의미이다[11]. 제어기 C의 교정 동작은 입력 변화에 따라서 상태가 순식간에 바뀌는 비동기 머신의 특성을 이용하다.

그림 1에서 Σ_c는 C와 Σ가 결합된 폐루프 시스템을 가리킨다. 만약 어떤 상태에서 머신의 동작이 모델과 똑같다면 제어기 C는 아무 일을 하지 않고 외부 입력 v를 그대로 Σ에 전달한다. Σ의 동작이 모델과 차이를 보인다면 제어기는 외부 입력 v와 피드백 y를 이용하여 제어 입력 u를 만들어내 Σ에 전달함으로써 제어를 수행한다. 즉 제어기 C는 외부 입력 v를 "가로챈 후" v 아닌 다른 값 u를 만들어 머신 Σ에 입력한다. 이러한 교정 동작이 전역 클럭 없이 비동기적으로 순식간에 일어나기 때문에 외부 사용자에게는 폐루프 시스템 Σ_c가

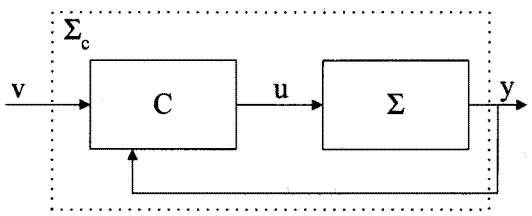

그림 1. 비동기 순차 머신을 위한 상태 피드백 제어 시스템

Fig. 1. State feedback control system for asynchronous sequential machines.

입력 v를 받아서 모델의 동작과 일치하는 원하는 출력 y를 내는 동작만이 관측된다.

Ⅲ. 불확실한 상태 천이 함수

입력/상태 비동기 순차 머신의 논리적 모델은 다음과 같은 유한 상태 머신으로 표현된다.

$$\Sigma = (A, Y, X, x_0, f, h) \quad (1)$$

A는 입력 알파벳 집합, Y는 출력 알파벳 집합, X는 n개의 원소로 이루어진 머신의 상태 집합이다. x_0는 머신의 초기 상태이며 $f:X\times A\rightarrow X$와 $h:X\times A\rightarrow Y$는 각각 머신의 상태 천이 함수(state transition function)와 출력 함수(output function)이다. Σ의 현재 상태를 x_k, 현재 입력을 u_k라 하면 다음 상태는 아래와 같이 정의된다.

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

머신의 스텝(step) k는 입력이나 상태 변수가 변경되었을 때마다 1씩 증가한다. Σ는 입력/상태 머신이므로 출력 함수는 항상 현재 상태 값을 낸다. 즉 Y=X이며 임의의 k에 대해서 $h(x_k,u_k) = x_k$이다.

비동기 순차 머신은 현재 입력이 바뀌지 않는 한 그 값이 바뀌지 않는 안정 상태(stable state)와 머신이 머무르지 못하고 순식간에 다음 상태로의 천이가 일어나는 불안정 상태(unstable state) 등 두 종류의 상태를 가진다. 외부 사용자에게는 비동기 순차 머신의 동작 과정 중 안정 상태에서의 모습만이 유효하므로 안정 상태에서만 비동기 순차 머신을 다룰 수 있는 수학적 모델이 필요하다. 비동기 순차 머신 Σ의 'stable-state 머신 Σ_{ls}'는 이 목적에 맞게 아래와 같이 정의된다[3, 11].

$$\Sigma_{ls} = (A, Y, X, x_0, s, h) \quad (3)$$

위 식에서 상태 천이 함수 f 대신 사용되는 'stable recursion 함수' s는 다음과 같이 정의된다.

$$s : X\times A \rightarrow X,\ s(x,u) := x',\ x\in X,\ u\in A \quad (4)$$

위 식에서 x'은 어떤 유효 조합(valid pair) (x,u), 즉 f(x,u)가 정의되는 상태와 입력 쌍 (x,u)의 '다음 안정 상태(next stable state)', 즉 상태 x에 있던 머신 Σ에 입력 u가 들어왔을 때 머신이 도달하는 첫 번째 안정 상태를 의미한다[3]. 머신 Σ의 상태 천이는 (x,u)에서 시작하여 $(x_1,u) \rightarrow \cdots \rightarrow (x_q,u) \rightarrow (x',u)$가 되며 (x',u)는 Σ의 안정 조합(stable combination)이 된다. (x와 x' 사이에 q 개의 불안정 상태가 있다고 가정한다.) 또 입력 알파벳 대신 입력 스트링(string)을 s의 변수로 설정하면 다음과 같이 일반화할 수 있다.

$$s(x,ut) = s(s(x,u),t),\ x\in X,\ u\in A,\ t\in A^+ \quad (5)$$

위 식에서 A^+는 A에 속한 알파벳으로 이루어지는 길이 1 이상의 스트링 집합을 말한다.

정의 1. $s(x,t)=x'$인 입력 스트링 $t\in A^+$가 존재하면 상태 x'는 상태 x로부터 'stably reachable'하다고 말한다.

본 논문에서는 불확실한 상태 천이를 가지는 r개의 상태-입력 조합이 Σ에 존재한다고 가정한다. 그러한 상태-입력 조합을 $(z_i,u_i)\in X\times A$, i=1,...,r이라 명명하면 이 조합에서 Σ의 stable recursion 함수 s는 아래와 같이 정의된다.

불확실한 상태 천이:

$$s(z_i,u_i) = \{z_{i1},...,z_{ip(i)}\},\ 2\le p(i)\le n-1 \quad (6)$$

즉 머신 Σ는 (z_i,u_i)의 다음 안정 상태가 무엇인지 알지 못하며, $z_{i1},...,z_{ip(i)}$ 중의 하나라는 사실만 알고 있다.

