論 文
56-9-24

입력 외란이 존재하는 비동기 순차 머신의 교정 제어 II : 제어기 설계

Corrective Control of Asynchronous Sequential Machines with Input Disturbance II : Controller Design

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 provide necessary and sufficient condition for the existence of a corrective controller that solves model matching problem of an asynchronous machine suffering from input disturbance. The existence condition can be stated in terms of a simple comparison of two skeleton matrices. The proposed controller eliminates the adversarial effect of input disturbance and makes the controlled machine mimic the behavior of a model in stable-state way. Whenever controller exists, algorithms for their design are outlined and demonstrated in a case study.

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

1. 서 론

본 논문에서는 입력 외란이 존재하는 비동기 순차 머신(Asynchronous Sequential Machine)의 교정 제어기를 설계하는 연구를 다룬다. 전역 클럭을 사용하지 않는 비동기식 설계 방법은 전력 소비와 속도 면에서 동기식(Synchronous) 설계 방법보다 우수하기 때문에 1950년대부터 주목을 받아 왔다[1]. 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 하며, 설계 후에도 머신의 동작에 대한 계속적인 테스트 과정이 수반되어야 한다[2]. 본 논문에서는 이러한 설계상의 난점을 해결하는 방안의 하나로 이미 존재하는 비동기 순차 머신의 동작을 원하는 목적에 맞게 바꾸어주는 제어기를 만드는 기법을 제안한다.

본 논문에서 사용하는 교정 제어(Corrective Control)는 Hammer가 일반적인 순차 머신의 제어를 위해서 맨 처음 도입하였으며[3],[4] 최근 레이스가 존재하는 비동기 순차 머신[5], 무한 순환(infinite cycle)이 존재하는 비동기 순차 머신[6]의 제어에 성공적으로 적용되었다. 교정 제어의 핵심은 비동기 순차 머신을 제어 대상 플랜트(plant)로 설정하여 전통적인 피드백(feedback) 제어 기법을 사용한다는 점이다. 즉 제어기는 머신의 출력을 피드백 받아서 오동작이 일어남을 감지하고 외부 입력과 플랜트 출력을 이용하여 제어 입력을 만들어낸다. 플랜트에 전역 클럭이 없으므로 교정 제어기의 동작도 비동기적으로 이루어져야 하며, 이런 점에서 비동기 순차 머신의 교정 제어는 디지털 제어(Digital Control)와도 차이점을 보인다. 기존에 존재하는 비동기 순차 머신의 동작을 보정한다는 면에서 이러한 교정 제어는 새로운 개념이라고 말할 수 있으며 다음과 같은 장점을 지닌다.

- 교정 제어기를 머신 앞에 달아 제어 입력을 생성하는 방법을 사용하기 때문에 오동작을 보이는 비동기 순차 머신을 폐기하거나 재설계할 필요가 없다.
- 레이스, 무한 순환, 입력 외란 등 재설계를 거치지 않고는 제거하기 힘들었던 비동기 순차 머신의 오동작들을 효율적으로 없애준다.
- 교정 제어기도 비동기 순차 머신으로 구성되므로 머신의 동작 속도를 그대로 유지할 수 있다.

국내에서는 비동기 순차 머신의 시험 벡터 생성에 관한 연구[7], 해저드 제거 알고리즘에 관한 연구[8] 등이 소개되었다. 하지만 이 연구들은 모두 비동기 순차 머신의 설계에 대한 것들이며, 아직까지 피드백 제어기를 이용하여 머신의 동작을 보상하는 방법에 대한 연구는 발표되지 않았다.

본 논문의 주요 목적은 입력 외란이 존재하는 비동기 순차 머신이 외란의 영향을 없애면서 정상적인 동작을 가지도록 만드는 교정 제어 문제를 푸는 것이다. 앞 논문 [9]에서는 입력 외란이 존재하는 비동기 순차 머신을 두 개의 입력

† 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

을 가지는 유한 상태 머신으로 모델링 하였다. 입력 외란의 행동 규범을 기본 모드 동작(fundamental mode operation)을 만족시키도록 정의하였고, 이러한 입력 외란의 영향을 없애는 모델 매칭 문제를 설정하였다. 또한 설정된 모델 매칭 문제에 맞는 상태 피드백 교정 제어기의 구조를 제안하였다. 본 논문에서는 앞 논문에서 제안된 교정 제어기가 존재할 구체적인 조건을 해석적으로 찾고 제어기의 설계 알고리즘을 제시한다. 앞 논문에서 정의된 도달가능성(reachability) 행렬들이 어떤 부등식을 만족시키면 교정 제어 문제를 해결할 수 있는 제어기를 찾을 수 있다. 본 논문에서 설계될 교정 제어기의 목적은 주어진 비동기 순차 머신의 동작에서 입력 외란의 영향을 완전히 없애는 일이다. 즉 제어기에서 만들어지는 적절한 제어 입력을 통해서 비동기 순차 머신은 입력 외란의 발생 여부와 상관없이 정상적인 동작을 하도록 교정된다.

