논문 2008-45SC-4-1

비결정 모델에 대한 비동기 순차 회로의 교정 제어 I: 도달가능성 분석

(Corrective Control of Asynchronous Sequential Machines for Nondeterministic Model I: Reachability Analysis)

양 정 민*

(Jung-Min Yang)

요 약

본 논문에서는 비동기 순차 머신의 교정 제어 문제를 다룬다. 교정 제어는 머신의 동작을 주어진 모델의 동작과 일치시키도록 하는 모델 매칭을 실현하는 제어를 말한다. 본 논문의 주요 목적은 비동기 순차 머신이 추종해야 하는 모델의 형태가 비결정적일 때, 즉 여러 개의 결정적 모델의 합으로 주어질 때 교정 제어기를 설계하는 일이다. 본 논문에서는 먼저 비동기 순차 머신을 위한 교정 제어 시스템의 형태와 동작 원리를 설명하고 비결정 모델의 표현 방법을 제안한다. 또한 교정 제어기 존재 조건을 표현하기 위해서 비동기 순차 머신과 비결정적 모델에 대한 도달가능성을 분석하고 예제를 통해서 분석 방법을 검증한다.

Abstract

The problem of controlling asynchronous sequential machines is addressed in this paper. Corrective control means to make behavior of an asynchronous sequential machine equal to that of a given model. The main objective is to develope a corrective controller, especially when a model is given as nondeterministic, or a set of reference models. The structure of corrective control system for asynchronous sequential machines is addressed first, followed by description of nondeterministic models. Then, we propose a method for analyzing reachability of asynchronous machines and nondeterministic models. Proposed methods are demonstrated in an example.

Keywords : Asynchronous Sequential Machines, Corrective Control, Model Matching, Nondeterministic Model

I. 서 론

본 논문에서는 비동기 순차 머신(Asynchronous Sequential Machine)의 제어를 다룬다. 전역 클럭(clock) 없이 동작하는 비동기 순차 머신은 동기 순차 머신(Synchronous Sequential Machine)에 비해서 전력 소비가, 적고 연산 속도가 뛰어나며, 언제나 평균적 성능(average-case performance)을 보인다는 등의 장점을 지닌다. 비동기 순차 머신에 대한 연구는 이러한 장점을 살리면서 효율적으로 비동기 회로를 설계하는 방법에 대한 연구에 집중되었다[1].

이번 연구에서는 비동기 순차 머신을 설계 대상으로 보지 않고 제어 대상으로 보는 새로운 관점에서 문제를 해결한다. 즉 전통적인 자동 제어(Automatic Control)의 원리를 적용하여 기존의 비동기 순차 머신의 동작을 원하는 목적에 맞도록 바꾸어주는 제어기를 설계하는 방법을 취한다. 이 제어기는 외부로부터 입력 신호, 제어 대상 비동기 순차 머신으로부터 출력 피드백(output feedback)을 각각 받아서 제어 입력을 만든 후 비동기 순차 머신의 입력부에 보낸다. 자동 제어의 피드백 제어기와 마찬가지로 만약 비동기 순차 머신이 원하는 동작을 보인다면 제어기는 외부 입력을 그대로

* 정회원, 대구가톨릭대학교 전자공학과
(Department of Electrical Engineering, Catholic University of Daegu)
접수일자: 2007년10월25일, 수정완료일: 2008년7월8일

비동기 순차 머신에 전달하는 역할만을 수행한다. 비동기 순차 머신이 원하지 않는 동작을 보이기 시작하면 제어기는 적절한 제어 입력을 만들어 낸다. 제어기는 비동기 순차 머신과 연동해야 하므로 역시 클럭 없는 비동기 순차 머신의 형태로 구성되어야 한다. 교정 제어(Corrective Control)라 불리는 이러한 개념은 Hammer[2]가 일반적인 순차 머신의 제어를 위해서 맨 처음 도입하였으며, 최근 레이스(race)가 존재하는 비동기 순차 머신[3], 입력/출력형 비동기 머신의 제어[4]에 성공적으로 적용되었다.

본 논문의 주요 목적은 비동기 순차 머신이 추종해야 하는 모델이 비결정적으로 주어질 때 제어기를 설계하는 문제를 해결하는 일이다. 교정 제어의 기본 목표는 시스템(비동기 순차 머신)이 원하는 동작, 즉 어떤 모델의 동작과 동일하게 움직이도록 만드는 것으로, 자동 제어의 모델 추종 제어(Model Following Control)와 유사하다. 기존 연구[3~4]에서는 이러한 모델이 모두 결정적 유한 상태 머신(Deterministic Finite State Machine)으로 주어졌다. 하지만 비동기 순차 머신이 모사(模寫) 해야 하는 모델이 경우에 따라서 복잡하게 표현될 수 있으며, 특히 결정적 유한 상태 머신으로 주어진 현재의 모델에서 시스템이 추종해야 하는 동작을 추가해야 할 필요가 생길 수도 있다[5].

먼저 비동기 순차 머신의 교정 제어기의 형태와 동작 과정이 기술된다. 또한 비결정적 모델을 여러 개의 결정적 유한 상태 머신의 합으로 표현하는 방법을 제안하고, 이러한 비결정적 모델과 비동기 순차 회로의 동작이 일치하도록 만드는 제어기의 설계를 위한 이론적 방법을 제안한다. 기존 연구[3~4]에서와 마찬가지로 제어기가 존재할 조건은 "skeleton 행렬"이라는 도달가능성 행렬의 부등식으로 표현할 수 있다. 본 논문에서는 비동기 순차 머신과 비결정적 모델의 도달가능성 행렬을 만드는 과정을 기술하고 예제를 통해서 행렬 연산을 검증한다. 교정 제어기의 구체적인 존재조건과 설계 과정은 후속 논문[6]에서 기술한다.

