# 입력 외란이 존재하는 비동기 순차 머신의 교정 제어 I : 모델링

論 文
56-9-23

# Corrective Control of Asynchronous Sequential Machines with Input Disturbance I : Modeling

楊 正 敏†
(Jung-Min Yang)

**Abstract** - This paper presents the problem of controlling asynchronous sequential machines in the presence of input disturbances, which may be also regarded as an adversary in a game theoretic setting. The main objective is to develope a new methodology for including unpredictable behavior of input disturbance into models of asynchronous machines. The input disturbance, representing uncontrollable noise input, is embedded into a new model of asynchronous machines in form of input/state finite state machines. It is shown that the proposed modeling preserves the fundamental model and well-pose of asynchronous machines. The reachability matrix, an important performance index of asynchronous machines, is also adapted according to input disturbance and will be used for constructing corrective controllers in the companion paper.

**Key Words** : Asynchronous Sequential Machines, Corrective Control, Input Disturbance, Model Matching

## 1. 서 론

비동기 순차 머신(Asynchronous Sequential Machine)은 상태변화가 클럭(clock)에 상관없이 발생하는 머신(machine)으로 동기 순차 머신(Asynchronous Sequential Machine)과 대조되는 특징을 가지고 있다. 비동기 순차 머신은 클럭 없이 동작하므로 동기 순차 머신에 비해서 전력 소비가 적고, 연산 속도가 빠르며, 모듈화된 설계 방법이 적용 가능하다는 점 등의 장점을 지니고 있다. 이러한 비동기 순차 머신의 설계에 대한 연구는 1950년대 중반부터 이루어져 왔으며, 최근 실용적인 설계 기법들이 개발되면서 다시 주목을 받고 있다[1]-[3]. 하지만 비동기 순차 머신은 해저드(hazard)나 메타안정성(meta-stability), 레이스(race) 등의 오동작이 내재할 수 있기 때문에 정상적인 동작을 하는 머신 설계를 완수하기 위해서는 많은 주의를 필요로 한다.

본 논문은 새로운 비동기 순차 머신의 설계에 관한 연구가 아니라 기존에 존재하는 비동기 순차 머신이 원하는 동작을 행하도록 해주는 제어기를 구현하는 접근 방식을 취한다. 본 논문에서는 전통적인 자동 제어의 개념을 도입하여 제어기를 비동기 순차 머신 앞에 연결시키는 기법을 사용한다. 이 제어기는 외부로부터 입력 신호를 받고 비동기 순차 머신의 출력을 피드백으로 받아 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 비동기 순차 머신에 전달하는 역할만을 수행할 것이다. 비동기 순차 머신이 원하지 않는 동작을 보이기 시작하면 제어기는 적절한 제어 입력을 만들어 낸다. 이 제어기는 비동기 순차 머신을 다루어야 하므로 역시 클럭 없는 비동기 순차 머신의 형태로 구성되어야 한다. 교정 제어(corrective control)라 불리는 이러한 개념은 Hammer[4]가 일반적인 순차 머신의 제어를 위해서 맨 처음 도입하였으며, 최근 레이스가 존재하는 비동기 순차 머신[5], 무한 순환(infinite cycle)이 존재하는 비동기 순차 머신[6]의 제어에 성공적으로 적용되었다.

본 논문의 주요 목적은 입력 외란(input disturbance)이 존재하는 비동기 순차 머신의 모델링 및 해석 방법을 제안하는 것이다. 입력 외란은 외부로부터 비동기 순차 머신에 유입되는 신호로서, 사용자나 제어기의 명령에 상관없이 시스템의 상태를 천이시킬 수 있는 신호라고 정의한다. 즉 본 논문에서 고려되는 비동기 순차 머신은 정상적인 입력 신호 외에 비정상적인 외란 입력 신호를 하나 더 가진다고 가정한다. 입력 외란은 정확한 값을 예측하기 불가능하며 제어기가 외란의 발생을 막을 수도 없다. 이러한 개념은 이산사건 시스템(Discrete-Event Systems: DES) 분야에서 정의되는 제어불능 이벤트(uncontrollable event)[7]와 유사한 특징을 가진다. 본 논문에서는 입력 외란을 비동기 순차 머신이 가지는 또 하나의 입력부라고 설정해서 두 개의 입력을 가지는 비동기 순차 머신을 모델링하고, 입력 외란이 존재하는 비동기 순차 머신을 위한 새로운 교정 제어 시스템을 제안한다. 본 교정 제어의 목적은 비동기 순차 머신의 폐루프 시스템 동작이 입력 외란에 영향을 받지 않도록 하는 것이다. 즉 제어기는 입력 외란이 비동기 순차 머신에 들어옴에도 불구하고 머신의 동작에서 입력 외란의 영향을

† 교신저자, 正會員 : 大邱가톨릭大 電子工學科 副敎授 · 工博
E-mail : jmyang@cu.ac.kr
接受日字 : 2007年 5月 10日
最終完了 : 2007年 7月 5日

없애고 시스템이 원하는 모델과 동일한 입력/출력 쌍을 가지도록 만든다. 비동기 순차 머신의 모델 매칭 문제를 풀기 위해서는 시스템이 가지고 있는 도달가능성(reachability)을 정량적으로 구할 필요가 있다. 기존 연구 [5],[6]에서는 "skeleton 행렬(matrix)"이라는 행렬식을 정의하여 비동기 순차 머신의 도달가능성을 표현하였다. 본 논문에서는 입력 외란이 존재하는 비동기 순차 머신의 도달가능성을 표현하는 새로운 행렬식을 제안하고 예제를 통해서 그 계산과정을 보인다. 입력 외란의 영향을 극복하면서 비동기 순차 머신의 모델 매칭 문제를 해결하는 제어기가 존재할 구체적인 조건과 제어기 설계 과정 등은 본 논문의 후속편[8]에서 기술한다.

## 2. 비동기 순차 머신

본 장에서는 먼저 입력 외란이 없는 정상적인 비동기 순차 머신의 모델링 방법을 기술한다. 일반적으로 사용되는 비동기 순차 머신의 논리적 모델은 다음과 같은 유한 상태 머신(Finite State Machine)이다.

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

위 식에서 $A$는 입력 알파벳 집합, $Y$는 출력 알파벳 집합, $X$는 $n$개의 원소로 이루어진 머신의 상태 집합이다. $x_0$는 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.

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

$u_0, u_1, u_2, \ldots$는 머신의 입력 시퀀스(sequence)이며 $x_0, x_1, x_2, \ldots$와 $y_0, y_1, y_2, \ldots$는 각각 머신의 상태와 출력 시퀀스를 가리킨다. 머신의 스텝(step) $k$는 입력이나 상태 변수가 변경되었을 때마다 1씩 증가한다.

'valid pair' $(x, u) \in (X \times A)$는 유한 상태 머신에서 $f$와 $h$가 각각 정의된 입력과 상태 쌍을 말한다. (즉 $f$와 $h$가 정의되는 않는 입력과 상태 쌍도 존재한다.) 본 논문에서는 비동기 순차 머신의 현재 상태가 출력값으로 나오는 입력/상태 머신(input/state machine)에서 대해서 다루기로 한다. 즉 출력 알파벳 집합 $Y$는 $X$와 일치하며 출력 함수도 아래와 같이 간단하게 표시된다.

$$y_k = x_k,\ k = 0, 1, 2, \ldots$$

valid pair $(x,u)$는 $x=f(x,u)$일 때 'stable combination'이라고 정의된다. 이때 $x$는 '안정 상태(stable state)'라 불리며 함수 $f$의 고정점(fixed point)이 된다. 비동기 순차 머신은 입력이 바뀌지 않는 한 stable combination에서 계속 머물러 있게 된다. stable combination이 아닌 입력과 상태 쌍을 'transient combination'(또는 transient pair)이라고 부르고 transient combination을 이루는 상태를 '불안정 상태(unstable state)'라고 정의한다. '잠재적 안정 상태(potentially stable state)'는 stable combination을 이루는 입력 알파벳을 적어도 하나 이상 가지고 있는 상태를 말한다.