본 논문의 구성은 다음과 같다. 2장에서는 [9]에서 제시된 비동기 순차 머신과 교정 제어기 구조를 기술한다. 3장에서는 비동기 순차 머신의 모델 매칭 문제를 정의하고 입력 외란이 없는 정상적인 비동기 순차 머신에 대한 모델 매칭 문제를 푼다. 4장에서는 입력 외란의 영향을 없애는 교정 제어기의 존재 조건을 제시하고 완전한 교정 제어기를 설계한다. 5장에서는 입력 외란이 존재하는 비동기 순차 머신의 예를 들어 제안된 교정 제어기를 설계하는 과정을 보여 준다. 6장에서는 본 논문의 결론을 내린다.

2. 비동기 순차 머신 및 제어기 구조

2.1 비동기 순차 머신

본 장에서는 비동기 순차 머신의 동작을 표현하기 위한 수학적 모델을 설명하고 앞 논문 [9]에서 제시된 교정 제어기 구조를 요약한다. 본 논문에서는 유한 상태 머신(Finite State Machine)을 이용하여 비동기 순차 머신을 논리적으로 모델링 한다. 논문에서 고려되는 비동기 순차 머신에 입력 외란이 존재하므로 입력 외란을 포함시킨 유한 상태 머신 식은 아래와 같이 두 개의 입력을 가진다.

$$\Sigma = (A_1 \times A_2, Y, X, x_0, f, h) \qquad (1)$$

위 식에서 A_1은 정상적인 외부 입력이 가질 수 있는 알파벳 집합이고 A_2는 입력 외란 u_2의 알파벳 집합이며 빈 스트링(empty string) ϵ를 포함한다. Y는 출력 알파벳 집합, X는 n개의 원소로 이루어진 머신의 상태 집합이며 x_0는 머신의 초기 상태이다. f는 상태 천이 함수, h는 출력 함수이며 각각 $f: X \times A_1 \times A_2 \to X$, $h: X \times A \to Y$의 매핑(mapping)을 가지며 아래와 같이 정의된다.

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

$(u_{10}, u_{20}), (u_{11}, u_{21}), (u_{12}, u_{22}), \ldots$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) k는 입력이나 상태 변수가 변경되었을 때마다 하나씩 증가한다. 본 논문에서는 비동기 순차 머신의 현재 상태가 출력값으로 나오는 입력/상태 머신(input/state machine)에 대해서 다루기로 한다. 즉 출력 알파벳 집합 Y는 X와 일치하며 출력 함수도 아래와 같이 간단하게 표시된다.

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

'valid pair' $(x, (u_1, u_2)) \in (X \times A_1 \times A_2)$는 유한 상태 머신에서 f와 h가 정의된 입력과 상태 쌍을 말한다. valid pair $(x, (u_1, u_2))$는 $x = f(x, (u_1, u_2))$일 때 '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 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 하나 이상 가지고 있는 상태를 말한다.

비동기 순차 머신에서 transient pair $(x, (u_1, u_2))$는 입력이 바뀌지 않는 한 $x_1 = f(x, (u_1, u_2))$, $x_2 = f(x_1, (u_1, u_2)), \ldots$와 같이 연쇄적으로 상태 천이를 시작하여 stable combination에 도달한다. 즉 $q \geq 1$인 자연수 q가 존재하여 $x' = f(x_q, (u_1, u_2))$이고 $x' = f(x', (u_1, u_2))$인 상태 x'가 존재한다면 $(x', (u_1, u_2))$는 비동기 순차 머신이 $(x, (u_1, u_2))$에서 시작하여 도달하는 stable combination이 된다. 이때 x'는 x로부터 'stably reachable'하다고 말한다. 만약 연쇄 천이가 끝나지 않고 transient combination 사이에서 계속 이루어진다면 $(x, (u_1, u_2))$는 무한 순환(infinite cycle)의 한 부분이 된다. 본 논문에서는 비동기 순차 머신에서 무한 순환이 존재하지 않는다고 가정한다.

비동기 순차 머신 Σ의 'stable recursion' 함수 s는 모든 valid pair에서 정의되는 함수로서 valid pair에서 시작하여 비동기 순차 머신이 다다르는 '다음 안정 상태(next stable state)'를 그 함수값으로 가진다. 즉 $s: X \times A_1 \times A_2 \to X$이고 valid pair $(x, (u_1, u_2))$에서 $s(x, (u_1, u_2)) := x'$이며 x'는 $(x, (u_1, u_2))$의 다음 안정 상태(next stable state)이다. 정의에 따라서 $(x', (u_1, u_2))$는 stable combination이 된다. f 대신 s를 이용하여 Σ의 'stable state 머신'을

$$\Sigma_{|s} = (A_1 \times A_2, Y, X, x_0, s, h)$$

이라고 정의하고 $\Sigma_{|s}$라고 명명한다. stable state 머신은 비동기 순차 머신이 실제 외부 사용자에게 보이는 모습을 나타낸다. 클럭이 없으므로 비동기 순차 머신이 불안정 상태에서 머무르는 시간은 이론적으로 0이며, 머신은 순식간에 다음 안정 상태로 천이된다. 따라서 외부에서 보면 비동기 순차 머신의 상태가 고정되는 stable combination 동작만이 유효하다고 말할 수 있다.