II. 비동기 순차 머신 모델링

일반적으로 사용되는 비동기 순차 머신의 논리적 모델은 다음과 같은 유한 상태 머신이다.

$$\Sigma = (A, Y, X, x_0, f, h) \tag{1}$$

위 식에서 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)를 가리키며 다음과 같이 정의된다.

$$\begin{aligned} x_{k+1} &= f(x_k, u_k), \\ y_k &= h(x_k, u_k),\ k = 0,1,2,\ldots \end{aligned} \tag{2}$$

$u_0, u_1, u_2, \ldots$는 머신의 입력 시퀀스(sequence)이며 $u_0, u_1, u_2, \ldots$와 $y_0, y_1, y_2, \ldots$는 각각 머신의 상태와 출력 시퀀스를 가리킨다. 머신의 스텝(step) k는 입력이나 상태 변수가 변경되었을 때마다 1씩 증가한다.

본 논문에서는 비동기 순차 머신의 현재 상태가 출력값으로 나오는 입력/상태 머신(input/state machine)에서 대해서 다루기로 한다. 즉 출력 알파벳 집합 Y는 X와 일치하며 출력 함수도 아래와 같이 간단하게 표시된다.

$$y_k = x_k,\ k = 0,1,2,\ldots \tag{3}$$

'유효 조합(valid pair)' $(x,u) \in (X\times A)$는 유한 상태 머신에서 f와 h가 각각 정의된 입력과 상태 쌍을 말한다. (즉 f와 h가 정의되는 않는 입력과 상태 쌍도 존재한다.) 유효 조합 (x,u)는 $x=f(x,u)$일 때 '안정 조합(stable combination)'이라고 정의된다. 이때 x는 '안정 상태(stable state)'라 불리며 함수 f의 고정점(fixed point)이 된다. 비동기 순차 머신은 입력이 바뀌지 않는 한 안정 조합에서 계속 머물러 있게 된다. 안정 조합이 아닌 입력과 상태 쌍을 '과도 조합(transient combination 또는 transient pair)'이라고 부르고 과도 조합을 이루는 상태를 '불안정 상태(unstable state)'라고 정의한다. '잠재적 안정 상태(potentially stable state)'는 안정 조합을 이루는 입력 알파벳을 적어도 하나 이상 가지고 있는 상태를 말한다.

비동기 순차 머신에서 과도 조합 (x,u)는 입력이 바뀌지 않는 한 $x_1 = f(x,u)$, $x_2 = f(x_1,u), \ldots$와 같이 연쇄적으로 상태 천이를 시작하여 안정 조합에 도달한다. 즉 $q \geq 1$인 자연수 q가 존재하여 $x' = f(x_q, u)$이고 $x' = f(x', u)$인 상태 x'가 존재한다면 (x',u)는 비동기 순차 머신이 (x,u)에서 시작하여 도달하는 안정 조합이 된다. 만약 연쇄 천이가 끝나지 않고 과도 조합 사이에서 계속 이루어진다면 (x,u)는 무한 순환(infinite cycle)의 한 부분이 된다. 본 논문에서는 비동기 순차

머신에서 무한 순환이 존재하지 않는다고 가정한다.

클럭 없이 움직이는 비동기 순차 머신에서 발생될 수 있는 모호한 상태를 없애기 위해서 비동기 순차 머신은 일반적으로 '기본 모드(fundamental mode)'로 동작한다고 가정한다. 기본 모드는 비동기 순차 머신이 한 스텝 움직일 때 입력, 출력, 상태 변수 중 한 개의 변수만이 변경 가능하다는 의미이다. 즉 머신의 입력은 비동기 순차 머신이 안정 조합에 도달한 후에만 바뀔 수 있다. 만약 비동기 순차 머신이 과도 조합에 있을 때 입력이 바뀐다면 (클럭이 없기 때문에) 머신이 현재 어떤 상태에 있는지 알기 모호해지므로 상태 천이 함수를 적용하기가 불가능해진다. 최근 여러 개의 입력 변화가 동시에 허용되는 burst 모드[7]에서 비동기 순차 머신을 설계하는 기법들도 연구되고 있으나 본 논문에서 다루는 교정 제어에서 사용하기에는 불가능하다. 따라서 본 논문에서는 비동기 순차 머신이 기본 모드에서 동작한다고 가정한다.

비동기 순차 머신 Σ의 'stable recursion' 함수 s는 모든 유효 조합에서 정의되는 함수로서 유효 조합에서 시작하여 비동기 순차 머신이 다다르는 다음 안정 상태(next stable state)를 그 함수값으로 가진다. 즉

$$s : X \times A \rightarrow X \tag{4}$$

이고 유효 조합 (x,u)에서 $s(x,u) := x'$이며 x'는 (x,u)의 다음 안정 상태이다. 정의에 따라서 (x',u)는 안정 조합이 된다. f 대신 s를 이용하여 Σ의 'stable-state 머신'을

$$\Sigma_{|s} = (A, Y, X, x_0, s, h) \tag{5}$$

이라고 정의하고 $\Sigma_{|s}$라고 명명한다. stable-state 머신 $\Sigma_{|s}$는 Σ에서 상태 천이 함수 f 대신 stable recursion 함수 s를 사용하여 정의된 유한 상태 머신이다. $\Sigma_{|s}$는 비동기 순차 머신이 실제 외부 사용자에게 보이는 모습을 나타낸다고 말할 수 있다. 클럭이 없으므로 비동기 순차 머신이 불안정 상태에서 머무르는 시간은 이론적으로 0이며, 머신은 순식간에 다음 상태로 천이된다. 따라서 외부에서 보면 비동기 순차 머신의 상태가 고정되는 안정 조합 동작만이 유효하다.