비동기 순차 머신에서 transient pair $(x,u)$는 입력이 바뀌지 않는 한 $x_1 = f(x, u)$, $x_2 = f(x_1, u), \ldots$와 같이 연쇄적으로 상태 천이를 시작하여 stable combination에 도달한다. 즉 $q \geq 1$인 자연수 $q$가 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 stable combination이 된다. 만약 연쇄 천이가 끝나지 않고 transient combination 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 stable combination에 도달한 후에만 바뀔 수 있다. 만약 비동기 순차 머신이 transient combination에 있을 때 입력이 바뀐다면 (클럭이 없기 때문에) 머신이 현재 어떤 상태에 있는지 알기 모호하므로 상태 천이 함수를 적용하기가 불가능해진다. 최근 여러 개의 입력 변화가 동시에 허용되는 burst 모드[9]에서 비동기 순차 머신을 설계하는 기법들도 연구되고 있으나 본 논문에서 다루는 교정 제어에서 사용하기는 난점이 있다. 따라서 본 논문에서는 비동기 순차 머신이 기본 모드에서 동작하는 것을 기본 가정으로 삼기로 한다.

비동기 순차 머신 $\Sigma$의 'stable recursion' 함수 $s$는 모든 valid pair에서 정의되는 함수로서 valid pair에서 시작하여 비동기 순차 머신이 다다르는 다음 안정 상태(next stable state)를 그 함수값으로 가진다. 즉 $s : X \times A \rightarrow X$이고 valid pair $(x,u)$에서 $s(x,u) := x'$이며 $x'$는 $(x,u)$의 다음 안정 상태이다. 정의에 따라서 $(x',u)$는 stable combination이 된다. $f$ 대신 $s$를 이용하여 $\Sigma$의 'stable state 머신'을

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

이라고 정의하고 $\Sigma_{|s}$라고 명명한다. stable state 머신 $\Sigma_{|s}$는 $\Sigma$에서 상태 천이 함수 $f$ 대신 stable recursion 함수 $s$를 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 stable combination 동작만이 유효하다고 말할 수 있다.

기본 모드에서는 한 번에 한 개의 변수만이 바뀔 수 있으므로 비동기 순차 머신의 입력은 머신이 stable combination에 있을 때에만 그 값이 변경 가능하다고 하였다. 머신이 초기 상태 $x_0$에 있고 입력 시퀀스 $u = u_0 u_1 \cdots u_{m-1} \in A^+$가 들어온다고 가정하자. 이 때 $A^+$는 $A$의 알파벳으로 만들어지는 스트링(string) 중 빈 스

트링 (empty string) ε를 제외한 집합을 말한다. 기본 모드 동작에서 첫 번째 입력 알파벳 $u_0$은 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 $x_1$에 도달하면 입력은 $u_0$에서 $u_1$로 바뀌며, Σ는 다시 다음 안정 상태 $x_2 := s(s(x_0, u_0), u_1)$에 도달할 때까지 입력을 $u_1$로 유지한다. 이런 식으로 비동기 순차 머신은 마지막 안정 상태 $x_m := s(\ldots s(s(s(x_0, u_0), u_1), u_2)\ldots, u_{m-1})$에 도달한다. 입력 시퀀스가 주어지면 s 함수는 일반적으로 다음과 같이 간편하게 표기된다.

$$s(x_0, u) := (\ldots s(s(s(x_0, u_0), u_1), u_2)\ldots, u_{m-1}), \quad u \in A^+ \tag{2}$$

기본 모드로 동작하는 비동기 순차 머신은 머신이 stable combination에 있을 때에만 제어값을 변경할 수 있다. 또한 제어기를 설계하려면 비동기 순차 머신이 stable state 머신으로 표현되었을 때 머신의 상태들이 서로 도달가능한지를 아는 것이 매우 중요하다. 본 논문에서는 기존 연구 [5],[6]에서 정의된 안정적 도달가능성(stable reachability)를 도입한다.

**정의 1**: $x$와 $x'$는 비동기 순차 머신의 잠재 안정 상태이고 $s$는 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 $x$에서 '안정적으로 도달가능(stably reachable)'하다.

본 장에서 도입한 정의와 표기법의 자세한 내용은 기존 연구 [5],[6]과 Kohavi[10]에서 찾을 수 있다.

## 3. 입력 외란과 폐루프 시스템

### 3.1 입력 외란 모델링

이번 절에서는 본 논문의 주요 목적인 입력 외란 모델링과 비동기 순차 머신 교정 제어를 위한 폐루프 시스템 구조를 제안한다.

그림 1은 기존 연구 [5],[6]에서 사용되었던 비동기 순차 머신을 위한 교정 제어의 기본 구조이다. 불완전한 동작을 보이는 비동기 순차 머신 Σ 앞에 상태 피드백 제어기 C가 연결되어 폐루프 시스템이 원하는 동작을 하도록 제어 입력을 만든다. $v$는 외부로부터 들어오는 입력 신호이고 $u$는 제어 입력, $y$는 비동기 순차 머신의 출력이다. 본 논문에서는 입력/상태 머신을 다루므로 $y$는 머신의 현재 상태 $x$의 값을 가진다. 그런데 입력 외란이 존재하는 비동기 순차 머신은 입력 단이 하나 더 필요하므로 그림 1과 같은 구조와는 다른 형태의 폐루프 시스템을 가져야 한다.

본 논문에서 제안하는 비동기 순차 머신 제어 시스템 구조는 그림 2와 같다. 그림 2에서 비동기 순차 머신은 두 개의 입력 $u_1$과 $u_2$를 가진다. 제어기 C는 상태 피드백 제어부(control unit) $C$와 모델 Σ'로 구성된다. 제어부 $C$는 역시 비동기 순환 머신으로 구현되는데, 외부 입력 $v$와 상태 피드백 $y$를 받아서 제어 입력 $u_1$을 생성하여 Σ에 보낸다. 모델 Σ'는 외부 입력 v에 대한 현재의 상태 $y_r$를 제어부 $C$에 보낸다. Σ'는 폐루프 시스템의 동작이 따라가야 할 모델이며, $y_r$는 자동 제어의 모델 추종 제어(model following control)에서 쓰이는 기준 입력(reference input)에 해당되는 신호이다. Σ'는 미리 주어지는 값이므로 본 논문에서는 앞으로 상태 피드백 제어부 $C$가 바로 제어기 C를 의미한다고 설정한다.