앞 논문 [9]에서 입력 외란 u_2는 비동기 순차 머신이 stable combination에 있을 때 발생한다고 설정하였다. u_2는 정상 입력 u_1에 상관없이 발생하므로 상태 x에서 입력 외란 u_2가 발생한다면 상태 천이 함수 f는 다음과 같이 표기될 수 있다.

$$f(x,(u_1,u_2)) := f(x,(-,u_2)),\ \forall u_1 \in U(x),\ \forall u_2 \neq \epsilon \tag{2}$$

위 식에서 $U(x)$은 x와 stable combination을 이루는 모든 정상 입력 집합을 말한다. '-'은 $U(x)$에 속한 임의의 정상 입력 u_1에 대해서 입력 외란 u_2가 정의된다는 의미이다. 외란 u_2을 포함하는 stable recursion 함수 s는 다음과 같이 정의된다.

$$s(x,(u_1^i,u_2)) := s(x,(u_1^j,u_2)),\ \forall u_1^i, u_1^j \in U(x),$$
$$s(x,(u_1,u_2)) := \text{undefined},\ \forall u_1 \not\in U(x) \tag{3}$$

식 (2)에서와 마찬가지로 입력 외란 u_2는 상태 x가 stable combination에 있을 때에만 발생 가능하므로 $u_1 \in U(x)$이어야 한다. 또 u_2는 u_1에 독립적이므로 $u_1 \in U(x)$인 모든 u_1에 대해서 동일한 stable recursion 값을 가진다.

2.2 교정 제어기 구조

앞 논문 [9]에서 제안된 비동기 순차 머신 제어 시스템 구조는 그림 1과 같다. 비동기 순차 머신 Σ는 두 개의 입력 u_1과 u_2를 가지며, 제어기 C는 상태 피드백 제어부(control unit) C와 모델 Σ'로 구성된다. 제어부 C는 역시 비동기 순환 머신으로 구현되는데, 외부 입력 v와 상태 피드백 y를 받아 제어 입력 u_1을 생성하여 Σ에 보낸다. 모델 Σ'는 외부 입력 v에 대한 모델의 현재 상태 y_r를 제어부 C에 보낸다. Σ'는 폐루프 시스템의 동작이 따라가야 할 모델이며, y_r는 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 C가 바로 제어기 C를 의미한다고 설정한다.

그림 1에서 알 수 있듯이 u_2는 제어기 C를 거치지 않고 Σ로 직접 들어가므로 제어기가 u_2의 발생을 막을 수 없다. u_2에 어떤 물리적 외란 값이 들어오고 현재 상태 x에서 제어 입력 u_1와 외란 u_2에 대한 상태 천이가 정의된다면 머신 Σ는 상태 x'로 원하지 않는 상태 천이를 겪어야 한다. 하지만 Σ가 어떤 조건을 만족한다면 입력 외란 u_2의 영향을 무력화시킬 수 있는 상태 피드백 제어기 C가 존재할 수 있고, 폐루프 시스템과 모델 Σ' 사이의 교정 문제를 풀 수 있다. 이러한 교정 제어가 가능한 가장 큰 이유는 Σ와 C가 모두 비동기 순차 머신이라는 사실이다. 제어기 C는 정상 입력 u_1의 값이 일정한 상황에서 x에서 x'로의 상태 천이를 감지하는 순간 즉시 교정 제어를 작동하여 Σ의 상태를 다시 x로 되돌려 놓는다. Σ의 상태는 순간적으로 $x \to x' \to x$의 천이 과정을 거치지만 외부 사용자에게는 상태가 x에 계속 머물러 있는 것처럼 보인다. 즉 교정된 결과는 폐루프 시스템의 stable state 머신의 동작으로 나타난다. 만약 Σ와 C가 클럭의 지배를 받는 동기 순차 머신(Synchronous Sequential Machine)이라면 이러한 교정 과정이 불가능해진다. 동기 순차 머신에서는 Σ의 상태가 x에서 x'로 천이된 것을 감지한 제어기 C가 교정 제어 입력을 만들더라도 다음 클럭의 발생 때까지는 작동될 수 없다. 하지만 다음 클럭이 발생하는 순간 제어기의 교정 제어 동작과 머신 Σ의 또 다른 상태 천이 과정이 동시에 일어나므로 원하는 교정 동작을 보장할 수 없다. 이렇듯 시스템의 비동기화 동작은 본 논문에서 제안할 교정 제어기가 존재할 가장 중요한 조건이다.