기본 모드에서는 한 번에 한 개의 변수만이 바뀔 수 있으므로 비동기 순차 머신의 입력은 머신이 안정 조합에 있을 때에만 그 값이 변경 가능하다고 하였다. 머신이 초기 상태 x_0에 있고 입력 시퀀스 $u = u_0 u_1 \cdots u_{m-1} \in A^+$가 들어온다고 가정하자. 이 때 A^+는 A의 알파벳으로 만들어지는 스트링(string) 중 빈 스트링 (empty string) ε를 제외한 집합을 말한다. 기본 모드 동작에서 첫 번째 입력 알파벳 u_0은 머신이 (x_0, u_0)의 다음 안정 상태, 즉 $x_1 := s(x_0, u_0)$에 도달할 때까지 그 값을 계속 유지한다. Σ가 상태 x_1에 도달하면 입력은 u_0에서 u_1로 바뀌며, Σ는 다시 다음 안정 상태 $x_2 := s(s(x_0,u_0),u_1)$에 도달할 때까지 입력을 u_1로 유지한다. 이런 식으로 비동기 순차 머신은 마지막 안정 상태

$$x_m := s(\ldots s(s(s(x_0,u_0),u_1),u_2)\ldots,u_{m-1}) \tag{6}$$

에 도달한다. 입력 시퀀스가 주어지면 s 함수는 일반적으로 다음과 같이 간편하게 표기된다.

$$s(x_0, u) := (\ldots s(s(s(x_0,u_0),u_1),u_2)\ldots,u_{m-1}),\quad u \in A^+ \tag{7}$$

기본 모드로 동작하는 비동기 순차 머신은 머신이 안정 조합에 있을 때에만 제어값을 변경할 수 있다. 또한 제어기를 설계하려면 비동기 순차 머신이 stable-state 머신으로 표현되었을 때 머신의 상태들이 서로 도달가능한지를 아는 일이 매우 중요하다. 본 논문에서는 기존 연구[3~4]에서 정의된 안정적 도달가능성(stable reachability)를 도입한다.

정의 1. x와 x'는 비동기 순차 머신의 잠재 안정 상태이고 s는 머신의 stable recursion 함수이다. $x' = s(x,u)$이 되는 입력 시퀀스 $u = u_0 u_1 \cdots u_k \in A^+$가 존재한다면 x'는 x에서 '안정적으로 도달가능(stably reachable)'하다. □

본 장에서 도입한 정의와 표기법의 자세한 내용은 기존 연구[3~4]와 Kohavi[8]에서 찾을 수 있다.

III. 교정 제어

1. 폐루프 시스템

Murphy[3] 등이 제안한 비동기 순차 머신을 위한 교정 제어 시스템을 그림 1에 나타내었다. 비동기 순차 머신 Σ 앞에 상태 피드백 교정 제어기 C가 연결되어 폐루프 시스템이 원하는 동작을 하도록 제어 입력을 만

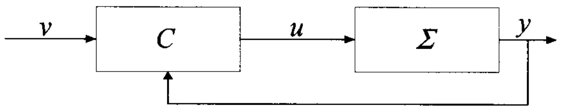

그림 1. 비동기 순차 머신을 위한 교정 제어 시스템
Fig. 1. Corrective control system for asynchronous sequential machines.

든다. v는 외부로부터 들어오는 입력 신호이고 u는 제어 입력, y는 비동기 순차 머신의 출력이다. 본 논문에서는 현재 상태가 출력으로 나오는 머신을 다루므로 y는 머신의 현재 상태 x의 값을 가진다.

교정 제어기는 외부 입력 v, Σ의 상태 피드백 $y(=x)$를 입력으로 가지므로 C의 입력 집합은 $A\times X$이다 (식 (1) 참조). 또한 C가 만드는 제어 입력 u는 머신 Σ의 입력으로 들어가므로 C의 출력 집합은 A이다. C를 유한 상태 머신으로 표현하면 아래와 같다.

$$C = (A\times X, A, \Xi, \xi_0, \phi, \eta) \tag{8}$$

위 식에서 Ξ은 C의 상태 집합이며 ξ_0는 초기 상태, ϕ는 recursion 함수, η는 출력 함수이다. 그림 1에서 구현되는 폐루프 시스템을 Σ_c라고 표기하면 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.

$$\Sigma_c = (A, X, X\times\Xi, (x_0,\xi_0), f_c, h_c) \tag{9}$$

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

2. 모델 매칭 문제

교정 제어의 목적은 그림 1의 폐루프 시스템 Σ_c이 원하는 동작을 보이도록, 즉 어떤 모델 Σ'의 동작과 일치하도록 하는 모델 매칭 문제를 푸는 일이다. 좀 더 자세히 말하면 모델 매칭 문제는 어떤 비동기 순차 머신의 stable-state 머신이 주어진 모델의 동작과 일치하도록 제어기를 설계하는 일이다[3]. 즉 모델 매칭이라는 의미는 두 머신의 동작이 안정 상태에서 일치해야 한다는 뜻이다. 두 머신의 동작을 불안정 상태에서까지 일치시키도록 하는 일은 머신을 물리적으로 재설계하지 않는 이상 불가능하다. 또한 앞에서 설명했듯이 외부 사용자에게는 비동기 순차 머신의 동작 과정 중 안정 상태에서의 모습만이 관찰되므로 이러한 안정 상태에서의 모델 매칭 문제 설정은 타당하다 하겠다.

모델 매칭 문제를 설정하기 전에 먼저 비동기 순차 머신의 stable equivalence[8]를 다음과 같이 정의한다.

정의 2. 두 개의 비동기 순차 머신 Σ와 Σ'가 다음과 같이 정의된다고 하자.

$$\Sigma = (A, Y, X, x_0, f, h) \tag{10}$$
$$\Sigma' = (A, Y, X, \zeta_0, f', h')$$

Σ와 Σ'은 동일한 입력과 상태 집합 A와 X를 가진다. $\Sigma_{|s}$와 $\Sigma'_{|s}$를 각각 Σ와 Σ'의 stable-state 머신이라고 표기한다. Σ의 상태 $x\in X$와 Σ'의 상태 $x'\in X$ 는 다음과 같은 조건을 만족할 때 서로 'stably equivalent($x\equiv x'$)'하다고 정의한다: $\Sigma_{|s}$는 상태 x에 있고 $\Sigma'_{|s}$는 상태 x'에 있을 때 i) $\Sigma_{|s}$와 $\Sigma'_{|s}$는 동일한 허용 입력 시퀀스를 가지고, ii) 모든 허용 입력 시퀀스에 대해서 $\Sigma_{|s}$와 $\Sigma'_{|s}$가 동일한 출력 시퀀스를 낸다. 또 두 머신의 초기 상태가 서로 stably equivalent하다면, 즉 $x_0\equiv\zeta_0$이면 비동기 순차 머신 Σ와 Σ'는 서로 'stably equivalent'하다고 정의하고 $\Sigma=\Sigma'$로 표기한다. □

stable equivalence는 두 개의 비동기 순차 머신의 동작이 외부 사용자에게 동일하게 보인다는 뜻이다. Σ와 Σ'가 stable equivalent 하고 두 머신이 모두 입력/상태 머신이라면 임의의 현재 입력에 대해서 두 머신은 항상 동일한 상태로 천이되어야 한다. stable equivalence 개념을 이용하여 모델 매칭 문제를 정립하면 다음과 같다.