위 불확실성 정의에서 상태 천이 함수 f 대신 stable recursion 함수 s를 쓴 것에 주목한다. f가 불확실함에도 불구하고 s가 결정적 동작을 보일 수도 있다. 예를 들어 어떤 상태-입력 조합 (x,u)의 상태 천이 함수가 불확실하여 $f(x,u)=\{x_1, x_2\}$가 된다고 하자. 그런데 경우에 따라서 $s(x_1,u)=s(x_2,u)=x_3$, 즉 x_1과 x_2가 수렴하는 다음 안정 상태가 x_3로 동일하게 나올 수도 있다. 앞서 설명했듯이 외부 사용자에게는 비동기 머신의 안정 상태에서의 동작만이 관측되므로 (x,u)는 정상적인 상태-입력 조합으로 간주되어야 한다.

Ⅳ. 도달가능성 계산

1. 도달가능성 행렬

비동기 머신 피드백 제어의 핵심은 머신의 도달가능성(reachability)을 파악하는 일이다. 입력/상태 비동기 머신 Σ가 어떤 상태-입력 조합 (x,u)에서 주어진 모델의 동작과 차이를 보인다고 가정하자. 즉 $s(x,u)=x_1$이

나 모델의 동작은 $s'(x,u)=x_2$이다. (s'는 모델의 stable recursion 함수이다.) 피드백 제어기를 통해서 만들어진 폐루프 시스템 Σ_c가 안정 상태에서 모델의 동작과 일치하기 위해서는 머신 Σ가 x에서 x_2까지 stably reachable할 수 있는(정의 1) 경로가 존재해야 한다[3],[5],[6]. 이전 연구에서는 이러한 머신의 도달가능성을 표시하기 위해서 'skeleton 행렬'이라는 행렬을 머신의 특성에 맞게 정의해서 사용하였다[3, 5, 12].

본 연구에서는 불확실한 상태 천이를 가지는 비동기 머신을 위해 확장된 skeleton 행렬을 제안한다. 먼저 머신 Σ의 stable transition 행렬[3]을 계산해야 한다.

정의 2.[3, 12] 비동기 머신 Σ의 상태 집합을 $X=\{x_1, ...,x_n\}$이라고 하자. 'm차 stable transition 행렬 (n×n) $R^m(\Sigma)$'의 원소 $R^m(\Sigma)(i,j)$은 다음과 같이 정의된다.

i) $R^m(\Sigma)(i,j) = \{t\in A^+ | s(x_i,t)=x_j, |t|=m\}$

ii) $R^m(\Sigma)(i,j) = N$ (N은 $N\notin A$인 알파벳) (i)에 해당하는 t가 존재하지 않을 때)

위 정의에 따르면 $R^m(\Sigma)$은 두 상태 사이에 존재하는 길이가 m인 가능한 모든 stable transition 입력 스트링을 원소로 가진다. 만약 그러한 입력 스트링이 존재하지 않는다면 N을 원소로 가진다.

정의 3.[3, 12] 비동기 머신 Σ의 상태 집합을 $X=\{x_1, ...,x_n\}$이라고 하자. 'stable transition 행렬 (n×n) $R^{(n-1)}(\Sigma)$'의 원소 $R^{(n-1)}(\Sigma)(i,j)$은 다음과 같이 정의된다.

$$R^{(n-1)}(\Sigma)(i,j) = \uplus_{m=1,...,n-1} R^m(\Sigma)(i,j)$$

위 정의에서 $\uplus$ 연산자는 합집합 $\cup$ 연산에 N을 제외시키는 일을 추가한 연산이다. 예를 들어 어떤 두 집합 {a, N}, {b}가 있을 때(a,b는 A의 원소) {a,N}$\uplus${b}={a,b}이다. $R^{(n-1)}(\Sigma)$는 길이가 n-1 이하인 모든 입력 스트링을 원소로 가지는 행렬이다. 이전 연구[3, 10]에서는 무한 순환이 없는 비동기 머신 Σ가 n개의 상태를 가진다면(#X=n) stable transition 행렬 $R^{(n-1)}(\Sigma)$을 구하면 비동기 머신의 가능한 모든 동작을 표현할 수 있다는 사실이 증명되었다.

정리 1.[3] 비동기 머신 Σ의 상태 집합을 $X=\{x_1, ...,x_n\}$이라고 하자. Σ가 무한 순환을 가지지 않는다면 $R^{(n-1)}(\Sigma)(i,j)$은 상태 x_i에서 x_j까지 천이하는 모든 입력 스트링을 포함한다(i,j=1,...,n).

증명. 기존 연구 [3]의 Lemma 3.9 참조.

stable transition 행렬 $R^{(n-1)}(\Sigma)$의 계산 과정은 다음 예제에서 논한다.

예제 1. 표 1과 같은 stable recursion 함수 s를 가지는 입력/상태 비동기 머신 $\Sigma=(A,X,X,x_0,f,h)$를 가정하자. 편의상 $\Sigma_{ls}=(A,X,X,x_0,s,h)$, 즉 stable-state 머신의 상태 천이만 표시한다. 실제로는 두 상태 천이 사이에 불안정한 상태들이 포함될 수도 있다. 머신 Σ의 입력과 상태 집합은 각각 A={a, b, c}, $X=\{x_1, x_2, x_3, x_4\}$이며 초기 상태는 $x_0=x_1$이라고 설정한다.

표 1에 나와 있듯이 Σ는 (x_2,a)에서 불확실한 상태 천이를 가진다. $s(x_2,a)=\{x_3, x_4\}$, 즉 (x_2,a)의 다음 안정 상태는 x_3와 x_4 중의 하나이다. 상태 집합 X의 원소가 4개이므로(n=4) stable transition 행렬은 길이가 4-1=3인 입력 스트링만 고려하면 된다(정의 3).