제어기 C는 상태 피드백 y값을 관측함으로써 입력 외란에 의한 상태 천이의 발생을 감지하며, 기준 입력 y_r을 이용하여 비동기 순차 머신을 원래 상태로 되돌리는 제어 입력 u_1을 만든다. 본 논문의 주제는 제어기 C가 존재할 구체적인 필요충분조건을 찾고, 조건이 만족하는 상황에서 제어기를 설계하는 과정을 보여주는 일이다.

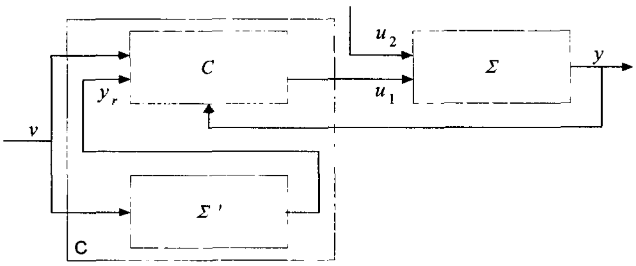


그림 1. 입력 외란이 존재하는 비동기 순차 머신의 폐루프 시스템.

Fig. 1. Closed-loop system of an asynchronous sequential machine with input disturbance.

그림 1의 폐루프 시스템이 정상적으로 동작할 또 하나의 조건은 Σ와 C가 모두 기본 모드[10], 즉 시스템 변수가 한 번에 하나씩만 바뀔 수 있다는 조건을 만족하면서 동작해야 한다는 점이다. 폐루프 시스템이 기본 모드로 동작하기 위해서 만족시켜야 할 조건은 앞 논문 [9]에서 아래와 같이 규명되었다.

정리 1: Σ와 C는 그림 1과 같이 상호 연결된 비동기 순차 머신이다. 다음 조건이 만족되면 폐루프 시스템은 기본 모드로 동작한다.

(i) Σ가 stable combination에 있지 않을 때 C의 출력 u_1과 입력 외란 u_2은 그 값이 바뀌지 않는다.

(ii) C가 stable combination에 있지 않을 때 Σ의 출력 y와 외부 입력 v, 그리고 기준 입력 y_r은 그 값이 바뀌지 않는다.

그림 1에서 C는 외부 입력 v, 상태 피드백 y, 그리고 기준 입력 y_r 등 세 개의 입력을 가진다. v는 정상 입력 알파벳 A_1의 값이며 Σ와 Σ' 모두 입력/상태 머신이므로 C의 입력 집합은 $A_1 \times X \times X$이 된다. C의 출력 집합은 Σ로 들어가는 제어 입력이므로 A_1이어야 한다. 따라서 C의 유한 상태 머신 표현은 아래와 같이 정의된다.

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

위 식에서 Ξ은 C의 상태 집합이며 ξ_0는 초기 상태, ϕ는 recursion 함수, η는 출력 함수이다. 그림 1의 폐루프 시스템을 Σ_c라고 표기하면 Σ_c는 C와 Σ로 이루어진 복합 시스템이므로 상태 집합 $X \times \Xi$, 입력 집합 $A_1 \times A_2$를 가진다. f_c와 h_c를 각각 Σ_c의 상태 천이 함수, 출력 함수라고 정의하면 폐루프 시스템의 유한 상태 머신 표현식은 아래와 같다.

$$\Sigma_c = (A_1 \times A_2, X, X \times \Xi, (x_0, \xi_0), f_c, h_c) \qquad (4)$$

Σ가 입력/상태 머신이므로 Σ_c의 출력 함수 h_c는 C의 모든 상태 $\xi \in \Xi$와 Σ_c의 모든 valid pair $(x, (u_1, u_2))$에 대해서 $h_c(\xi, x, (u_1, u_2)) := x$와 같이 되는 항등 함수(identity function)가 된다.

다음으로 비동기 순차 머신의 도달 가능성을 정량화하는 skeleton 행렬을 정의한다. 앞 논문 [9]에서는 기존 연구 [5]에서 제안되었던 정상적인 비동기 순차 머신에 대한 skeleton 행렬과 함께 입력 외란의 변화에 따른 도달가능성을 알려주는 skeleton 행렬도 정의하였다. [5],[9]를 바탕으로 skeleton 행렬 정의를 다시 쓰면 아래와 같다.