정의 3. 입력/상태 머신의 모델 매칭 문제: 두 개의 입력/상태 비동기 순차 머신 $\Sigma=(A,X,X,x_0,f,h)$와 $\Sigma'=(A,X,X,x_0,f',h')$가 동일한 입력 집합 A와 상태 집합 X를 공유한다고 하고 $\Sigma_{|s}$와 $\Sigma'_{|s}$를 각각 Σ와 Σ'의 stable-state 머신이라고 표기한다. 그림 1의 폐루프 시스템 $\Sigma_{c|s}$이 $\Sigma'_{|s}$와 stably equivalent하는 동작을 보이도록 만드는 교정 제어기 C가 존재할 필요충분조건을 구하고, C가 존재한다면 설계 알고리즘을 찾는다. □

모델 매칭 문제는 stable-state 머신에서 구현되어야

하므로 비동기 순차 머신이 추종해야 하는 모델 Σ'가 항상 stable-state 머신으로 표현된다고 설정해도 일반성을 잃지 않는다. 따라서 본 논문에서는 Σ'가 stable-state 머신이라고 가정한다.

Ⅳ. 비결정 모델

1. 비결정 모델 정의

이번 장에서는 본 논문의 주요 목적인 비결정 모델을 정의하고 그 물리적 의미를 기술한다.

일반적으로 비결정 모델은 비결정적 유한 상태 머신 (Nondeterministic Finite State Machine: NFSM)으로 표현된다. NFSM에서 상태 천이 함수 f는 다음과 같이 정의된다.

$$f: X \times A \rightarrow P(X) \tag{11}$$

위 식에서 $P(X)$는 X의 모든 부분 집합을 원소로 가지는 멱집합(power set)이다. 즉 NFSM에서는 현재 상태와 현재 입력이 주어질 때 f의 값이 상태 한 개로 결정되지 않고 X의 부분 집합으로 주어진다. NFSM은 비동기 순차 머신 내부에서 발생할 수 있으나 외부로는 노출되지 않는 입력에 의한 영향, 외란이나 다른 요인들로 인해서 생기는 상태 천이 등의 물리적 현상, 또는 다수의 결정적 유한 상태 머신(Deterministic Finite State Machine: DFSM)의 행동을 결합한 행동 등을 모델링 한다고 말할 수 있다[9].

본 논문에서는 비동기 순차 머신의 모델 매칭 문제를 다루기 위해서 NFSM을 여러 개의 DFSM의 합으로 표현하는 방법을 도입하기로 한다. NFSM을 표현하는 방법으로는 비결정적 상태 천이 함수를 도입하는 기법[10], 널리 알려진 대로 NFSM을 한 개의 DFSM으로 치환하는 방법[9], 그리고 본 논문에서처럼 NFSM을 여러 개의 DFSM의 합으로 표현하는 방법[11] 등이 제안되었다. 이 중 비결정적 상태 천이 함수를 도입하는 방법을 쓰면 기존의 비동기 교정 제어에서 사용된 도달가능성 행렬로는 머신의 도달가능성을 표현할 수 없게 되므로 적용하기가 곤란하다. 또 NFSM에서 DFSM로 변환하는 방법은 변환된 DFSM가 원래의 상태 집합과는 다른 상태 집합을 가지게 되므로 정의 3에서 설정한 입력/상태 비동기 순차 머신의 모델 매칭 문제에 적용하기가 불가능하다. 모델 Σ'의 상태 집합은 비동기 순차 머신 Σ의 상태 집합과 반드시 일치해야 모델 매칭 문제를 풀 수 있다. 본 논문에서 도입되는 비결정 모델 표현 방법은 아래와 같다.

$$\Sigma' = \Sigma_1' \parallel \Sigma_2' \parallel \cdots \parallel \Sigma_p' \tag{12}$$

Σ'는 NFSM으로 주어지는 비결정 모델이며 Σ_i'는 Σ'로부터 분해된 DFSM이다. 위 식에서 표기된 '$\parallel$' 기호는 일반적인 병렬 합성(parallel composition)[9]을 의미하지 않으며 Σ'의 상태 천이 함수로부터 DFSM 부(sub) 모델을 여러 개 만든다는 뜻이다.

식 (12)의 의미를 부연 설명하기 위해서 그림 2(a)와 같은 비결정 모델을 가정하자. 그림에서 알 수 있듯이 Σ'는 유효 조합 (x_1, a)에서 비결정성을 가지며 $s'(x_1, a) = \{x_2, x_3\}$이다. 본 논문에서는 이러한 비결정 모델을 결정적 모델 두 개로 분해해서 고려하는 방