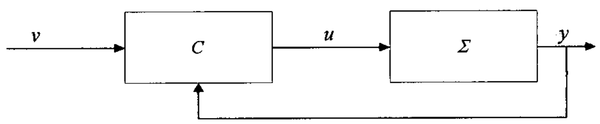



**그림 1.** 기본 제어 구조.

**Fig. 1.** Basic control configuration.

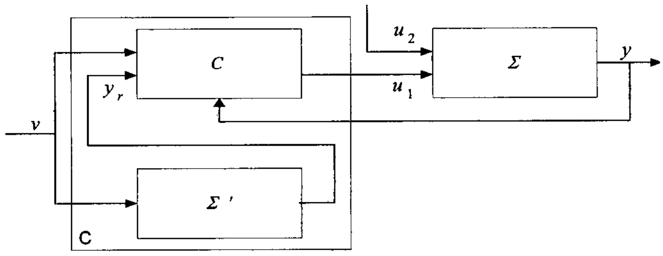



**그림 2.** 입력 외란이 존재하는 비동기 순차 머신의 폐루프 시스템.

**Fig. 2.** Closed-loop system of an asynchronous sequential machine with input disturbance.

그림 2에서 $u_2$는 입력 외란을 표시한다. 그림에서 알 수 있듯이 $u_2$는 제어기 $C$를 거치지 않고 Σ로 직접 들어가므로, 제어기가 $u_2$의 발생을 막을 수 없다. 즉 $u_2$는 제어 불능(uncontrollable) 입력이다. 본 논문에서 $u_2$는 간헐적으로 물리적 신호값을 가진다고 설정한다. $u_2$가 아무 값을 가지지 않을 때에는 $u_2 = \epsilon$(비어 있는 스트링)로 표시하며, 비동기 순차 머신 Σ는 입력이 한 개일 때와 마찬가지로 정상적인 동작을 한다. $u_2$로부터 물리적 신호(외란)가 들어온다면 $u_2$는 제어 입력 $u_1$에 상관없이 비동기 순차 머신 Σ의 상태를 변경할 수 있다. $u_2$가 Σ의 현재 상태 $x$, 제어 입력 $u_1$와 valid pair를 구성한다면, 즉 $(x, (u_1, u_2))$에 대해서 함수 $f$가 정의되어 있다면 비동기 순차 머신은 원하지 않는 상태 천이를 겪어야 한다.

하지만 Σ가 어떤 조건을 만족한다면 입력 외란 $u_2$의 영향을 무력화시킬 수 있는 상태 피드백 제어기 $C$가 존재할 수 있고, 폐루프 시스템과 모델 Σ' 사이의 모델 매칭 문제를 풀 수 있다. 제어기 $C$는 상태 피드백 $y$값을 관측함으로써 입력 외란에 의한 상태 천이의 발생을 감지하며, 기준 입력 $y_r$을 이용하여 비동기 순차 머신을 원래 상태로 되돌리

는 제어 입력 $u_1$을 만든다. 서론에서 밝혔듯이 제어기 $C$가 존재할 구체적인 조건과 제어기 설계 과정 등은 후속 논문에서 기술한다.

입력 외란을 포함시킨 비동기 순차 머신의 유한 상태 머신 식은 아래와 같다.

$$\Sigma = (A_1 \times A_2, Y, X, x_0, f, h) \tag{3}$$

위 식에서 $A_1$은 정상적인 외부 입력이 가질 수 있는 알파벳 집합이며 $A_2(\epsilon \in A_2)$는 입력 외란 $u_2$의 알파벳 집합이다. 상태 천이 함수 $f$의 매핑(mapping)은 $f: X \times A_1 \times A_2 \rightarrow X$으로 확대된다. $u_2 = \epsilon$일 때 $f(x, (u_1, \epsilon))$은 정상 입력 $u_1$에 의한 동작을 보이지만 $u_2 \neq \epsilon$일 때에는 $f(x, (u_1, u_2))$가 원하지 않는 상태로 천이될 수 있다. 두 개의 입력이 동시에 나타나는 비동기 순차 머신 모델을 설정하였기 때문에 반드시 $A_1 \cap A_2 = \varnothing$일 필요는 없다. 즉 입력 외란 $u_2$는 정상적인 입력 $u_1$의 알파벳 값을 가질 수도 있다.

그림 2의 폐루프 시스템이 기본 모드로 동작하기 위해서는 제어기 C와 입력 외란 $u_2$도 기본 모드를 만족시켜야 한다. Geng과 Hammer[11]가 정립한 복합 시스템의 기본 모드 동작 조건을 이용하면 폐루프 시스템이 만족시켜야 할 조건은 아래와 같이 표현된다.

**정리 1**: $\Sigma$와 $C$는 그림 2와 같이 상호 연결된 비동기 순차 머신이다. 다음 조건이 만족되면 폐루프 시스템은 기본 모드로 동작한다.

(i) $\Sigma$가 stable combination에 있지 않을 때 $C$의 출력 $u_1$과 입력 외란 $u_2$은 그 값이 바뀌지 않는다.

(ii) $C$가 stable combination에 있지 않을 때 $\Sigma$의 출력 $y$와 외부 입력 $v$, 그리고 기준 입력 $y_r$은 그 값이 바뀌지 않는다.

위 정리에서 조건 (i)은 비동기 순차 머신 $\Sigma$가 stable combination에 다다를 때까지 입력 외란 $u_2$의 값이 바뀌지 말아야 한다는 것을 의미한다. 외란의 기본적인 성질은 예측 불가능성이므로 이러한 조건은 제한적이라고도 볼 수 있다. 그러나 앞서 거론했듯이 비동기 순차 머신이 기본 모드를 지키지 못하면 복수 개의 레이스가 발생할 수 있기 때문에[10] 교정 제어 자체가 불가능해진다. 또한 비동기 순차 머신의 상태를 변경시킬 수 있는 입력 외란의 존재는 이 외란이 시스템 외부로부터 침투해올 때 기본 모드 동작을 준수하여 생존했다는 사실을 의미하므로 본 논문의 가정은 타당하다 할 수 있다. 비동기 순차 머신 교정 제어에 대한 기존 연구들[5],[6],[11]도 모두 폐루프 시스템이 기본 모드를 만족시킨다는 가정 하에서 제어기를 설계하였다.

입력 외란 $u_2$는 정상 입력 $u_1$에 상관없이 발생하므로 $u_2$가 정의된 상태에서 $u_1$의 값은 중요하지 않다. 따라서 상태 $x$에서 입력 외란 $u_2$가 발생한다면 상태 천이 함수 f는 다음과 같이 표기될 수 있다.

$$f(x, (u_1, u_2)) := f(x, (-, u_2)),\ \forall\, u_1 \in A_1,\ \forall\, u_2 \neq \epsilon \tag{4}$$

위 식에서 '-'은 모든 정상 입력 $u_1$에 대해서 입력 외란 $u_2$가 정의된다는 의미이다. 식 (4)는 입력 외란 $u_2$가 $u_1$과 독립적으로 발생 가능하다는 것을 의미한다. 그러나 정리 1에서 $u_2$ 역시 기본 모드를 만족시켜야 하므로, 최초로 발생하는 입력 외란 $u_2$는 비동기 순차 머신의 상태가 stable combination에 있을 때에만 가능하다. 즉 $\Sigma$가 정상 입력 $u_1$에 의해서 안정 상태에 머물러 있을 때 입력 외란 $u_2$로부터 어떤 물리적 값이 들어와 상태 천이가 시작되며, $\Sigma$는 다시 $u_2$에 대한 stable combination에 다다르게 된다. 따라서 외란 $u_2$을 포함하는 stable recursion 함수 $s$는 다음과 같이 정의된다.

$$\begin{aligned} &s(x, (u_1^i, u_2)) := s(x, (u_1^j, u_2)),\ \forall\, u_1^i, u_1^j \in U(x), \\ &s(x, (u_1, u_2)) := \text{undefined},\ \forall\, u_1 \not\in U(x) \end{aligned} \tag{5}$$

위 식에서 $U(x)(\subset A_1)$는 상태 $x$와 stable combination을 이루는 모든 정상 입력 알파벳 집합이다. 입력 외란 $u_2$는 상태 $x$가 stable combination에 있을 때에만 발생 가능하므로 $u_1 \in U(x)$이어야 한다. 또 $u_2$는 $u_1$에 독립적이므로 $u_1 \in U(x)$인 모든 $u_1$에 대해서 동일한 stable recursion 값을 가진다.