정의 1: 입력 외란이 존재하는 비동기 순차 머신 $\Sigma = (A_1 \times A_2, Y, X, x_0, f, h)$의 상태 집합을 $X = \{x_1, x_2, \cdots x_n\}$이라고 하자. Σ의 '정상 입력에 대한 skeleton 행렬' $S(\Sigma)(n \times n)$은 다음과 같이 정의된다. 상태 $s(x_i, \alpha_i) := x_j$를 만족시키는 정상 입력 스트링 $\alpha_i := (v_1, \epsilon)(v_2, \epsilon) \cdots (v_q, \epsilon) \in A_1 \times \epsilon$이 존재하면 $S(\Sigma)$의 (i,j)번째 원소는 $S_{ij}(\Sigma) = 1$이고 그렇지 않을 때에는 $S_{ij}(\Sigma) = 0$이다. Σ의 '입력 외란에 대한 skeleton 행렬' $S_d(\Sigma)(n \times n)$은 다음과 같이 정의된다. $u_1 \in U(x_i)$인 임의의 u_1에 대해서 $x_j = s(x_i, (u_1, u_2))$를 만족시키는 입력 외란 $u_2 \in A_2$이 적어도 하나 존재하면 $S_{dij}(\Sigma) = 1$이고 그렇지 않으면 $S_{dij}(\Sigma) = 0$이다.

위 정의에서 중요한 부분은 입력 외란에 대한 도달가능성을 정량화할 때에는 한 스텝(step)만을 계산해주면 된다는 사실이다. 이것은 입력 외란이 발생하여 Σ가 원하지 않는 상태 천이를 겪은 후 또 다른 입력 외란이 발생하기 전에 그림 1에서 제안된 교정 제어기가 개입을 하도록 설계되기 때문이다. $S(\Sigma)$와 $S_d(\Sigma)$의 자세한 유도 과정은 앞 논문 [9]를 참조하면 찾을 수 있다.

3. 모델 매칭 문제

본 장에서는 먼저 기존 연구 [5],[12]에서 발표된 정상적인 비동기 순차 머신의 모델 매칭 문제를 푸는 제어기를 설명한다. 본 제어기가 존재할 필요충분조건 및 설계 과정은 다음 장에서 제안될 입력 외란을 없애주는 교정 제어기를 유도하는 데 사용될 이론적 토대를 마련해준다. 입력 외란의 영향이 없다고 설정하려면 (1)의 비동기 순차 머신 식에서 항상 $u_2 = \epsilon$라고 설정해주면 된다.

비동기 순차 머신의 모델 매칭 문제는 어떤 비동기 순차 머신의 stable state 머신이 주어진 모델의 동작과 일치하도록 제어기를 설계하는 일이다[5],[11]. 즉 모델 매칭이라는 의미는 두 머신의 동작이 stable state에서 일치해야 한다는 뜻이다. 두 머신의 동작을 불안정 상태에서까지 일치시키도록 하는 일은 한 머신을 물리적으로 재설계하지 않는 이상 불가능하다. 또한 앞에서 설명했듯이 외부 사용자에게는 비동기 순차 머신의 동작 과정 중 stable state에서의 모습만이 관찰되므로 이러한 stable state에서의 모델 매칭 문제 설정은 타당하다 하겠다. 비동기 순차 머신의 stable equivalence 정의[10]과 입력/상태 머신에 대한 일반적인 모델 매칭 문제[5]를 정의하면 아래와 같다.

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

$$\Sigma = (A_1 \times A_2, Y, X, x_0, f, h)$$
$$\Sigma' = (A_1 \times A_2, Y, X, \zeta_0, f', h')$$

Σ와 Σ'가 동일한 입력과 상태 집합을 공유한다고 하고 $\Sigma_{|s}$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 개의 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.

정상적인 비동기 순차 머신의 모델 매칭 문제: 두 개의 비동기 순차 머신 $\Sigma = (A_1 \times A_2, X, X, x_0, f, h)$와 $\Sigma' = (A_1 \times A_2, X, X, x_0, f', h')$가 동일한 입력과 상태 집합을 공유한다고 하고 항상 $u_2 = \epsilon$라고 가정하자. $\Sigma_{|s}$와 $\Sigma'_{|s}$를 각각 Σ와 Σ'의 stable state 머신이라고 표기한다.

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.

위 모델 매칭 문제는 입력 외란이 없는($u_2=\epsilon$) 정상적인 비동기 순차 머신에 대해서 설정되었다. 비동기 순차 머신 Σ는 Σ'와 stable equivalence되는 것이 목적이므로 $\Sigma'=\Sigma_{|s}$, 즉 모델 Σ'가 stable state 머신으로 주어진다고 해도 일반성을 잃지 않는다[5]. 본 논문에서도 $\Sigma'=\Sigma_{|s}$라고 가정하고 제어기를 설계한다.

stable state 입력/상태 머신 $\Sigma'=(A_1\times A_2, X, X, x_0, s', h')$를 모델이라고 설정한다. Σ'는 Σ와 동일한 입력과 상태 집합을 공유한다. 위에서 정의한 모델 매칭 문제를 풀기 위해서는 아래 식을 만족하는 제어기 C를 설계해야 한다.