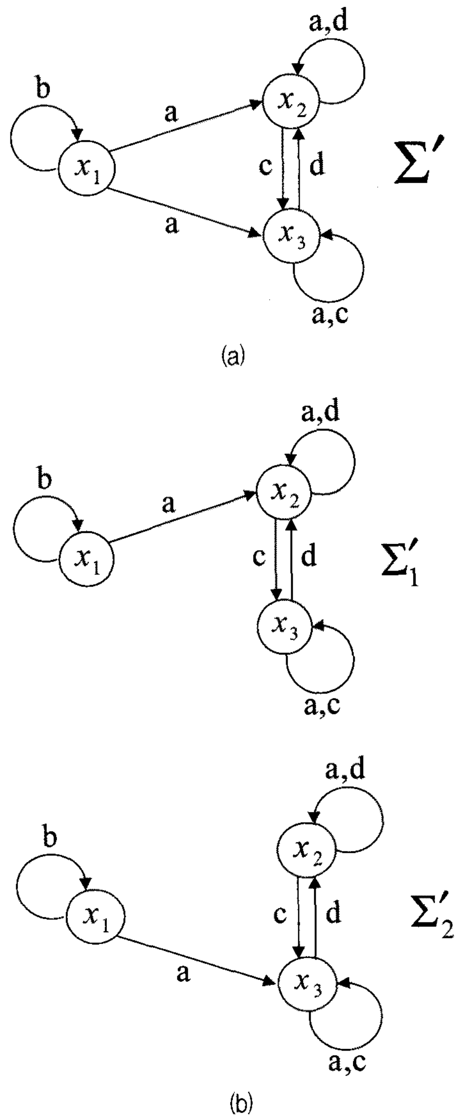

그림 2. 비결정 모델의 분해

Fig. 2. Decomposition of nondeterministic model.

법을 쓴다. 즉 Σ'는 그림 2(b)의 부 모델 Σ'_1과 Σ'_2 두 개로 나뉠 수 있다. 그림 2는 결정적 모델 Σ'_1과 Σ'_2가 먼저 주어지고 난 후 비결정 모델 Σ'을 합성하는 과정으로도 해석 가능하다. 즉 여러 개의 결정적 모델을 먼저 만든 후 이 모델들의 동작을 모두 가져야 하는 비결정 모델을 최종적으로 구현하는 식이다. 후자의 해석이 더 현실적으로 적용되는 문제도 많이 존재한다.

식 (1)과 유사하게 각 DFSM Σ'_i를 표현하면 다음과 같다.

$$\Sigma'_i = (A, X, X, x_0, s'_i, h), \ \ i = 1, \cdots, p \tag{13}$$

앞 장에서 stable-state 머신으로 모델을 표현한다고 설정하였으므로 상태 천이 함수 f 대신 stable recursion 함수 s'_i를 사용하였다. Σ'_i, i=1,...,p 가 주어질 때 비결정 모델 Σ'는 다음과 같이 합성된다.

$$\begin{aligned} &\Sigma' = (A, X, X, x_0, s', h) \\ &s'(x,a) := \bigcup_{i=1}^{p} s'_i(x,a), \ x \in X, \ a \in A \end{aligned} \tag{14}$$

식 (12)과 (14)에서 정의한 합성(또는 분해) 방법은 기존의 병렬 합성 방법[9]과 유사하나 유한 상태 머신들의 입력 집합과 상태 집합이 모두 동일해야 한다는 조건이 있다. 식 (14)의 모델링 방법은 레이스가 존재하는 비동기 순차 머신을 여러 개의 순차 머신 합으로 표시한 기존 연구[3]의 접근 방법을 일반화시켰다고 해석될 수도 있다.

식 (14)과 같은 방법으로 비결정 모델을 정의하면 모델의 도달가능성(reachability)이 기존 연구[3~4]에서 제안된 행렬식으로 간단하게 표현되는 장점을 얻는다. 따라서 모델 매칭을 위한 교정 제어기 설계도 용이해진다. 또한 이 방법은 비동기 순차 머신이 추종해야 하는 모델의 동작을 다시 설정해야 할 때 문제 해결을 더 쉽게 만든다. 예를 들어 모델이 Σ'에서 $\Sigma''=\Sigma'+\Delta\Sigma'$로 바뀐다면 추가된 동작 $\Delta\Sigma'$을 DFSM으로 구현하고 식 (14)에 의해서 합성하면 새로운 모델 Σ''을 얻는다.

식 (14)에서 제안된 모델의 합성/분해 방법의 약점은 p, 즉 NFSM으로부터 분해되는 DFSM의 개수가 NFSM의 상태 집합의 크기에 따라서 기하급수적으로 늘어날 가능성이 있다는 사실이다. 하지만 NFSM을 단일 DFSM으로 치환하는 기존 방법[9]도 똑같은 성질을 가진다. 또한 다음 장에서 나오듯이 단일 DFSM로 표시된 비동기 순차 머신의 도달가능성을 나타내는 대수적 행렬은 산술급수적인 계산 복잡도(Computational Complexity)를 가지므로 식 (14)의 비결정 모델을 위한 도달가능성을 계산하는 경우의 복잡도는 크게 늘어나지 않는다.

2. 비결정 모델과 모델 매칭 문제

비동기 순차 머신이 추종해야 할 모델을 비결정적으로 확장하면 정의 3의 기본적인 모델 매칭 문제도 여러 가지로 확장할 수 있다. 본 논문에서는 Di Benedetto[5] 등이 일반적인 유한 상태 머신 기반 하에서 제안한 방법과 유사하게 다음과 같은 두 가지 문제를 설정한다.

가. 온라인 모델 매칭 문제

먼저 가장 제한적인 모델 매칭 문제를 생각하면 제어되는 비동기 순차 머신의 교정 동작이 비결정 모델의 가능한 모든 동작을 가지도록 목적을 설정할 수 있다. 예를 들어 그림 2(a)의 모델 Σ'은 상태 x_1에서 외부 입력 a가 들어오면 x_2(부 모델 Σ'_1) 또는 x_3(부 모델 Σ'_2)으로 천이된다. 제한적인 모델 매칭 문제를 위한 교정 제어기 C의 목적은 폐루프 시스템 Σ_{cls}의 동작을 항상 모델 Σ'와 일치시키는 일이다. 즉 Σ'가 (x_1, a)에서 Σ'_1의 동작을 보이든 Σ'_2의 동작을 보이든 상관없이 Σ_{cls}을 Σ'과 stable equivalent되도록 만들어야 한다. 이 문제는 모델 Σ'가 비동기적으로 움직인다는 설정 외에는 다음과 같이 정의 3과 동일하게 기술된다.