지면 관계상 x_2에서 x_1로 이동하는 상태 천이만을 구한다. 다른 상태에서의 천이도 비슷한 방법으로 구해진다. 먼저 1차 stable transition 행렬은 표 1로부터 아래와 같이 나온다.

$$R^1(\Sigma)(2,1) = \{N\} \quad (7)$$

즉 $s(x_2,u)=x_1$을 만족시키는 단위 입력 $u\in A$는 존재하지 않는다. 마찬가지로 2차 및 3차 stable transition 행렬을 각각 구하면 다음과 같다.

$$R^2(\Sigma)(2,1) = \{ad, ac\} \quad (8)$$

$$R^3(\Sigma)(2,1) = \{bad, cad, aad, adc, add, bac, cac, aac, acc, acd, adc\} \quad (9)$$

표 1. 비동기 머신 Σ의 stable recursion 함수 s
Table 1. Stable recursion function s of Σ.

s(x,u)	a	b	c	d
x_1	-	x_2	x_1	x_1
x_2	$\{x_3, x_4\}$	x_2	x_2	-
x_3	x_3	-	x_2	x_1
x_4	x_4	-	x_1	x_1

정의 3과 (7)~(9)의 결과를 이용하여 stable transition 행렬의 (2,1) 원소 $R^{(3)}(\Sigma)(2,1)$을 구하면 다음과 같다.

$$R^{(3)}(\Sigma)(2,1) = \{ad, ac, bad, cad, aad, adc, add, bac, cac, aac, acc, acd, adc\} \quad (10)$$

2차 및 3차 stable transition 행렬은 $R^{1}(\Sigma)$을 거듭제곱해서 쉽게 구할 수 있다(기존 연구[3, 10, 12] 참조).

2. 확장된 skeleton 행렬

비동기 머신 Σ의 폐루프 시스템 동작이 만족시켜야 하는 모델을 $\Lambda=(A,X,X,x_0,s',h)$라고 하자. s'는 Λ의 stable recursion 함수이다. Λ는 불확실한 상태 천이를 가지고 있지 않은 정상적인 비동기 머신이다. Σ가 입력/상태 머신이므로 모델 Λ가 Σ와 동일한 상태 집합을 가져야 모델 정합 문제가 성립한다[3, 6]. 또 Λ는 stable-state 머신으로 주어져도 일반성을 잃지 않는다.

앞 장에서 설명했듯이 폐루프 시스템의 안정 상태 동작 Σ_{cls}가 모델 Λ의 동작과 일치하기 위해서는 임의의 상태에서 머신 Σ의 도달가능성이 모델 Λ의 도달가능성보다 항상 커야 한다. 다시 말하면 모델에서는 정의된 상태 천이가 머신에서는 정의되지 않으면 주어진 제어 목적을 절대 달성할 수 없다. skeleton 행렬은 바로 머신과 모델 사이의 도달가능성 비교를 간단한 부등식으로 보여주기 위해서 도입된다.

정의 4. 상태 집합 $X=\{x_1,...,x_n\}$을 가지는 입력/상태 비동기 머신 Σ의 stable transition 행렬을 $R^{(n-1)}(\Sigma)$이라고 하자. Σ의 skeleton 행렬 $K(\Sigma)$는 다음과 같이 정의된다.

i) $K(\Sigma)(i,j) = 1$ if $R^{(n-1)}(\Sigma) \neq N$
ii) $K(\Sigma)(i,j) = 0$ if $R^{(n-1)}(\Sigma) = N$

상태 x_j가 x_j로부터 stably reachable하면 $K(\Sigma)(i,j)=1$이고 그렇지 않으면 $K(\Sigma)(i,j)=0$이다. 머신 Σ와 모델 Λ의 skeleton 행렬이 주어졌을 때 모델 정합을 위한 그림 1의 피드백 제어기 C가 존재할 필요충분조건은 아래와 같이 나온다[3, 10]. 아래 정리에서 Σ는 불확실한 상태 천이를 가지고 있지 않는 정상적인 비동기 머신이다.

정리 2. 상태 집합 $X=\{x_1,...,x_n\}$을 가지는 입력/상태 비동기 머신 Σ와 모델 Λ에 대해서 다음 두 명제는 동치(equivalent)이다.

(i) Σ의 폐루프 시스템 Σ_{cls}의 동작이 안정 상태에서 모델 Λ의 동작과 일치하도록 만드는 피드백 제어기 C가 존재한다.

(ii) $K(\Sigma) \geq K(\Lambda)$

본 논문에서 다루는 과제는 비동기 머신 Σ에 불확실한 상태 천이가 존재하기 때문에 stable transition 행렬 $R^{(n-1)}(\Sigma)$로부터 skeleton 행렬 $K(\Sigma)$를 바로 구할 수 없다는 문제이다. 이러한 불확실한 상태 천이를 다루기 위해서는 새로운 기법이 필요하다.

본 논문에서는 불확실한 상태 천이가 포함된 입력 스트링 집합 안에 결정적인(deterministic) 입력 스트링이 존재하는지를 알려주는 새로운 연산을 제안한다. Σ가 식 (6)에서 정의된 불확실한 상태 천이를 가진다고 가정하고 입력 알파벳 $a \in A$가 $a=u_i$라고 하자($1 \leq i \leq r$). 이론을 간단하게 기술하기 위해서 식 (6)에서 $p(i)=2$라고 설정한다. 즉 $s(z_i,a)=\{z_{i1}, z_{i2}\}$이다.