$$\Sigma_{c|s}=\Sigma'$$

위 식에서 $\Sigma_{c|s}$는 (4)에서 정의된 폐루프 시스템의 stable state 머신이다. $\Sigma_{c|s}$의 stable recursion 함수를 γ라고 표기하면 $\gamma: X\times \Xi\times A_1\times A_2 \to X\times \Xi$와 같은 매핑 관계를 가진다. 폐루프 시스템 $\Sigma_{c|s}$의 임의 상태 (x,ξ)에서 x만을 추출하는 투영 함수 Π_x를 아래와 같이 정의하자.

$$\Pi_x: X\times \Xi \to X$$

γ와 Π_x를 이용하여 $\Sigma_{c|s}$와 Σ'의 stable equivalence를 다시 기술하면 다음과 같다. Σ'에서 정의된 모든 valid pair $(x,v)\in X\times A_1$에 대하여 $(x,\xi,(v,\epsilon))$가 $\Sigma_{c|s}$의 valid pair가 되는 C의 상태 $\xi\in \Xi$가 존재하며 $(x,\xi,(v,\epsilon))$는 아래 식을 만족한다.

$$\Pi_x\gamma(x,\xi,(v,\epsilon))=s'(x,(v,\epsilon)) \qquad (5)$$

아래 정리는 기존 연구 [5],[6]에서 따온 것으로 비동기 순차 머신의 모델 매칭 제어기의 기본 모듈을 제공해준다. 두 집합 α와 β가 있을 때 α/β를 α 원소에서 β 원소를 뺀 차집합이라고 정의한다. 또한 $s'[S]$를 집합 $S\subset X\times A_1\times A_2$가 가지는 stable recursion 함수 s'의 치역이라고 표기한다. 입력 외란이 발생하지 않는다고 가정했기 때문에 아래 정리에서 모델 Σ'의 기준 입력 y_r은 아직까지 사용하지 않는다.

정리 2: $A_2=\{\epsilon\}$인 정상적인 비동기 순차 머신 $\Sigma=(A_1\times A_2, X, X, x_0, f, h)$가 있다. $x_1\times U_1,\ldots, x_q\times U_q\subset X\times A_1$는 서로소인 Σ의 valid pair 집합이다 (x_i는 Σ의 상태, U_i는 입력 알파벳 A_1의 부분 집합). $i=1,\ldots,q$에 대해서 x'_i를 x_i가 stably reachable할 수 있는 어떤 상태라고 하자. Σ의 폐루프 시스템 $\Sigma_{c|s}$가 다음과 같은 stable recursion 함수 s'를 가지는 모델 $\Sigma'=(A_1\times A_2, X, X, x_0, s', h')$와 stably equivalent되도록 하는 제어기 C가 존재한다.

(i) $s'[x_i\times U_i\times \epsilon]=x'_i$ for all i=1,...,q

(ii) $s'(x,(u,\epsilon))=s(x,(u,\epsilon))$ for all pairs $(x,u)\in X\times A_1/\cup_{i=1,\ldots,q}x_i\times U_i$

증명: 제어기 C의 목적은 머신의 상태가 x_i에 있을 때 $(v,\epsilon)\in U_i\times\epsilon$인 외부 입력이 들어오면 x'_i로 상태 천이를 시키는 일이다. 그림 1에서 머신이 x_i에 있을 때 $v\not\in U_i$인 입력이 들어온다면 C는 아무런 일을 하지 않고 외부 입력 v를 그대로 제어 입력 u_1으로 보낸다. v가 집합 U_i에 속한 값으로 바뀌는 순간 C는 Σ의 상태를 x_i에서 x'_i으로 천이시키는 입력 스트링을 내보낸다. 정리 2의 가정에서 x'_i는 x_i로부터 stably reachable하기 때문에 이러한 입력 스트링은 존재한다.

제어기 C의 recursion 함수와 출력 함수를 각각 ϕ와 η이라고 정의하고 $\xi\in\Xi$를 C가 가지는 임의의 상태라고 하자. 그림 1에서 C는 세 개의 입력을 가지므로 ϕ와 η를 각각 $\phi(\xi,x_r,(x,v))$, $\eta(\xi,x_r,(x,v))$라고 표기하자. $x_r(=y_r)$는 모델의 기준 입력으로 현재는 사용되지 않는다. ξ_0를 C의 초기 상태라고 정의하면 ξ_0에서 제어기가 해야 할 일은 머신의 상태가 $x_1,\ldots,x_q$ 중 하나에 속하는지를 알아서 다음 제어 동작을 하도록 준비하는 것이다. ξ_0에서 C의 동작을 아래와 같이 정의하자.

$$\phi(\xi_0,x_r,(x,v)):=\xi_0,$$
$$\forall\,(x,v)\in X\times A_1/\cup_{i=1,\ldots,q}x_i\times U(x_i)$$
$$\phi(\xi_0,x_r,(x_i,v)):=\xi_0(x),$$
$$\forall\,(x,v)\in U(x_i), i=1,\ldots,q$$