정의 4. 비결정 모델을 가진 입력/상태 머신의 온라인 모델 매칭 문제:

두 개의 입력/상태 비동기 순차 머신 $\Sigma = (A, X, X, x_0, f, h)$와 $\Sigma' = (A, X, X, x_0, s', h')$가 동일한 입력 집합 A와 상태 집합 X를 공유한다고 하고

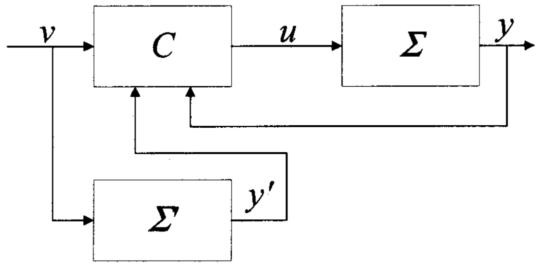

그림 3. 비결정 모델에 대한 교정 제어 시스템

Fig. 3. Corrective control system with nondeterministic model.

stable state 모델 Σ'은 비결정적이라고 하자. 그림 3의 폐루프 시스템 $\Sigma_{c|s}$이 Σ'와 stably equivalent하는 동작을 보이도록 만드는 교정 제어기 C가 존재할 필요충분조건을 구하고, C가 존재한다면 설계 알고리즘을 찾는다. □

결정적 모델에 대한 교정 제어 문제는 모델의 동작이 임의의 상태에서 항상 한 가지로 정해지므로 그림 1에서 볼 수 있듯이 모델을 폐루프 시스템에 넣을 필요가 없다. 하지만 비결정 모델에 대한 제한적인 모델 매칭 문제를 풀기 위해서는 비동기 순차 머신의 현재 상태에서 모델이 어떠한 동작을 보이는가를 확인하여 교정 제어기에 알려줄 필요가 있다. 따라서 전통적인 자동 제어에서와 마찬가지로 폐루프 제어 시스템이 모델을 포함해야 한다. 그림 3은 정의 4에서 제안된 문제를 위한 비동기 순차 머신의 교정 제어 시스템이다. 그림 3에서 y'는 모델 Σ'의 출력이다. 교정 제어기 C는 y'의 값을 보고 모델이 천이되는 다음 상태를 먼저 관측한다. 그런 다음 외부 입력 v와 머신 Σ의 상태 피드백 y를 받아서 제어 입력 u를 만든다. 교정 제어기가 모델의 다음 상태를 확인한 다음 제어 입력을 넣기 때문에 폐루프 시스템 $\Sigma_{c|s}$은 모델 Σ'의 동작과 시간차를 두고 생성된다. 그림 3에서 다루는 모든 머신들이 비동기적이므로 모델과 폐루프 시스템의 동작이 정확하게 동기화될 필요는 없다.

나. 오프라인 모델 매칭 문제

다음으로는 제한적 모델 매칭 문제와 반대되는 경우를 생각할 수 있다. 즉 비동기 순차 머신은 비결정 모델 Σ'이 가질 수 있는 결정적 모델 중 적어도 하나와 동작이 일치되도록 제어기를 꾸미는 일이다. 물론 머신 Σ의 동작을 모델 Σ'의 가능한 모든 동작과 일치시킬 수 있다면 정의 4에서 설정한 문제도 해결된다. 하지만 그렇지 못할 경우 모델 매칭을 위한 차선책을 구해야 하며, 오프라인 제어를 통해서 모델 Σ' 안에서 생성될 수 있는 결정적 유한 상태 머신 중 비동기 순차 머신 Σ과 매칭될 수 있는 것을 찾는다.

정의 5. 비결정 모델을 가진 입력/상태 머신의 오프라인 모델 매칭 문제: 입력/상태 비동기 순차 머신 $\Sigma=(A,X,X,x_0,f,h)$와 $\Sigma'=(A,X,X,x_0,s',h')$가 동일한 입력 집합 A와 상태 집합 X를 공유한다고 하고 stable-state 머신 모델 Σ'은 $\Sigma'=\Sigma_1' \parallel \Sigma_2' \parallel \cdots \parallel \Sigma_p'$와 같이 분해 가능한 비결정 모델이라고 하자. 그림 1의 폐루프 시스템 $\Sigma_{c|s}$이 $\Sigma_i, i=1,\ldots,p$ 중 적어도 한 개와 stably equivalent하는 동작을 보이도록 만드는 교정 제어기 C가 존재할 필요충분조건을 구하고, C가 존재한다면 설계 알고리즘을 찾는다. □

위에서 정의된 모델 매칭 문제에서 교정 제어기는 모델 Σ'의 행동을 실시간으로 관측할 필요가 없으므로 그림 1의 형태를 지닌다.

V. 도달가능성

비동기 순차 머신의 도달가능성이란 비동기 순차 머신의 상태들 사이에서 서로 천이할 수 있는 경로가 존재하는지 여부를 말한다. 후속 논문[6]에서 기술될 교정 제어기의 존재 여부는 비동기 순차 머신 Σ와 모델 Σ'의 도달가능성 사이의 관계에 달려 있으므로 도달가능성을 정량적으로 표현하는 일은 모델 매칭 문제 해결 과정에서 반드시 필요하다. 본 장에서는 기존 연구[3~4]에서 도입된 skeleton 행렬을 소개하고 앞 장에서 정의된 비결정 모델을 위한 skeleton 행렬 계산 과정을 제시한다. 비동기 순차 머신의 모델 매칭 문제는 항상 stable-state 머신에서 고려되므로 skeleton 행렬의 계산도 stable-state 머신에서 시작되어야 한다.