본 논문에서 제안하는 비동기 순차 머신의 입력 외란 모델링을 요약하면 다음과 같다. 비동기 순차 머신은 평상시 정상 입력 $u_1$의 값을 받아서 상태 천이를 하며, stable combination에 다다르면 입력 값이 바뀔 때까지 대기한다. 이때 입력 외란이 발생하여 $u_2$로부터 물리적 값이 들어온다면 비동기 순차 머신은 $u_2$에 의해서 원하지 않는 상태 천이를 겪게 되고, 사용자가 원했던 상태가 아닌 다른 stable combination에 도달하게 된다. 다른 stable combination에 도달하는 순간 $u_2$의 값은 $\epsilon$이 된다. 만약 이 상태에서 교정 제어를 하지 않은 채 새로운 정상 입력 $u_1$가 들어온다면 비동기 순차 머신은 원래의 상태가 아닌 곳에서 다음 상태 천이를 하므로 엉뚱한 결과를 낳을 것이다. 이러한 문제를 해결하는 제어 기법을 제안하는 것이 본 논문과 후속 논문의 주제이다.

### 3.2 제어기 구조

폐루프 시스템에서 사용되는 제어기 $C$를 기술한다. 그림 2에서 $C$는 외부 입력 $v$, 상태 피드백 $y$, 그리고 기준 입력 $y_r$ 등 세 개의 입력을 가진다. $v$는 정상 입력 알파벳 $A_1$의 값이며 $\Sigma$와 $\Sigma'$ 모두 입력/상태 머신이므로 $C$의 입력 집합은 $A_1 \times X \times X$이 된다. C의 출력 집합은 $\Sigma$로 들어가는 제어 입력이므로 $A_1$이어야 한다. 따라서 $C$의 유한 상태 머신 표현은 아래와 같이 정의된다.

$$C=(A_1\times X\times X, A_1, \Xi, \xi_0, \phi, \eta)$$

위 식에서 $\Xi$은 $C$의 상태 집합이며 $\xi_0$는 초기 상태, $\phi$는 recursion 함수, $\eta$는 출력 함수이다. 그림 2의 폐루프 시스템을 $\Sigma_c$라고 표기하면 $\Sigma_c$는 $C$와 $\Sigma$로 이루어진 복합 시스템이므로 상태 집합 $X\times\Xi$, 입력 집합 $A_1\times A_2$를 가진다. $f_c$와 $h_c$를 각각 $\Sigma_c$의 상태 천이 함수, 출력 함수라고 정의하면 폐루프 시스템의 유한 상태 머신 표현식은 아래와 같다.

$$\Sigma_c=(A_1\times A_2, X, X\times\Xi, (x_0,\xi_0), f_c, h_c)$$

$\Sigma$가 입력/상태 머신이므로 $\Sigma_c$의 출력 함수 $h_c$는 $C$의 모든 상태 $\xi\in\Xi$와 $\Sigma_c$의 모든 valid pair $(x,(u_1,u_2))$에 대해서 $h_c(\xi,x,(u_1,u_2))=x$와 같은 항등 함수(identity function)가 된다.

앞서 기술했듯이 비동기 순차 머신이 불안정 상태에 머무르는 시간은 이론적으로 0이므로 외부 사용자는 오직 비동기 순차 머신의 stable combination만을 인지할 수 있다. 이러한 성질을 바탕으로 비동기 순차 머신의 stable equivalence를 아래와 같이 정의한다[5].

**정의 2**: 두 개의 비동기 순차 머신 $\Sigma=(A,Y,X,x_0,f,h)$와 $\Sigma'=(A,Y,X',\zeta_0,f',h')$가 동일한 입력과 출력 집합을 공유한다고 하고 $\Sigma_{|s}$와 $\Sigma'_{|s}$를 각각 $\Sigma$와 $\Sigma'$의 stable state 머신이라고 표기한다. 상태 $x\in X$와 $x'\in X'$는 다음과 같은 조건을 만족할 때 서로 'stable equivalent($x\equiv x'$)'하다고 정의한다: $\Sigma_{|s}$는 상태 $x$에 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 개의 초기 상태가 서로 stable equivalent하다면, 즉 $x_0\equiv\zeta_0$이면 비동기 순차 머신 $\Sigma$와 $\Sigma'$는 서로 'stable equivalent'하다고 정의하고 $\Sigma=\Sigma'$로 표기한다.

비동기 순차 머신 두 상태가 받을 수 있는 입력 시퀀스와 만들어 내는 출력 시퀀스가 동일하다면, 그 상태들은 외부 사용자에게 다르게 구별되지 않을 것이므로 equivalent하다고 말할 수 있다. 정의 2에서 잊지 말아야 할 중요한 조건은 반드시 비동기 순차 머신의 stable state 머신 하에서 stable equivalence를 판단해야 한다는 사실이다.

stable equivalence 조건을 가지고 입력 외란이 존재하는 비동기 순차 머신의 모델 매칭 문제를 다음과 같이 정의한다. 아래 정의에서 $\Sigma_{c|s}$는 폐루프 시스템 $\Sigma_c$의 stable state 머신을 가리킨다.

**입력 외란이 존재하는 비동기 순차 머신의 모델 매칭 문제**: $\Sigma$는 입력 외란이 존재하는 비동기 순차 머신이며 식 (3)과 같은 유한 상태 머신으로 표현된다. $\Sigma'$는 정상적인 비동기 순차 머신이며 stable state 머신으로 표현된다. 정상적인 입력 알파벳 $A_1$에 대해서 폐루프 시스템 $\Sigma_{c|s}$이 $\Sigma'$와 stable equivalent하는 동작을 보이도록 만드는 교정 제어기 $C$가 존재할 필요충분조건을 구하고, $C$가 존재한다면 설계 알고리즘을 찾는다.

두 비동기 순차 머신이 서로 stable equivalent 하기 위해서는 동일한 입력 집합을 가져야 한다. 하지만 식 (1)과 (3)에서 알 수 있듯이 $\Sigma$는 입력 외란을 포함해서 두 개의 입력부를 가지며 모델 $\Sigma'$는 한 개의 입력만을 가진다. 이 문제는 $\Sigma'$가 가상의 입력부를 하나 더 가지며 이 입력의 값은 항상 $\epsilon$라고 가정함으로써 해결할 수 있다. 즉 모델 $\Sigma'$의 stable recursion 함수 $s'(x,v)$를 $s'(x,(v,\varepsilon))$와 같이 설정한다.

위 문제 제기에서 모델 $\Sigma'$는 stable state 머신으로 표현된다고 하였다. 모델 매칭 문제를 푸는 교정 제어 $C$는 $\Sigma$에서 발생하는 입력 외란의 영향을 없애면서 stable state에서 $\Sigma$가 $\Sigma'$와 똑같은 행동을 보이게 만든다. 즉 모델 매칭 문제의 주 관심사는 stable state 머신이므로 모델 $\Sigma'$을 stable state 머신으로 정의해도 일반성을 잃지 않는다.

위에서 기술한 입력 외란이 존재하는 비동기 순차 머신의 모델 매칭 문제는 레이스가 존재하는 비동기 순차 머신의 기존 모델 매칭 문제[5]와 분명한 차이점을 가진다. 식 (4)와 (5)에서 기술했듯이 입력 외란은 머신이 안정 상태에 있을 때 발생하므로 불안정 상태에서 발생하는 레이스를 없애는 문제와는 다른 접근 방법을 사용해야 할 것이다. 제어기를 구현하기 위해서 가정 먼저 해야 할 일은 비동기 순차 머신이 두 개의 입력에 대해서 가지는 도달가능성을 각각 정량화하는 것이다. 다음 장에서는 본 장에서 제안한 비동기 순차 머신 모델에 맞는 도달가능성 행렬을 정의하고 계산 과정을 제시한다.

## 4. 비동기 순차 머신의 도달가능성

비동기 순차 머신의 도달가능성이란 비동기 순차 머신의 상태들이 서로 천이할 수 있는 경로가 존재하는지 여부를 말한다. 비동기 순차 머신이 입력 외란에 의해서 원하지 않는 상태로 천이된다면 그 상태에서 원래의 상태를 연결해주는 입력 경로가 존재한다면 교정 제어에 의해서 머신의 상태를 되돌리는 일이 가능할 것이다. 본 논문에서는 기존 연구 [5],[11]에서 제시된 도달가능성 행렬들을 이용하여 입력 외란에 대한 비동기 순차 머신의 도달가능성을 정량화한다.