초기 상태에서 제어기 C는 외부 입력을 머신에 그대로 전달하는 역할만을 수행하므로 C의 출력 함수는 아래와 같이 정의된다.

$$\eta(\xi_0,x_r,(x,v)):=v,\ \forall\,(x,v)\in X\times A_1$$

위 설정에 따라서 제어기는 머신이 상태 $x_1,\ldots,x_q$ 중 하나와 stable combination을 이루면 초기 상태 ξ_0에서 상태 $\xi_0(x)$로 천이된다. $U(x_i)$는 x_i와 stable combination을 이루는 모든 입력 알파벳 집합이라고 앞에서 정의하였다. 상태 $\xi_0(x)$를 C의 transition 상태라고 부른다[5]. 이 상태에서 C는 외부 입력 v의 값을 보고 머신 Σ의 상태 천이 함수를 바꾸는 동작을 시작할 수 있다. 하지만 제어기는 상태 $\xi_0(x)$에서 외부 입력 값이 변하기 전까지는 Σ의 상태를 그대로 유지시켜야 한다. 즉 Σ가 상태 x_i와 stable combination을 계속 이루도록 해야 하므로 C의 출력 함수 η는 $\xi_0(x)$에서 아래와 같이 설정된다.

$$\eta(\xi_0(x), x_r, (x_i, v)) := t,\ \exists\, t \in U(x_i),$$
$$\forall\, v \in A_1,\ i = 1, \ldots, q$$

위 식에서 t는 머신의 현재 상태 x_i와 stable combination을 이루는 입력 알파벳 중의 하나로 선택된다. 이와 같이 제어 입력을 설정하면 머신의 동작은 외부 입력 v에 영향을 받지 않게 되어 차후 제어기 C가 Σ의 동작을 바꿀 수 있다.

Σ가 x_i와 stable combination을 이루고 있을 때 U_i에 속한 외부 입력 v가 들어왔다고 하자. 제어기는 머신이 현재 상태 x_i에서 x'_i로 천이할 수 있도록 입력 스트링을 만든다. 가정에 의해서 $s(x_i, \alpha_i) = x'_i$가 되는 다음과 같은 입력 스트링이 존재한다.

$$\alpha_i := (v_i^1, \epsilon)(v_i^2, \epsilon) \cdots (v_i^{m(i)}, \epsilon) \in (A \times \epsilon)^+$$

위 식에서 $m(i)$는 스트링 α_i의 길이이다. 이 스트링이 들어오면 Σ는 x_i에서 시작해서 x'_i까지 $m(i)$번의 stable transition 과정을 거친다. Σ이 거치는 중간 상태들을 아래와 같이 정의하자.

$$x_i^1 := s(x_i, (v_i^1, \epsilon)),$$
$$x_i^2 := s(x_i^1, (v_i^2, \epsilon)),$$
$$\vdots$$
$$x_i^{m(i)-1} := s(x_i^{m(i)-2}, (v_i^{m(i)-1}, \epsilon)),$$
$$x'_i := s(x_i^{m(i)-1}, (v_i^{m(i)}, \epsilon))$$

Σ가 이 상태들을 거치게 하기 위해서 제어기 C에 $\Sigma_{i=1}^q m(i)$개의 새로운 상태를 정의하고 아래와 같이 표기하자.

$$\xi^k(x_i, x'_i, U_i), \quad k = 1, \ldots, m(i),\ i = 1, \ldots, q$$

$\xi^k(x_i, x'_i, U_i) \in \Xi$는 외부 입력으로 U_i의 원소가 들어올 때 Σ의 상태를 x_i에서 x'_i으로 천이시키는 데 사용되는 제어기의 상태라는 뜻이다. C가 $\xi_0(x)$에 있을 때 U_i의 원소가 외부 입력으로 들어오면 제어기는 먼저 $\xi^1(x_i, x'_i, U_i)$으로 천이한다. 따라서 ϕ는 i=1,...,q에 대해서 다음과 같다.

$$\phi(\xi_0(x), x_r, (x_i, v)) := \xi^1(x_i, x'_i, U_i),$$
$$\forall\, v \in U_i,$$
$$\phi(\xi_0(x), x_r, (x_i, v)) := \xi_0(x),$$
$$\forall\, v \in U(x_i)/U_i,$$
$$\phi(\xi_0(x), x_r, (x_i, v)) := \xi_0,$$
$$\forall\, v \in A_1/(U(x_i) \cup U_i)$$

위 식의 두 번째 줄은 외부 입력이 U_i에 속하지는 않지만 x_i와 stable combination을 이루는 것이라면 제어기는 $\xi_0(x)$에 계속 머무른다는 의미이며, 세 번째 줄은 외부 입력이 $U_i(x)$와 U_i 어디에도 속하지 않는다면 제어기는 다시 초기 상태로 돌아와야 한다는 의미이다. $\xi^1(x_i, x'_i, U_i)$에서 제어기의 출력 함수 η는 다음과 같이 설정된다.