정의 3에서 정의된 stable transition 행렬의 (i,j)번째 원소 $R^{(n-1)}(\Sigma)(i,j)$가 가지는 입력 스트링 안에 a가 존재한다고 가정한다. 편의상 $R^{(n-1)}(\Sigma)(i,j)$ 안에 다른 불확실한 상태 천이는 없다고 가정한다. 그런데 입력 a는 식 (6)에 속하는 불확실한 상태 천이 외에 정상적인 상태 천이에서도 정의될 수 있다. 이 사실을 표시하기 위해서 상태 z_i에서 z_{i1} 또는 z_{i2}로 천이될 때 정의된 입력 a를 각각 a_1, a_2로 표기한다. 식 (6)에 속하는 다른 불확실한 상태 천이도 비슷하게 표기한다.

본 논문에서는 $R^{(n-1)}(\Sigma)$가 가지는 입력 스트링 중 이렇게 첨자(subscript)가 붙은 입력 알파벳, 즉 불확실한 상태 천이를 포함하는 스트링을 '불확실한 입력 스트링(uncertain input string)'이라고 부르고, 불확실한 상태 천이를 하나도 가지고 있지 않은 스트링을 '결정적 입력 스트링(deterministic) input string)'이라고 명명한다.

$R^{(n-1)}(\Sigma)(i,j) = \{a_1bc, a_2cd\}$라고 가정하자. a_1bc와 a_2cd는 개별적으로 보면 모두 불확실한 입력 스트링이다. 즉 a_1와 a_2 중 한 개만 실제 발생하는 입력이며 다른 한 개는 정의되어 있지만 현재 동작에서는 발생하지 않는 입력이다. 그런데 $p(i)=2$라고 가정했으므로 a_1과 a_2는 $s(z_i,a)$에서 정의된 모든 다음 안정 상태로 갈 수 있는 입력 알파벳이다. 이러한 경우 두 개의 불확실한 상태 천이를 하나로 묶어서 결정적인 상태 천이로 간주할

수 있다. 이번 연구에서는 아래와 같이 강조 문자를 사용하여 그러한 결합된 결정적 상태 천이(combined deterministic transition)를 표시한다.

$$a_1bc + a_2cd = \mathbb{a}(bc+cd) \quad (11)$$

$\mathbb{a}(bc+cd)$는 실제로는 두 개의 불확실한 입력 스트링을 품고 있지만 하나의 결정적인 입력 스트링이라고 해석한다. 결합된 결정적 상태 천이 공식을 일반화하면 다음과 같다.

정의 5. $(z_i,a)(a=u_i)$는 식 (6)에서 정의된 불확실한 상태 천이이다. $p(i)=k$라 하고 z_i에서 z_{ij}까지의 상태 천이에서 정의된 입력 a를 a_j라 하자$(j=1,...,k)$. k개의 입력 스트링 $a_1p_1,...,a_kp_k$가 $(p_j\in A^*, \forall j)$ stable transition 행렬의 한 원소에 포함된다고 가정하자. '결합된 결정적 입력 스트링(combined deterministic input string)'은 아래와 같다.

$$a_1p_1 + \cdots + pa_kp_k = \mathbb{a}(p_1+\cdots+p_k) \quad (12)$$

정의 5에서 중요한 것은 불확실한 상태 천이를 하나로 결합하기 위해서는 입력 알파벳 하나에 구속된 모든 불확실성을 포함해야 한다는 사실이다. 식 (12)에서 k는 p(i), 즉 식 (6)에서 정의된 모든 불확실한 다음 안정 상태 개수이다. $a_1,...,a_k$ 중 stable transition 행렬의 한 원소 안에 포함되지 않는 입력이 한 개라도 존재하면 결정적인 상태 천이를 보장하지 못한다.

식 (12)의 형태도 주목해야 한다. 예를 들어 $p(i)=2$이고 두 개의 입력 스트링 aa_1b, ca_2가 stable transition 행렬의 한 원소 내에 존재한다고 가정하자. 가능한 모든 불확실한 상태 천이를 나타내는 입력 a_1과 a_2가 입력 스트링 안에 모두 존재하지만 앞에 붙는 스트링(prefix string)이 서로 다르기 때문에 aa_1b과 ca_2는 결합될 수 없다. 즉 결정적 동작을 보장할 수 없다. aa_1b와 ca_2 대신 ca_1b와 ca_2가 행렬 내에 있다면 $c\mathbb{a}(b+\varepsilon)$로 결합할 수 있다($\varepsilon$는 빈 스트링(empty string)). 입력 스트링 사이의 결합 과정은 일반 정규 언어(regular language)에서 정의된 연산과 유사하게 이루어진다[13].

불확실한 상태 천이가 존재하는 비동기 머신 Σ이 가지는 확장된 skeleton 행렬의 정의와 계산 방법은 다음과 같다.

알고리듬 1. 불확실한 상태 천이가 존재하는 비동기 머신 Σ의 skeleton 행렬 계산 (#X=n)

1) 정의 3에서 정의된 stable transition 행렬 $R^{(n-1)}(\Sigma)$을 구한다.
2) 식 (6)을 이용하여 불확실한 상태 천이를 일으키는 입력 알파벳에 첨자를 붙인다.
3) 정의 5의 입력 스트링 연산을 이용하여 결합된 결정적 입력 스트링을 구한다.
4) 불확실한 상태 천이가 존재하는 비동기 머신 Σ가 가지는 skeleton 행렬을 $\mathbb{K}(\Sigma)$로 표기한다. $\mathbb{K}(\Sigma)$는 다음과 같이 정의된다. 먼저 $R^{(n-1)}(\Sigma)(i,j)$가 결정적 입력 스트링이나 결합된 결정적 입력 스트링을 적어도 하나 가지고 있으면 $\mathbb{K}(\Sigma)(i,j)=1$이다. $R^{(n-1)}(\Sigma)(i,j)$가 불확실한 입력 스트링이나 N만 포함하고 있으면 $\mathbb{K}(\Sigma)(i,j)=0$이다.