### 4.1 Stable Transition 행렬

**정의 3**: $\Sigma$는 입력 외란이 존재하는 비동기 순차 머신이며 식 (3)과 같이 표현된다. $\Sigma$의 상태 집합을 $X=\{x_1,x_2,\cdots x_n\}$이라 하고 $s$를 $\Sigma$의 stable recursion 함수라고 하자. 'one-step stable transition 행렬' $R(\Sigma)$ (n×n)은 다음과 같이 정의된다. $R(\Sigma)$의 원소 $R_{ij}(\Sigma)$은

$x_j = s(x_i, (u_1, \epsilon))$를 만족시키는 모든 정상 입력 알파벳 $u_1 \in A_1$의 집합이다. 만약 $R_{ij}(\Sigma)$가 공집합이면 $R_{ij}(\Sigma) := N$으로 표기되며 $N$은 $A_1$에 속하지 않는 알파벳이다.

$R(\Sigma)$은 $\Sigma$의 상태 사이에서 연결된 정상 입력 경로들을 행렬로 표현한 것으로 유한 상태 머신에서 구할 수 있다. $\Sigma$의 도달가능성을 더 자세하게 표현하기 위해서 아래와 같은 연산자 $\uplus$를 정의한다[5]. 또 스트링 집합 $A_1^*(N)$를 $A_1^* = \{e | e \in A_1^* \text{ or } e = N\}$와 같이 정의한다. 이 때 $A^*$는 $A_1$의 알파벳으로 이루어지는 (빈 스트링 ε을 포함하는) 모든 스트링 집합을 의미한다. 즉 $A_1^*(N)$는 $A_1$에서 만들어지는 모든 스트링과 알파벳 $N$을 합친 집합이다.

**정의 4**: 공집합이 아닌 $S_1$과 $S_2$는 $A_1^*(N)$의 부분 집합이다. 'unison' 연산 $S_1 \uplus S_2$는 다음과 같이 정의된다.

$$S_1 \uplus S_2 = \begin{cases} (S_1 \cup S_2) \cap A^*, & \text{if } (S_1 \cup S_2) \cap A^* \neq \varnothing \\ N & \text{otherwise} \end{cases}$$

unison 연산자는 일반적인 합집합 연산자와 동일하나 합집합 $S_1 \cup S_2$가 $A^*$의 스트링을 하나 이상 보유하고 있을 시에는 ($(S_1 \cup S_2) \cap A^* \neq \varnothing$) $S_1 \cup S_2$에서 알파벳 N을 제외시키는 일을 수행한다.($(S_1 \cup S_2) \cap A^*$)

**정의 5**: $A_1^*(N)$의 두 원소 $w_1, w_2 \in A_1^*(N)$에 대한 '연결(concatenation) 연산자'는 아래와 같이 정의된다.

$$conc(w_1, w_2) := \begin{cases} N & \text{if } w_1 = N \text{ or } w_2 = N \\ w_1 w_2 & \text{otherwise} \end{cases}$$

위 연결 연산자는 스트링에 대한 일반적인 연결 연산자와 같은 의미를 가지나 알파벳 $N$이 이 연산자에 대해서 곱셈 연산자의 0과 같은 역할을 한다는 점이 특징이다. unison 연산자를 이용하여 연결 연산자를 스트링의 집합에까지 확장해서 적용할 수 있다. 두 스트링 집합 $W_1$과 $W_2$가

$$W_1 = \{w_{1,1}, w_{1,2}, \dots, w_{1,q}\},$$
$$W_1 = \{w_{2,1}, w_{2,2}, \dots, w_{2,r}\}$$

와 같이 정의되어 있다고 할 때 $W_1$과 $W_2$에 대한 연결 연산자는 다음과 같다.

$$conc(W_1, W_2) := \biguplus_{\substack{i=1,\dots,q \\ j=1,\dots,r}} conc(w_{1,i}, w_{2,j})$$

**정의 6**: $A$와 $B$를 $A_1^*(N)$의 부분집합을 원소로 가지는 $n \times n$ 행렬이라고 하자. $A$, $B$의 곱행렬 $C := AB$는 다음과 같이 정의된다. $C_{ij}$는 $C$의 $(i,j)$번째 원소이다.

$$C_{ij} := \biguplus_{k=1,\dots,n} conc(A_{ik}, B_{kj}),\ i, j = 1, \dots, n$$

**정의 7**: $A$와 $B$를 $A_1^*(N)$의 부분집합을 원소로 가지는 $n \times n$ 행렬이라고 하자. $A$, $B$의 unison 연산 $A \uplus B$은 $(i,j)$번째 원소가 아래와 같이 정의되는 행렬이다.

$$(A \uplus B)_{ij} := A_{ij} \uplus B_{ij},\ i, j = 1, \dots, n$$

정의 6과 7은 $A_1^*(N)$의 부분집합을 원소로 가지는 행렬들의 곱과 합 연산을 각각 정의한다. 이 연산들은 일반적인 행렬의 곱과 합 연산자와 유사한 모습을 보이나, 곱셈 연산에는 연결 연산자 conc(*,*)를 사용하고 합 연산에는 unison 연산자 $\uplus$를 사용한다.

정의 6의 곱행렬을 이용하여 정의 3에서 도입한 one-step stable transition 행렬 $R(\Sigma)$의 거듭 제곱 행렬을 정의한다. $R(\Sigma)$을 $m$번 곱해서 만들어지는 행렬을 $R^m(\Sigma)$이라고 하자. 이 때 $R^m(\Sigma)$의 $(i,j)$번째 원소 $(R^m(\Sigma))_{ij}$가 $N$이 아니라면 이것은 비동기 순차 머신 $\Sigma$가 상태 $x_i$에서 $m$번의 stable transition을 거쳐서 $x_j$까지 도달하는 경로가 존재한다는 사실을 의미한다. $R(\Sigma)$의 거듭 제곱 행렬 $R^m(\Sigma)$과 unison 연산자를 이용하여 다음과 같은 행렬 $R^{(m)}(\Sigma)$을 정의하자.

$$R^{(m)}(\Sigma) := \biguplus_{i=1,\dots,m} R^i(\Sigma),\ m = 2, 3, \dots \quad (6)$$

행렬 $R^{(m)}(\Sigma)$의 $(i,j)$번째 원소는 비동기 순차 머신 $\Sigma$의 상태를 $x_i$에서 $x_j$로 천이시키는 입력 시퀀스 중 길이가 $m$보다 작거나 같은 것들을 모은 스트링 집합이다. 식 (6)과 유사하게 $R^*(\Sigma)$을 다음과 같이 정의한다.

$$R^*(\Sigma) := \biguplus_{i \geq 1} R^i(\Sigma)$$

$R^*(\Sigma)$는 $\Sigma$의 두 상태를 연결하는 모든 stable transition 입력 경로를 모은 행렬이다. 본 논문에서 고려하는 비동기 순차 머신 $\Sigma$는 무한 순환을 가지고 있지 않다고 가정했기 때문에 다음과 같은 관계가 성립한다.

**정리 2**: $R^*(\Sigma)$의 (i,j)번째 원소가 $N$일 필요충분조건은 $R^{(n-1)}(\Sigma)$의 (i,j)번째 원소가 $N$일 때이다.

**증명**: 위 정리에서 $n$은 $\Sigma$의 총 상태 개수이다. $\Sigma$에서 무한 순환이 일어나지 않으므로 한 상태에서 다른 상태로 천이될 때 $\Sigma$가 가질 수 있는 최대한의 stable transition 횟수는 $n$-1이다 ([5]의 Lemma 3.2 참조). 만약 $n$번 이상의 stable transition을 거친다면 어떤 불안정 상태가 이 경로 안에서 두 번 이상 도달되어야 하는데 이것은 무환 순환이 존재함을 의미하므로 원래의 가정과 모순된다. 즉 $R(\Sigma)$을 $n$-1번까지 거듭 제곱해서 구해진 행렬들을 모두 더하면 비동기 순차 머신 $\Sigma$의 도달가능성을 완전히 표현할 수 있다.