정의 6. skeleton 행렬: 입력/상태 비동기 순차 머신 Σ의 stable-state 머신을 $\Sigma_{|s}=(A,X,X,x_0,s,h)$이라고 하고 상태 집합 X의 원소를 $X=\{x_1,\ldots,x_n\}$라고 표시하자. Σ의 skeleton 행렬 $K(\Sigma)$은 아래와 같이 정의되는 $n\times n$차 정방 행렬이다.

$$K_{ij}(\Sigma)=\begin{cases}1 & \exists u\in A^+ \ s.t.\ x_j=s(x_i,u)\\ 0 & \text{otherwise}\end{cases} \tag{15}$$

□

$K_{ij}(\Sigma)=1$이면 $\Sigma_{|s}$의 상태를 x_i에서 x_j로 천이시키는 입력 스트링이 적어도 하나 존재한다. 입력 스트링에 대한 상태 천이 함수 s의 정의는 식 (14)에 나와 있다. II장에서 비동기 순차 머신이 무한 순환을 가지지 않는다고 규정했기 때문에 $K_{ij(\Sigma)=1}$을 만족시키는 입력 스트링 u의 길이는 항상 n-1 이하로 줄일 수 있다[3]. skeleton 행렬을 구하는 자세한 과정은 뒤에 나오는 예

제에서 기술한다.

skeleton 행렬은 비동기 순차 머신의 안정 상태에서의 도달가능성을 간명하게 나타내기 때문에 교정 제어기 존재 조건을 표시하고 제어기를 설계하는 과정에서 유용하게 사용된다. 그러나 식 (15)은 결정적 모델에 대한 도달가능성만 표현해줄 뿐 비결정적 유한 상태 머신에 대해서는 적용할 수 없다. 본 논문에서는 식 (14)에서 제안된 모델의 분해 방법을 이용하여 비결정 모델 및 정의 4, 5의 온라인/오프라인 모델 매칭 문제에 대한 skeleton 행렬 계산 방법을 제안한다. 먼저 비결정 모델 Σ'이 $\Sigma' = \Sigma_1' \parallel \Sigma_2' \parallel \cdots \parallel \Sigma_p'$와 같이 분해된다고 설정하자. 정의 6에 나와 있는 skeleton 행렬은 DFSM로 표시된 비동기 순차 머신에 적용 가능하므로 각 부-모델 $\Sigma'_i, i = 1,\ldots,p$의 skeleton 행렬 $K(\Sigma'_i)$은 쉽게 계산된다. $K(\Sigma'_i)$을 계산한 다음 정의 4와 5에서 제안된 각 모델 매칭 문제에 맞는 모델의 skeleton 행렬을 계산한다.

정의 4에서 제안된 문제는 폐루프 시스템 $\Sigma_{c|s}$의 동작이 모델 Σ'의 가능한 모든 동작과 일치하도록 해주는 것이므로 도달가능성 행렬도 모델 Σ'의 동작 모두를 표시해주어야 한다. 즉 p개의 부-모델 $\Sigma'_1,\ldots,\Sigma'_p$에서 어떤 두 상태 사이의 상태 천이가 한 번이라도 정의되어 있다면 skeleton 행렬 원소의 값은 1이 되어야 한다. 이 과정을 일반적으로 표현하면 비결정 모델 Σ'의 skeleton 행렬 $K(\Sigma')$은 아래와 같이 정의된다.

$$K_{ij}(\Sigma') = \max_{k=1,\ldots p} K_{ij}(\Sigma'_k), i = 1,\ldots,n \quad (16)$$

skeleton 행렬 원소는 0 또는 1의 값만 가지므로 식 (16)과 같이 행렬식을 계산하면 모델 Σ'의 모든 동작(또는 도달가능성)을 표현할 수 있다. 즉 (16)에서 제안된 skeleton 행렬은 기존 연구[3~4]에서 도입된 skeleton 행렬을 비결정 모델에 대해서 일반화한 결과라고 말할 수 있다. 즉 기존 연구에서는 모델 Σ'이 DFSM으로 표현되므로 식 (16)에서 p=1로 잡으면 $K(\Sigma')$는 기존 연구에서 제안된 skeleton 행렬로 귀결된다.

다음으로는 정의 5에서 제시된 오프라인 모델 매칭 문제를 위한 모델의 skeleton 행렬을 알아본다. 정의 5에서 나와 있듯이 비동기 순차 머신은 비결정 모델 Σ'이 만드는 동작 중 적어도 하나의 결정적 유한 상태 머신만 모사하면 된다. 따라서 즉 p개의 부-모델 $\Sigma'_1,\ldots,\Sigma'_p$에 대한 skeleton 행렬을 모두 구한 후 제어 대상 비동기 순차 머신 Σ의 skeleton 행렬 $K(\Sigma)$과 개별적 비교만 하면 되므로 다른 계산 과정은 필요하지 않다.

VI. 예 제

본 장에서는 비결정 모델의 skeleton 행렬 계산 과정을 예제를 통해서 설명한다. 비결정 모델 Σ'이 그림 2(a)와 같이 주어진다고 하자. 이 모델은 그림 2(b)와 같이 두 개의 DFSM Σ'_1과 Σ'_2로 분해된다. 그림에서 알 수 있듯이 Σ'_1과 Σ'_2는 모두 stable-state 머신이다. skeleton 행렬을 얻기 위해서는 먼저 각 머신의 stable transition 행렬[3]을 계산해야 한다. stable transition 행렬은 두 상태 사이에 존재하는 가능한 모든 입력 스트링을 원소로 가진다. 또 상태 집합 X의 원소가 세 개이므로 길이가 1 또는 2인 입력 스트링만 취하면 된다[3]. 길이가 m보다 적거나 같은 입력 스트링을 원소로 가지는 stable transition 행렬을 $R^{(m)}(\Sigma)$이라고 정의하면 $R^{(2)}(\Sigma'_1)$, $R^{(2)}(\Sigma'_2)$는 각각 아래와 같이 구해진다.

$$R^{(2)}(\Sigma'_1) = \begin{bmatrix} \{b,bb\} & \{a,aa,ad,ba\} & \{ac\} \\ N & \{a,d,aa,dd,cd\} & \{c,ca,cc,ac,dc\} \\ N & \{d,da,dd,ad,cd\} & \{a,c,aa,cc,dc\} \end{bmatrix}$$

$$R^{(2)}(\Sigma'_2) = \begin{bmatrix} \{b,bb\} & \{ad\} & \{a,aa,ac,ba\} \\ N & \{a,d,aa,dd,cd\} & \{c,ca,cc,ac,dc\} \\ N & \{d,da,dd,ad,cd\} & \{a,c,aa,cc,dc\} \end{bmatrix} \quad (17)$$