예제 2. 예제 1의 비동기 머신 Σ을 생각하자. $s(x_2,a)=\{x_3, x_4\}$이므로 정의 5에 따라서 x_2에서 x_3까지의 상태 천이를 일으키는 a를 a_1, x_2에서 x_4까지의 상태 천이를 일으키는 a를 a_2라고 정의한다. $R^{(3)}(\Sigma)(2,1)$ 안의 스트링은 다음과 같이 표기된다.

$$R^{(3)}(\Sigma)(2,1) = \{a_1d, a_2c, ba_1d, ca_1d, a_1ad, a_1dc, a_1dd, a_2c, ca_2c, a_2ac, a_2cc, a_2cd, a_1dc\}$$

정의 5에서 도입한 연산을 $R^{(3)}(\Sigma)(2,1)$의 원소들에 적용하면 아래와 같이 나온다.

$$\begin{aligned} R^{(3)}(\Sigma)(2,1) &= \{a_1d, a_1ad, a_1dc, a_1dd, a_1dc, a_2c, a_2ac, a_2cc, a_2cd, ba_1d+ba_2c, ca_1d+ca_2c\} \\ &= \{a_1d, a_1ad, a_1dc, a_1dd, a_1dc, a_2c, a_2ac, a_2cc, a_2cd, ba(d+c), ca(d+c)\} \end{aligned}$$

결합된 결정적 입력 스트링 $ba(d+c)$와 $ca(d+c)$이 존재하므로 x_1은 x_2로부터 stably reachable하다고 말할 수 있다. 따라서 알고리듬 1에 의해서 $\mathbb{K}(\Sigma)(2,1)=1$이다.

알고리듬 1에서 정의된 skeleton 행렬 $\mathbb{K}(\Sigma)$을 이용하면 불확실한 상태 천이를 가지는 비동기 머신 Σ의 폐루프 동작이 모델 Λ와 일치하도록 하는 피드백 제어기 C가 존재할 조건을 표현할 수 있다. 아래 조건은 정상적인 머신에 대한 조건을 명시하는 정리 2를 일반화한 것이라고 해석될 수 있다.

정리 3. 식 (6)의 불확실성이 존재하는 입력/상태 비

동기 머신 Σ와 정상적인 모델 Λ에 대해서 다음 두 명제는 동치(equivalent)이다(상태 집합은 $X=\{x_1,...,x_n\}$이다).

(i) Σ의 폐루프 시스템 Σ_{cls}의 동작이 안정 상태에서 모델 Λ의 동작과 일치하도록 만드는 피드백 제어기 C가 존재한다.

(ii) $\mathbb{K}(\Sigma) \geq K(\Lambda)$

V. 견실 제어기 설계

1. 기본 제어 모듈

본 장에서는 제어기 C의 설계에 대해서 기술한다. 설계 과정을 예시하기 위해서 제어 목적을 다음과 같이 설정한다. Σ가 어떤 상태 $z\in X$와 안정 조합을 이룰 때 입력 $b\in A$가 들어오면 Σ는 다음 안정 상태 z_1로 천이한다. 즉 $s(z,b)=z_1$이다. 그런데 모델 Λ의 동작은 $s'(z,b)=z_2(\neq z_1)$이다. 즉 상태-입력 조합 (z,b)에서 모델 부정합(model mismatch)이 발생한다. 제어기 C의 역할은 폐루프 시스템이 입력 b가 들어오면 z에서 z_2로 천이하도록 해주는 일이다.

정리 3의 조건 ii)는 모델 Λ에서 어떤 상태가 다른 상태로부터 stably reachable하면 머신 Σ도 그렇다는 것을 의미한다. 따라서 $s(z,t)=z_2$인 입력 스트링 $t\in A^+$가 항상 존재한다. 제어기 C는 t의 존재를 이용하여 폐루프 동작을 변경시킨다.

그림 1을 보면 제어기 C는 외부 입력 v, Σ의 상태 피드백 y(=x)를 입력으로 가지므로 C의 입력 집합은 $A\times X$이다. C가 만드는 제어 입력 u는 머신 Σ의 입력으로 들어가므로 C의 출력 집합은 A이다. C를 유한 상태 머신으로 표현하면 아래와 같다.

$$C = (A\times X,\ A,\ \Xi,\ \xi_0,\ \Phi,\ \eta) \tag{13}$$

위 식에서 Ξ은 C의 상태 집합이며 ξ_0는 초기 상태, Φ는 recursion 함수, η는 출력 함수이다. 폐루프 시스템 Σ_c는 C와 Σ로 이루어진 복합 시스템이므로 상태 집합 $X\times\Xi$, 입력 집합 A를 가진다. f_c와 h_c를 각각 Σ_c의 상태 천이 함수, 출력 함수라고 정의하면 폐루프 시스템의 유한 상태 머신 표현식은 아래와 같다.

$$\Sigma_c = (A,\ X,\ X\times\Xi,\ (x_0,\xi_0),\ f_c,\ h_c) \tag{14}$$

Σ가 입력/상태 머신이므로 출력 함수 h_c는 C의 모든 상태 $\xi\in\Xi$와 Σ_c의 모든 유효 조합 $((x,\xi),v)$에 대해서 $h_c((x,\xi),v):=x$와 같이 되는 항등 함수(identity function)가 된다.