**정의 8**: $R^{(n-1)}(\Sigma)$을 비동기 순차 머신 $\Sigma$의 'stable transition 행렬'이라고 정의한다.

요약하면 $R^{(n-1)}(\Sigma)$의 $(i,j)$번째 원소는 $\Sigma$의 상태를 $x_i$에서 $x_j$로 천이시킬 수 있는 모든 입력 스트링들을 포함한다. $(i,j)$번째 원소가 $N$이라면 $x_i$에서 $x_j$로 연결되는 stable transition 경로는 존재하지 않음을 의미한다.

위에서 기술한 stable transition 행렬은 입력 외란은 발생하지 않는다는 가정 하에서(정의 3 참조) 정상 입력 $u_1$의 도달가능성을 나타낸 것으로 기존 연구 [5]에서 도입되었다. 하지만 입력 외란의 영향을 없애는 제어기를 설계하기 위해서는 비동기 순차 머신이 입력 외란 $u_2$에 대해서 어떠한 도달가능성을 가지고 있는지도 조사해야 한다. 본 논문에서는 다음과 같이 입력 외란에 대한 도달가능성 행렬을 정의한다.

**정의 9**: $\Sigma$는 입력 외란이 존재하는 비동기 순차 머신이며 식 (3)과 같이 표현된다. $\Sigma$의 상태 집합을 $X=\{x_1, x_2, \cdots x_n\}$이라 하고 $s$를 $\Sigma$의 stable recursion 함수라고 하자. '입력 외란에 대한 one-step stable transition 행렬' $R_d(\Sigma)$(n×n)은 다음과 같이 정의된다. $R_d(\Sigma)$의 원소 $R_{dij}(\Sigma)$은 $x_j = s(x_i, (u_1, u_2))$, $\forall u_1 \in U(x_i)$를 만족시키는 모든 입력 외란 $u_2 \in A_2$의 집합이다. 만약 $R_{dij}(\Sigma)$가 공집합이면 $R_{dij}(\Sigma) = N$으로 표기된다.

정의 3과 정의 9에서 도입된 $R(\Sigma)$과 $R_d(\Sigma)$의 차이점을 비교해보자. $R(\Sigma)$은 $u_2=\epsilon$, 즉 입력 외란의 값을 완전히 배제한 후 정상 입력에 대한 도달가능성을 표현한다. 반면 $R_d(\Sigma)$에서 $u_1$은 $u_1 \in U(x_i)$, 즉 현재 상태와 stable combination을 이루는 알파벳이라고 설정하였다. 식 (5)가 보여주듯이 입력 외란에 의한 상태 천이는 $u_1$의 값과 상관없으므로 $u_1 \in U(x_i)$인 조건하에서 $R_{dij}(\Sigma)$는 일정한 값을 가진다.

정상 입력에 대한 완전한 도달가능성을 구하기 위해서는 $R(\Sigma)$을 $n$-1번 거듭 제곱하는 과정이 필요하지만(정의 8), 입력 외란에 대한 도달가능성을 구할 때에는 이러한 거듭제곱 연산을 할 필요가 없다. 그 이유는 그림 2에서 제안된 교정 제어기가 입력 외란이 발생하자마자 작동을 하도록 설계되기 때문이다. 제어기의 동작 과정을 간략하게 설명하기 위해서 비동기 순차 머신 $\Sigma$가 정상 입력에 의해서 어떤 stable combination에 도달하였다고 가정하자. 이 때 $u_2 \neq \epsilon$인 입력 외란이 발생한다면 원하지 않는 상태 천이가 일어나 $\Sigma$의 상태가 다른 값으로 바뀌게 된다. 이 과정은 입력 외란에 의한 stable transition이 1회 발생한 것이라고 말할 수 있다. 정리 1에서 입력 외란도 기본 모드 동작을 만족시킨다고 규정하였기 때문에 $\Sigma$가 입력 외란이 만드는 stable combination에 도달하기 전까지 $u_2$의 값은 바뀌지 않는다. $\Sigma$가 입력 외란이 만드는 stable combination에 도달하면 교정 제어기 $C$는 그 즉시 상태 피드백 $y$의 값을 보고 외란에 의한 상태 천이가 일어났음을 감지한다. 그런 다음 적절한 제어 입력 $u_1$을 만들어 $\Sigma$가 원래의 상태로 되돌아오도록 하는 교정 제어를 실현한다. 교정 제어기 $C$도 비동기 순차 머신으로 구성되므로 외란의 발생을 감지하고 제어 입력을 만드는 데 걸리는 시간은 이론적으로 0이다. 즉 중요한 것은 원하지 않는 상태 천이가 일어난 후 입력 외란의 값이 또다시 바뀌어 새로운 상태 천이를 야기하기 전에 제어기 $C$의 동작이 시작된다는 점이다. 따라서 입력 외란의 도달가능성을 구할 때 길이가 2 이상인 입력 외란 시퀀스를 생각할 필요가 없으며, 정의 9에서 제시된 $R_d(\Sigma)$만을 찾으면 충분할 것이다.

### 4.2 Skeleton 행렬

본 절에서는 앞에서 구한 stable transition 행렬을 간략하게 표시하는 skeleton 행렬을 정의한다. skeleton 행렬은 교정 제어기의 존재 조건 규명과 설계 과정에서 유용하게 사용되는 중요한 행렬이다[5],[11].

**정의 10**: $R^{(n-1)}(\Sigma)$는 정의 8에서 제시된 비동기 순차 머신 $\Sigma$의 stable transition 행렬이다. $\Sigma$의 '정상 입력에 대한 skeleton 행렬' $S(\Sigma)$(n×n)은 다음과 같이 정의된다. $R^{(n-1)}(\Sigma)$의 $(i,j)$번째 원소가 $N$이면 $S_{ij}(\Sigma)=0$이고 $R^{(n-1)}(\Sigma)$의 $(i,j)$번째 원소가 $N$이 아니면 $S_{ij}(\Sigma)=1$이다.

상태 $x_i$에서 $x_j$로 가는 stable transition 경로가 존재하면 skeleton 행렬의 원소는 1의 값을 가지며, 경로가 존재하지 않으면 0의 값을 가진다. skeleton 행렬은 이렇게 비동기 순차 머신 도달가능성을 0, 1로 간단하게 정량화한다. 입력 외란에 대한 skeleton 행렬도 비슷하게 정의된다.

**정의 11**: $R_2(\Sigma)$는 정의 9에서 제시된 입력 외란에 대한 $\Sigma$의 one-step stable transition 행렬이다. $\Sigma$의 '입력 외란에 대한 skeleton 행렬' $S_d(\Sigma)$(n×n)은 다음과 같이 정의된다. $R_d(\Sigma)$의 $(i,j)$번째 원소가 $N$이면 $S_{dij}(\Sigma)=0$이고 $R_d(\Sigma)$의 $(i,j)$번째 원소가 $N$이 아니면 $S_{dij}(\Sigma)=1$이다.