위 행렬식에서 N은 두 상태 사이를 연결하는 입력 스트링이 존재하지 않는다는 의미이다. skeleton 행렬은 stable transition 행렬로부터 간단하게 구해진다. stable transition 행렬의 원소가 N이면 skeleton 행렬의 해당 원소는 0, 그렇지 않으면 해당 원소는 1의 값을 가진다.

$$K(\Sigma'_1) = \begin{bmatrix} 1\,1\,1 \\ 0\,1\,1 \\ 0\,1\,1 \end{bmatrix}$$

$$K(\Sigma'_2) = \begin{bmatrix} 1\,1\,1 \\ 0\,1\,1 \\ 0\,1\,1 \end{bmatrix} \quad (18)$$

식 (16)에 따라서 비결정 모델 Σ'의 skeleton 행렬을 구하면 아래와 같다.

$$K(\Sigma') = \begin{bmatrix} 1\,1\,1 \\ 0\,1\,1 \\ 0\,1\,1 \end{bmatrix} \quad (19)$$

그림 2의 모델에 그림 4와 같은 DFSM $\Delta\Sigma'$이 추가

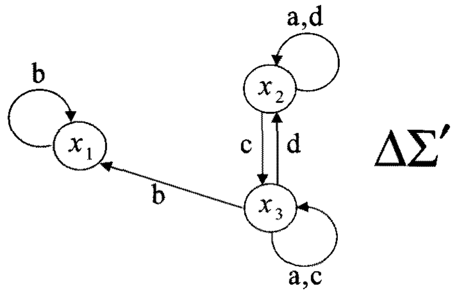

그림 4. 추가 모델 $\Delta\Sigma'$
Fig. 4. Added model $\Delta\Sigma'$.

되었다고 하자. 식 (17), (18)과 유사하게 $\Delta\Sigma'$의 skeleton 행렬을 구하면 아래와 같이 나온다.

$$K(\Delta\Sigma') = \begin{bmatrix} 1\,0\,0 \\ 0\,1\,1 \\ 1\,1\,1 \end{bmatrix} \tag{20}$$

새로운 비결정 모델 $\Sigma''=\Sigma'+\Delta\Sigma'$의 skeleton 행렬은 식 (19)와 (20)에서 구한 각 skeleton 행렬을 식 (16)의 공식을 이용하여 결합하면 된다. 따라서 $K(\Sigma'')$는 아래와 같이 나온다.

$$K(\Delta\Sigma'') = \begin{bmatrix} 1\,1\,1 \\ 0\,1\,1 \\ 1\,1\,1 \end{bmatrix} \tag{21}$$

이상과 같이 이번 연구에서 제안한 도달가능성 행렬은 결정적 모델을 포함하는 모든 비결정 비동기 머신 모델의 내부 연결성을 간략하게 표시하는 기능을 가진다.

이와 같이 구해진 도달가능성 행렬들은 교정 제어기의 존재 조건을 표시하고 제어기를 설계하는 과정에서 사용될 것이다.

Ⅶ. 결 론

본 논문의 성과를 요약하면 다음과 같다. 먼저 비동기 순차 머신의 교정 제어를 소개하고 머신이 추종해야 하는 모델이 비결정적일 때의 모델 매칭 문제를 정의하였다. 또 비결정 모델을 폐루프 시스템에 포함시킨 교정 제어 시스템(그림 3)을 제안하고 그 동작 과정을 기술하였다. 또한 비결정 유한 상태 머신으로 표현되는 모델을 분해하고 합성하는 방법을 제안하였으며 도달가능성 행렬 계산 과정을 제시하였다.

본 논문에서 소개하는 비동기 순차 머신의 모델 매칭 및 교정 제어는 컴퓨터 회로, 통신 네트워크 등 비동기적으로 구성된 여러 가지 시스템을 전통적인 피드백 제어 방식으로 다루는 새로운 방법론이다[4]. 해저드(hazard) 등 비동기 머신 고유의 문제가 존재하는 시스템의 동작을 재설계 과정을 거치지 않고 바꾼다는 점에 연구 의의를 둘 수 있다.

본 논문에서 제안된 모델 매칭 문제를 풀기 위한 교정 제어기의 존재 조건 규명과 자세한 설계 과정은 후속 논문[6]에서 기술한다.

참 고 문 헌

[1] S. Hauck, “Asynchronous design methodologies: an overview,” *Proceedings of the IEEE*, vol. 83, no. 1, pp. 69-93, 1995.

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

[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] X. Geng and J. Hammer, “Input/output control of asynchronous sequential machines,” *IEEE Transactions on Automatic Control*, vol. 50, no. 12, pp. 1956-1970, 2005.

[5] M. D. Di Benedetto, 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.

[6] 양정민, “비결정 모델에 대한 비동기 순차 회로의 교정 제어 II: 제어기 설계,” *대한전자공학회 논문지* 제45권 SC 제7호.

[7] S. M. Nowick and D. L. Dill, “Synthesis of asynchronous state machines using a local clock,” in *Proceedings of IEEE International Conference on Computer Design*, pp. 192-197, 1991.

[8] Z. Kohavi, *Switching and Finite Automata Theory*, McGraw-Hill, New York, 1970.

[9] C. G. Cassandras and S. Lafortune, *Introduction to Discrete Event Systems*, Kluwer Academic Publishers, Boston, 1999.

[10] R. Kumar and M. A. Shayman, “Nonblocking supervisory control of nondeterministic systems via prioritized synchronization,” *IEEE Transactions on Automatic Control*, vol. 41, no. 8, pp. 1160-1175, 1996.

[11] F. Lin, “Robust and adaptive supervisory control of discrete event systems,” *IEEE Transactions on Automatic Control*, vol. 38, no. 12, pp. 1848-1852, 1993.

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>