초기 상태 ξ_0에 있던 C는 Σ가 z와 안정 조합을 이룰 때 Ξ에 속한 어떤 상태 $\xi_0(x)$로 천이한다. ($\xi_0(x)$를 보통 C의 transition 상태[5]라고 부른다) C는 ξ_0에서 $\xi_0(x)$로 이동함으로써 상태 z에서의 동작을 변경할 준비를 한다. 따라서 C의 천이 함수 Φ는 아래와 같이 정의된다.

$$\begin{aligned}\Phi(\xi_0,(x,v)) &= \xi_0 \qquad \forall (x,v)\notin\{z\}\times U(z)\\ \Phi(\xi_0,(z,v)) &= \xi_0(x) \quad \forall v\in U(z)\end{aligned} \tag{15}$$

위 식에서 $U(z)\subset A$는 상태 z와 안정 조합을 이루는 모든 입력들의 집합을 말한다. C가 아직 어떠한 제어 동작도 하지 않으므로 출력 함수 값으로 외부 입력을 그대로 전달해준다.

$$\eta(\xi_0,(x,v)) = v \quad \forall (x,v)\in X\times A \tag{16}$$

$\xi_0(x)$에서 제어기는 입력 b의 발생을 기다린다. 입력이 바뀌지 않는 한 Σ를 계속 상태 z에 머무르게 하기 위해서 C는 $w\in U(z)$인 입력 w를 하나 정하여 출력 함수 값으로 지정한다.

$$\eta(\xi_0(x),(z,v)) = w \quad \forall v\in A \tag{17}$$

입력 b가 들어오는 순간 제어기 C는 교정 동작을 실행한다. $s(z,t)=z_2$인 t를 $t=u_1u_2\cdots u_m$라 하자. 또 상태 z에서 t가 들어올 때 Σ가 거치는 중간 상태들을 $x_1, x_2,...,x_{m-1}$이라 하자. 상태 z에서 스트링 t가 들어올 때 Σ가 거치는 안정 조합들은 다음과 같다.

$$(x_1,\ u_1)\rightarrow(x_2,\ u_2)\rightarrow\cdots\rightarrow(x_{m-1},\ u_{m-1})\rightarrow(z_2,\ u_m) \tag{18}$$

비동기 머신을 위한 제어의 핵심은 폐루프 시스템 안에서 이러한 안정 조합이 과도 조합의 특성을 보이게 만드는 일이다. 즉 머신 Σ가 상태 z에서 입력 b를 받는 즉시 (18)의 상태 천이가 연쇄적으로 일어나게 해서 극히 짧은 시간에 폐루프 시스템이 z_2를 다음 안정 상태로 취하게 한다. (즉 외부 사용자는 중간 상태 $x_1, x_2,...,x_{m-1}$를 감지하지 못한다.) 먼저 $\xi_0(x)$에서 제어기의 동작은 아래와 같이 정의된다.

$$\begin{aligned}\Phi(\xi_0(x),(z,v)) &= \xi_0(x) \quad \forall v\in U(z)\\ \Phi(\xi_0(x),(z,v)) &= \xi_0 \qquad \forall v\notin U(z)\\ \Phi(\xi_0(x),(z,b)) &= \xi_1\end{aligned} \tag{19}$$

$\xi_1\in\Xi$은 상태 z_2로 가기 위해서 제어기 C가 이동하는

첫 번째 상태이다.

ξ_1로 천이한 후 C는 Σ에 입력 스트링 t의 첫 번째 알파벳 u_1을 제어 입력으로 넣는다. u_1을 받은 Σ는 첫 번째 중간 상태 x_1로 천이한다. x_1을 상태 피드백으로 받은 C는 다시 다음 상태 $\xi_2 \in \Xi$로 옮겨 가고 두 번째 입력 알파벳 u_2를 제어 입력으로 준다. 이러한 연쇄적인 C와 Σ의 동작을 구현하면 다음과 같다.

$$\begin{aligned} \Phi(\xi_i,(x_i,b)) &= \xi_{i+1} \\ \eta(\xi_i,(x_i,b)) &= u_i, \quad i=1,\ldots,m-1 \end{aligned} \tag{20}$$

마지막으로 상태 ξ_m로 옮겨간 C는 외부 입력이 다시 다른 값으로 바뀔 때까지 t의 마지막 알파벳 u_m을 제어 입력으로 넣어 Σ를 z_2에 머무르게 한다.

$$\begin{aligned} \Phi(\xi_m,(z_2,b)) &= \xi_m \\ \Phi(\xi_m,(x,v)) &= \xi_0 \qquad \forall (x,v) \neq (z_2,b) \\ \eta(\xi_m,(x,v)) &= u_m, \qquad \forall (x,v) \in X \times A \end{aligned} \tag{21}$$

이런 식으로 설계된 제어기를 그림 1과 같이 연결하면 폐루프 시스템은 안정 상태 z에서 입력 b가 들어오면 z_1 대신 z_2로 천이하는 교정 동작을 구현하게 된다. 제어기가 가지는 상태는 ξ_0, $\xi_0(x)$, $\xi_1,\ldots,\xi_m$로 총 m+2개이다.

(z,b) 조합 외의 다른 상태-입력에서 벌어지는 모델 부정합에 대한 제어기도 위에서 기술한 방법과 비슷하게 설계된다. 설계된 각 제어기들은 기존 연구[5]에서 사용되었던 연산 "∨"을 이용하여 하나의 통합 제어기로 결합될 수 있다.

2. 견실 제어기 설계 규칙

불확실한 상태 천이 (6)이 존재하는 비동기 머신을 위한 견실 제어기를 설계할 때에는 교정을 위한 입력 스트링 t를 선택하는 일이 중요하다. 입력 스트링을 선택하는 기준은 제어기의 상태를 줄이면서 불확실한 상태 천이가 교정 경로에 포함되지 않도록 하는 것이다. 피드백 제어기의 상태 개수는 폐루프 시스템을 실제 구현할 시 중요한 비용(cost)으로 작용한다. 또 앞에서 설명했듯이 불확실한 상태 천이는 고장 등의 문제를 품고 있을 수 있으므로 제어 동작 시 되도록이면 피해야 한다.