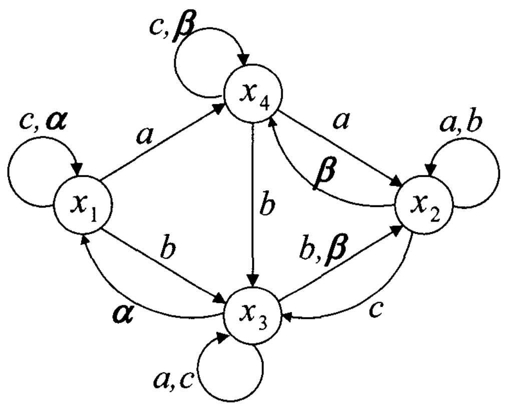
앞서 설명한 대로 입력 외란의 도달가능성은 한 번의 stable transition 경로만 찾으면 되므로 skeleton 행렬도 one-step stable transition 행렬로부터 바로 구해진다.

정의 10과 11에서 구한 skeleton 행렬을 이용하여 교정 제어의 존재 조건을 찾고 설계 과정을 구현하는 부분은 후속 논문 [8]에서 기술한다. 다음 절에서는 입력 외란이 존재하는 비동기 순차 머신의 stable state 머신을 구하고 본 장에서 새롭게 제안된 도달가능성 행렬들을 계산하는 과정

을 사례 연구를 통해 기술한다.

## 5. 예 제

그림 3(a)와 같은 유한 상태 머신으로 표시되는 비동기 순차 머신 $\Sigma$를 설정한다. 정상 입력 집합은 $A_1 = \{a, b, c\}$이고 입력 외란 집합은 $A_2 = \{\epsilon, \alpha, \beta\}$, 상태 집합은 $X = \{x_1, x_2, x_3, x_4\}$이며 초기 상태는 $x_0 = x_1$이다. $\Sigma$의 상태 천이 함수를 표로 표시하면 그림 3(b)와 같다. 그림에서 알 수 있듯이 $\Sigma$는 입력 외란 $u_2$가 아무 값을 가지지 않을 때에는($u_2 = \epsilon$) 정상적인 동작을 한다. 하지만 머신이 상태가 $x_2$나 $x_3$에서 stable combination을 가질 때에는 입력 외란이 발생하여 원하지 않는 상태 천이가 벌어질 수 있다. 한편 그림 3(a)에서 확인할 수 있듯이 상태 $x_1$과 $x_4$에서도 입력 외란 $\alpha$와 $\beta$가 각각 발생할 수 있다. 그러나 입력 외란에 의한 천이의 결과로 머신의 상태가 바뀌지 않기 때문에 외부 사용자의 관점에서 보면 머신의 동작은 외란 발생과 상관없이 정상적으로 유지된다고 말할 수 있다. 따라서 이 경우는 제어 고려 대상에서 제외시켜도 무방하다.




(a)

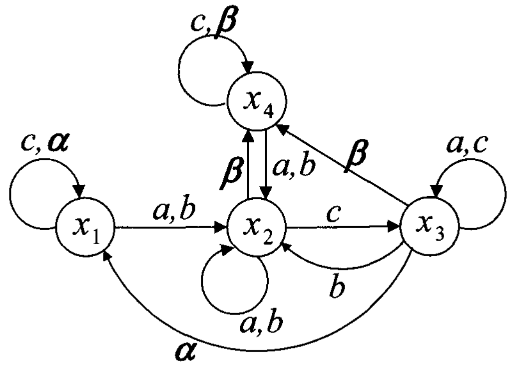
| | $(a,\epsilon)$ | $(b,\epsilon)$ | $(c,\epsilon)$ | $(-,\alpha)$ | $(-,\beta)$ |
|---|---|---|---|---|---|
| $x_1$ | $x_4$ | $x_3$ | $x_1$ | $x_1$ | • |
| $x_2$ | $x_2$ | $x_2$ | $x_3$ | • | $x_4$ |
| $x_3$ | $x_3$ | $x_2$ | $x_3$ | $x_1$ | $x_2$ |
| $x_4$ | $x_2$ | $x_3$ | $x_4$ | • | $x_4$ |

(b)

**그림 3.** 입력 외란이 존재하는 비동기 순차 머신 $\Sigma$: (a) 유한 상태 머신, (b) 상태 천이 함수 f.

**Fig. 3.** Asynchronous sequential machine $\Sigma$ with input disturbance: (a) finite state machine and (b) state transition function f.

$\Sigma$의 stable state 머신 $\Sigma_{|s}$는 그림 4와 같이 구해진다. 비동기 순차 머신의 stable state 동작을 예를 들어 설명하기 위해서 $\Sigma$가 $x_1$에서 안정 상태로 있을 때 입력 $(b,\epsilon)$이 들어 왔다고 가정하자. 그림 3(b)에서 $f(x_1,(b,\epsilon)) = x_3$이므로 $\Sigma$는 $x_3$로 상태 천이된다. 그런데 $(x_3,(b,\epsilon))$는 transient combination이므로 ($f(x_3,(b,\epsilon)) \neq x_3$) $\Sigma$는 다시 상태 천이를 해서 stable combination $(x_2,(b,\epsilon))$에 도달한다. 따라서 stable recursion 함수는 $s(x_1,(b,\epsilon)) = x_2$와 같이 계산되고 그림 4와 같은 결과가 나온다. 입력 외란이 발생했을 경우에도 유사한 과정을 거쳐서 s를 구할 수 있다.




(a)

| | $(a,\epsilon)$ | $(b,\epsilon)$ | $(c,\epsilon)$ | $(U(x_i),$ | $(U(x_i),$ |
|---|---|---|---|---|---|
| $x_1$ | $x_2$ | $x_2$ | $x_1$ | $x_1$ | • |
| $x_2$ | $x_2$ | $x_2$ | $x_3$ | • | $x_4$ |
| $x_3$ | $x_3$ | $x_2$ | $x_3$ | $x_1$ | $x_4$ |
| $x_4$ | $x_2$ | $x_2$ | $x_4$ | • | $x_4$ |

(b)

**그림 4.** stable state 머신 $\Sigma_{|s}$: (a) 유한 상태 머신, (b) stable recursion 함수 s.

**Fig. 4.** stable state machine $\Sigma_{|s}$: (a) finite state machine and (b) stable recursion function s.

다음으로는 정상 입력에 대한 skeleton 행렬과 본 논문에서 제안한 입력 외란에 대한 skeleton 행렬을 각각 구한다. skeleton 행렬을 구하려면 먼저 stable transition 행렬을 계산해야 한다. 그림 4(b)를 이용하여 정상 입력에 대한 one-step stable transition 행렬 $R(\Sigma)$을 계산하면 아래와 같다.

$$R(\Sigma) = \begin{Bmatrix} \{c\} & \{a,b\} & N & N \\ N & \{a,b\} & \{c\} & N \\ N & \{b\} & \{a,c\} & N \\ N & \{a,b\} & N & \{c\} \end{Bmatrix}$$

$|X| = 4$이므로 $R(\Sigma)$을 세 번까지 거듭 제곱해야 한다. $R^2(\Sigma)$와 $R^3(\Sigma)$을 계산하면 본 페이지의 맨 아래식과

같다. $R^2(\Sigma)$와 $R^3(\Sigma)$ 원소들의 자세한 계산 과정은 정의 5와 6을 참조하면 알 수 있다. 비동기 순차 머신 $\Sigma$의 'stable transition 행렬' $R^{(3)}(\Sigma)$은 식 (6)에 의해서 아래와 같이 구해진다. (각 원소들의 정확한 기술은 지면 관계상 생략한다.)

$$R^{(3)}(\Sigma) = R(\Sigma) \uplus R^2(\Sigma) \uplus R^3(\Sigma)$$

skeleton 행렬 $S(\Sigma)$은 $R^{(3)}(\Sigma)$의 값으로부터 쉽게 유도될 수 있다. (1,1)번째 원소의 예를 들면

$$R_{11}^{(3)}(\Sigma) = \{c\} \uplus \{cc\} \uplus \{ccc\} = \{c, cc, ccc\}$$

와 같이 계산되므로 $S_{11}(\Sigma) = 1$이다. 반면 $R_{21}^{(3)}(\Sigma) = N$이므로 $S_{21}(\Sigma) = 0$이다. 나머지 원소들의 값도 구하면 $S(\Sigma)$은 다음과 같이 된다.

위 식에서 볼 수 있듯이 skeleton 행렬은 두 상태 사이의 stable state에서의 연결성을 간단하게 도시하므로 교정 제어의 구현에 유용하게 사용될 수 있다.

다음으로 입력 외란에 대한 one-step stable transition 행렬 $R_d(\Sigma)$을 구한다. 그림 4(b)를 이용하여 $R_d(\Sigma)$을 계산하면 아래와 같다.

$$R_d(\Sigma) = \begin{Bmatrix} \{\alpha\} & N & N & N \\ N & N & N & \{\beta\} \\ \{\alpha\} & N & N & \{\beta\} \\ N & N & N & \{\beta\} \end{Bmatrix}$$

입력 외란에 의한 skeleton 행렬 $S_d(\Sigma)$은 다음과 같이 유도된다.

$$S_d(\Sigma) = \begin{Bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 \end{Bmatrix}$$

$S(\Sigma)$과 $S_d(\Sigma)$를 조사하면 입력 외란의 영향을 없애는 교정 제어기가 존재할 조건을 찾을 수 있다. 예를 들어 $S_{d31}(\Sigma) = 1$은 상태 $x_3$에서 입력 외란이 발생하여 $\Sigma$가 상태 $x_1$로 천이될 수 있음을 의미한다. 앞서 설명했듯이 교정 제어기의 역할은 $\Sigma$의 상태를 $x_1$에서 원래의 상태 $x_3$으로 되돌리는 일이다. 하지만 이 일을 성공적으로 수행하려면 $x_1$에서 $x_3$으로 연결되는 stable transition 입력 시퀀스가 존재해야 한다. 그런데 $S_{13}(\Sigma) = 1$이므로 교정 제어기의 구현이 가능함을 예상할 수 있다.

## 6. 결 론

본 논문이 제안한 연구 결과를 요약하면 아래와 같다.

- 입력 외란이 존재하는 비동기 순차 머신을 유한 상태 머신으로 모델링 하였다. 특히 입력 외란의 행동 규범을 기본 모두 동작을 만족시키도록 정의하였고 이러한 입력 외란의 영향을 없애는 모델 매칭 문제를 설정하였다. 또한 설정된 모델 매칭 문제에 맞는 상태 피드백 교정 제어기의 구조를 제안하였다.
- 비동기 순차 머신의 교정 제어에서 유용하게 사용될 도달가능성 행렬들을 입력 외란에 대해서 확장시켜 정의하였다. one-step stable transition 행렬, skeleton 행렬 등을 정의하고 그 의미를 기술하였으며 사례 연구를 통해서 계산 과정을 보였다.

본 논문에서 다루는 비동기 순차 머신의 교정 제어는 넓은 의미로 이산 사건 시스템 제어에 속한다고 말할 수 있다. 그러나 비동기 순차 머신이 가지는 안정/불안정 상태, stable combination, 기본 모드 동작 준수 같은 성질들은 이산 사건 시스템의 모델링으로는 표현할 수 없는 것들이다. 또한 본 논문에서 사용되는 상태 피드백 제어의 개념은 이산 사건 시스템 관리 제어(Supervisory Control)에서 사용하는 Generator/Accepter의 개념[7]과는 근본적으로 다르다. 본 논문에서 다루는 모델 매칭 및 교정 제어는 비동기 순차 머신으로 구성된 컴퓨터 시스템뿐만 아니라

$$R^2(\Sigma) = \begin{Bmatrix} \{cc\} & \{ca, cb, aa, ab, ba, bb\} & \{ac, bc\} & N \\ N & \{aa, ab, ba, bb\} & \{ac, bc, ca, cc\} & N \\ N & \{ba, bb, ab, cb\} & \{bc, aa, ac, ca, cc\} & N \\ N & \{aa, ab, ba, bb, ca, cb\} & \{ac, bc\} & \{cc\} \end{Bmatrix}.$$

$$R^3(\Sigma) = \begin{Bmatrix} \{ccc\} & \begin{Bmatrix} cca, ccb, caa, cab, \\ cba, cbb, aaa, aab, \\ aba, abb, baa, bab, \\ bba, bbb \end{Bmatrix} & \begin{Bmatrix} cac, cbc, aac, abc, \\ bac, bbc, aca, acc, \\ bca, bcc \end{Bmatrix} & N \\ N & \begin{Bmatrix} aaa, aab, aba, abb, \\ baa, bab, bba, bbb, \\ acb, bcb, cab, ccb \end{Bmatrix} & \begin{Bmatrix} aac, abc, bac, bbc \\ aca, acc, bca, bcc, \\ caa, cac, cca, ccc \end{Bmatrix} & N \\ N & \begin{Bmatrix} baa, bab, bba, bbb, \\ aba, abb, cba, cbb, \\ bcb, aab, acb, cab, \\ ccb \end{Bmatrix} & \begin{Bmatrix} bac, bbc, abc, cbc, \\ bca, bcc, aaa, aac \\ aca, acc, caa, cac, \\ cca, ccc \end{Bmatrix} & N \\ N & \begin{Bmatrix} aaa, aab, aba, abb, \\ baa, bab, bba, bbb, \\ caa, cab, cba, cbb, \\ acb, bcb, cca, ccb \end{Bmatrix} & \begin{Bmatrix} aac, abc, bac, bbc, \\ cac, cbc, aca, acc, \\ bca, bcc \end{Bmatrix} & \{ccc\} \end{Bmatrix}$$

ATM(Asynchronous Transfer Mode) 방식의 통신 네트워크의 트래픽 제어(traffic control) 등에서 활용될 수 있을 것이다.

본 논문에서 제안된 교정 제어기의 존재 조건 규명과 자세한 설계 과정은 후속 논문 [8]에서 기술하기로 한다.

## 참 고 문 헌


[1] G. Birtwistle and A. Davis (eds.), Asynchronous Digital Circuit Design, London: Springer-Verlag, 1995.

[2] M. Singh and S. M. Nowick, "Synthesis-for-initializability of asynchronous sequential machines," in Proceedings of International Test Conference, pp. 232-241, 1996.

[3] S. W. Moore, G. S. Taylor, P. A. Cunningham, R. D. Mullins and P. Robinson, "Self calibrating clocks for globally asynchronous locally synchronous systems," in Proceedings of IEEE International Conference on Computer Design, pp. 73-78, 2000.

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

[5] 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.

[6] 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.

[7] C. G. Cassandras and S. Lafortune, Introduction to Discrete Event Systems, Boston, MA: Kluwer, 1999.

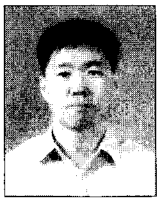
[8] 양정민, "입력 외란이 존재하는 비동기 순차 머신의 모델 매칭 제어 II: 제어기 설계," 전기학회논문지, 제57D권, 2007.

[9] 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.

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

[11] 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.


## 저 자 소 개




**양 정 민 (楊 正 敏)**

1971년 3월 31일생. 1993년 2월 한국과학기술원 전기 및 전자공학과 졸업. 1995년 2월 한국과학기술원 전기 및 전자공학과 졸업(석사). 1999년 2월 한국과학기술원 전기 및 전자공학과 졸업(공박). 1999년 3월~2001년 2월 한국전자통신연구원 컴퓨터·소프트웨어연구소 선임연구원. 2001년 3월~현재 대구가톨릭대학교 전자공학과 부교수. 주관심분야: 비동기 순차 머신 제어, 결음새 연구 등.

Tel : 053-850-2736, Fax : 053-850-2704

E-mail : jmyang@cu.ac.kr