비동기 머신 Σ의 상태 집합을 $X=\{x_1,x_2,\ldots,x_n\}$이라 하고 Σ가 안정 상태 x_i에 있을 때 입력 u를 받아 상태 x_j로 가도록 하는 제어기를 설계한다고 가정하자. 정리 3에 의해서 $\mathbb{K}(\Sigma)(i,j)=1$이므로 제어기 설계를 위한 입력 스트링을 $R^{(n-1)}(\Sigma)(i,j)$에서 하나 찾을 수 있다. 본 연구에서는 다음과 같은 입력 스트링 선택 규칙을 제안한다.

i) $R^{(n-1)}(\Sigma)(i,j)$가 결정적 입력 스트링만을 가지고 있을 때: $R^{(n-1)}(\Sigma)(i,j)$에 속하는 스트링 중 길이가 가장 짧은 스트링을 하나 선택하여 t로 사용한다. 앞 절에서 제어기의 상태 개수는 m+2라고 나왔다(m=|t|).

ii) $R^{(n-1)}(\Sigma)(i,j)$가 결합된 결정적 입력 스트링만을 가지고 있을 때: 정의 5에서 확인할 수 있듯이 결합된 결정적 입력 스트링은 실제로는 여러 개의 불확실한 입력 스트링의 합집합이다. 따라서 결합된 결정적 입력 스트링을 피드백 제어기에서 사용한다면 불확실한 입력 스트링들이 만드는 교정 경로들을 모두 구현해야 한다. $R^{(n-1)}(\Sigma)(i,j)$가 결합된 결정적 입력 스트링만을 가지고 있다면 그 중 불확실한 입력 스트링 길이의 최대값이 가장 짧은 것을 교정 제어기가 사용하도록 한다. 예를 들어 $R^{(n-1)}(\Sigma)(i,j)=\{ab(c+d), ac(ef+g)\}$라고 하자. ab(c+d)가 포함하는 불확실한 입력 스트링은 최대 길이가 3이다. 그런데 ac(ef+g) 안에 있는 불확실한 입력 스트링의 최대 길이는 4이므로(acef) 교정 제어기는 ab(c+d)를 사용한다.

iii) $R^{(n-1)}(\Sigma)(i,j)$가 결정적 입력 스트링과 결합된 결정적 입력 스트링을 모두 가지고 있을 때: 교정 동작이 이루어질 때 머신이 불확실한 상태 천이를 통과하지 않도록 하기 위해서는 결정적 입력 스트링을 사용해야 한다. 따라서 i)과 마찬가지로 $R^{(n-1)}(\Sigma)(i,j)$에 속한 결정적 입력 스트링 중 최소 길이를 가지는 원소를 찾아서 제어 입력으로 사용한다.

VI. 제어기 설계 예제

예제 1에 나온 비동기 머신 Σ를 위한 피드백 제어기를 설계한다. 제어 목적은 폐루프 시스템 Σ_c의 동작을 표 2와 같은 stable recursion 함수 s'를 가지는 정상적인 모델 Λ의 동작과 일치시키는 일이다. 모델 정합 문제가 성립하기 위해서 Σ와 Λ는 동일한 상태 및 입력 집합을 가진다.

알고리듬 1과 기존 연구 결과를 바탕으로 Σ과 Λ의 skeleton 행렬을 각각 구하면 아래와 같다.

$$\mathbb{K}(\Sigma)=\begin{pmatrix}1&1&0&0\\1&1&0&0\\1&1&1&0\\1&1&0&1\end{pmatrix},\ \mathbb{K}(\Lambda)=\begin{pmatrix}1&1&0&0\\1&1&0&0\\1&1&1&0\\1&1&0&1\end{pmatrix} \tag{22}$$

표 2. 모델 Λ의 stable recursion 함수 s'
Table 2. Stable recursion function s' of model Λ.

s'(x,u)	a	b	c	d
x_1	-	x_2	x_1	x_1
x_2	-	x_2	x_2	x_1
x_3	x_3	-	x_2	x_1
x_4	x_4	-	x_1	x_1

$\mathbb{K}(\Sigma)\geq K(\Lambda)$이므로 정리 3에 의해서 모델 정합을 이룩하는 피드백 제어기가 존재한다.

표 1과 표 2를 비교하면 (x_2,d)에서 모델 부정합이 생긴다는 사실을 알 수 있다. 즉 $s(x_2,d)$는 정의되지 않은 반면 모델 Λ에서는 $s'(x_2,d)=x_1$로 천이한다. 따라서 폐루프 시스템이 x_2에 있을 때 d가 들어오면 x_1로 천이하도록 하는 제어기를 꾸며야 한다. 한편 모델 Λ에서 $s'(x_2,a)$는 정의되어 있지 않으므로 머신 Σ가 안정 상태 x_2에 있을 때 입력 a가 들어오면 Σ에 아무런 제어 입력 u를 넣어 주지 않으면 그 부분에서의 모델 정합은 간단하게 구현된다.

제어기를 구현하기 위해서 예제 2에서 구한 $R^{(3)}(\Sigma)(2,1)$을 다시 보면 결합적 결정적 입력 스트링 ba(d+c)와 ca(d+c)가 $R^{(3)}(\Sigma)(2,1)$에 포함되어 있다. 두 스트링의 길이가 같으므로 둘 중 임의의 스트링을 하나 선택할 수 있다. 본 예제에서는 ba(d+c)을 선택한다. 앞에서 기술했듯이 이 스트링은 실제로는 bad, bac 두 개의 불확실한 입력 스트링을 포함한다. 입력 알파벳 a가 발생하면 이 중 어느 경로로 머신이 움직일지 모르므로 두 스트링에 대해서 각각 제어기를 설계하여 결합해야 불확실성에 대한 견실성(robustness)을 이룩할 수 있다. bad를 이용한 제어기를 C_1, bac를 이용한 제어기를 C_2라고 하자.

먼저 C_1을 설계한다. 비동기 머신이 안정 상태 x_2로 진입할 때 제어기가 자신의 초기 상태 ξ_0에서 transition 상태 $\xi_0(x)$로 옮기는 동작은 아래와 같이 구현된다(식 (15), (16) 참조).

$$\begin{aligned}
&\Phi(\xi_0,(x,v)) = \xi_0 && \forall(x,v)\notin\{x_2\}\times\{b,c\}\\
&\Phi(\xi_0,(x2,v)) = \xi_0(x) && v\in\{b,c\}\\
&\eta(\xi_0,(x,v)) = v && \forall(x,v)\in X\times A
\end{aligned} \tag{23}$$

참고로 $U(x_2)$, 즉 x_2와 안정 조합을 이루는 입력 알파벳은 {b,c}이다(표 1 참조).

C_1이 $\xi_0(x)$에 있을 때 외부 입력 d가 들어오면 교정 동작을 시작한다. 가정에서 bad를 입력 스트링으로 사용한다고 하였다. 그런데 첫 번째 알파벳 b는 x_2와 안정 조합을 이루는 값이므로 실제 상태 천이를 일으키는 스트링은 t=ad로 간주해도 된다. 또 Σ는 t가 들어오면 x_2에서 x_3, 그리고 x_1로 천이한다. (입력 알파벳 a가 들어오면 x_2에서 x_3으로 천이된다고 설정한다. x_3 대신 x_4로 천이하는 경우는 제어기 C_2에서 다루어진다.) |t|=2이므로 제어기 C_1은 두 개의 중간 상태 ξ_1, ξ_2를 필요로 한다. 식 (19)~(21)에 따라서 C_1의 나머지 동작을 완성하면 다음과 같다.

$$\begin{aligned}
&\Phi(\xi_0(x),(x_2,v)) = \xi_0(x) \quad v\in\{b,c\}\\
&\Phi(\xi_0(x),(x_2,a)) = \xi_0\\
&\Phi(\xi_0(x),(x_2,d)) = \xi_1
\end{aligned} \tag{24}$$

$$\begin{aligned}
&\Phi(\xi_1,(x_2,d)) = \xi_2\\
&\eta(\xi_1,(x_2,d)) = a\\
&\Phi(\xi_2,(x_2,d)) = \xi_2\\
&\Phi(\xi_2,(x,v)) = \xi_0 \quad \forall(x,v)\neq(x_2,d)\\
&\eta(\xi_2,(x,v)) = d \qquad \forall(x,v)\in X\times A
\end{aligned} \tag{25}$$

입력 스트링 bac를 이용하는 제어기 C_2의 동작도 식 (23)~(25)와 유사하게 설계된다. 통합 제어기 C는 C_1과 C_2를 결합하여 $C=C_1\vee C_2$로 구해진다. 결합 연산 "∨"의 자세한 적용 과정은 Venkatraman과 Hammer의 기존 연구[5]를 참조하면 알 수 있다.

Ⅶ. 결 론

본 논문에서는 비동기 순차 머신의 안정 상태 동작을 바꾸는 새로운 제어 시스템을 제안하였다. 제어 대상 비동기 머신은 다음 안정 상태 값이 결정적이지 않는 불확실한 상태 천이를 가진다. 본 연구에서는 불확실한 상태 천이가 머신의 도달가능성에 미치는 영향을 표현하기 위해서 '결합된 결정적 입력 스트링'이라는 개념을 도입하였다. 도달가능성 행렬인 skeleton 행렬도 불확실한 상태 천이를 고려한 일반적인 형태로 확장되었다. 기존 연구와 유사하게 제어기의 존재조건을 skeleton 행렬의 부등식으로 표현하였으며, 제어기의 필요충분조건이 만족될 경우 폐루프 시스템과 정상적 모델과의 정합을 위한 제어기 설계 과정을 제시하고 입력 스트링

선택 규칙을 설정하였다. 또 예제를 통하여 도달가능성 행렬의 계산 과정과 제안된 피드백 제어기의 설계 과정을 각각 검증하였다.

참 고 문 헌

[1] P. J. Ramadge and W. M. Wonham, "The control of discrete event systems," Proceedings of IEEE, vol. 77, no. 1, pp. 81-98, 1989.

[2] 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.

[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] J. Hammer, "On corrective control of sequential machines," International Journal of Control, vol. 65, no. 2, pp. 249-276, 1996.

[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.

[6] 양정민, "입력 외란이 존재하는 비동기 순차 머신의 모델 매칭," 전기학회논문지, 제57A권 제1호, pp. 109-116, 2008.

[7] 양정민, "비결정 모델에 대한 비동기 순차 회로의 교정 제어 II: 제어기 설계," 전자공학회논문지 제45권 SC 제4호, pp. 11-20, 2008.

[8] L. Sterpone and M. Violante, "Analysis of the robustness of the TMR-architecture in SRAM-based FPGAs," IEEE Transactions on Nuclear Science, vol. 53, no. 5, pp. 1545-1549, 2005.

[9] L. Sterpone, M. Violante and S. Rezgui, "An analysis based on fault injection of hardening techniques for SRAM-based FPGAs," IEEE Transactions on Nuclear Science, vol. 53, no. 4, pp. 2054 - 2059.

[10] X. Geng, Model Matching for Asynchronous Sequential Machines, Ph.D. dissertation, Department of Electrical and Computer Engineering, University of Florida, 2003.

[11] Z. Kohavi, Switching and Finite Automata Theory (2nd ed.), McGraw-Hill, 1978.

[12] 양정민, "비결정 모델에 대한 비동기 순차 회로의 교정 제어 I: 도달가능성 분석," 전자공학회논문지 제45권 SC 제4호, pp. 1-10, 2008.

[13] M. Sipser, Introduction to the Theory of Computation (2nd ed.), Thomson Course Technology, 2006.

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~현재 대구가톨릭대학교 전자공학과 부교수

<주관심분야 : 비동기 머신 제어, 보행 로봇 시스템 등>