$$\eta(\xi^1(x_i, x'_i, U_i), x_r, (x, v)) := v_i^1,$$
$$\forall\, (x, v) \in X \times A_1$$

상태 $\xi_0(x)$에서 U_i에 속하는 외부 입력을 받고 $\xi^1(x_i, x'_i, U_i)$로 천이한 후 제어기는 $x_i \rightarrow x'_i$ 경로를 만들어주는 첫 번째 입력 v_i^1를 Σ에 준다. Σ는 v_i^1를 받으면 첫 번째 중간 상태 x_i^1으로 천이되며, 이 천이는 다시 제어기 C의 상태를 $\xi^1(x_i, x'_i, U_i)$에서 $\xi^2(x_i, x'_i, U_i)$로 바꾸어준다. 이러한 C와 Σ의 연쇄적인 반응 형태를 공식화하면 아래와 같다.

$$\phi(\xi^k(x_i, x'_i, U_i), x_r, (x_i^k, v)) := \xi^{k+1}(x_i, x'_i, U_i),$$
$$\forall\, v \in U_i,$$
$$\eta(\xi^k(x_i, x'_i, U_i), x_r, (x, v)) := v_i^k,\ \forall\, (x, v) \in X \times A_1,$$
$$k = 1, 2, \ldots, m(i) - 1,\ i = 1, 2, \ldots, q$$

위의 설정은 C와 Σ가 모두 기본 모드를 만족하면서 동작하게끔 만든다.

마지막으로 제어기가 상태 $\xi^{m(i)}(x_i, x'_i, U_i)$에 다다르면 제어 입력으로 $v_i^{m(i)}$를 출력하고 Σ는 목적 상태인 x'_i에 도달하게 된다. 다음과 같이 제어기의 동작을 설정하자.

$$\phi(\xi^{m(i)}(x_i, x'_i, U_i), x_r, (x'_i, v)) := \xi^{m(i)}(x_i, x'_i, U_i),$$
$$\forall\, v \in U_i,$$
$$\phi(\xi^{m(i)}(x_i, x'_i, U_i), x_r, (x, v)) := \xi_0(x),$$
$$\forall\, (x, v) \in \cup_{i=1,\ldots,q} x_i \times U(x_i),$$
$$\phi(\xi^{m(i)}(x_i, x'_i, U_i), x_r, (x, v)) := \xi_0,$$
$$\forall\, (x, v) \in X \times A_1/\cup_{i=1,\ldots,q} x_i \times U(x_i),$$
$$\eta(\xi^{m(i)}(x_i, x'_i, U_i), x_r, (x, v)) := v_i^{m(i)},$$
$$\forall\, (x, v) \in X \times A_1.$$

위 설정에 따라서 외부 입력 v가 U_i에 속하는 한 Σ는 stable combination $(x'_i, v_i^{m(i)})$에 계속 머무른다. $v \notin U_i$인 외부 입력이 들어오는 순간 제어기는 최종 상태 $\xi^{m(i)}(x_i, x'_i, U_i)$를 벗어나서 ξ_0 또는 $\xi_0(x)$로 천이하여 또 다른 제어 동작을 준비한다.

이러한 과정을 거치는 머신의 폐루프 시스템 Σ_{cls}의 recursion 함수 s는 조건 (i)과 (ii)를 만족시킨다고 말할 수 있다. ((i)에 해당되지 않는 valid pair를 그대로 두면 조건 (ii)가 자동적으로 만족된다.) 따라서 제어기 C의 존재가 증명되었다. □

정리 2의 증명 과정에서 설계된 제어기는 비동기 순차 머신의 교정 과정을 실증하는 중요한 모듈이다. 제어기의 동작 중 가장 주목해야 할 부분은 제어기가 입력 스트링 α_i을 머신에 전달하는 연쇄 동작이 모두 비동기적으로 이루어진다는 사실이다. 앞에서 설명했듯이 이러한 비동기적 움직임에 걸리는 시간은 이론적으로 0이므로 외부 사용자는 Σ의 최종 상태, 즉 x'_i만을 감지하게 되어 원하는 형태로의 동작 변환이 구현된다.

정상적인 비동기 순차 머신의 모델 매칭 문제를 해결하는 제어기가 존재할 필요충분조건을 정리 2의 결과와 정의 1에서 도입한 skeleton 행렬을 이용하여 표현하면 다음과 같다.

정리 3: 비동기 순차 머신 $\Sigma = (A_1 \times A_2, X, X, x_0, f, h)$와 stable state 머신 $\Sigma' = (A_1 \times A_2, X, X, x_0, s', h')$가 동일한 입력과 상태 집합을 공유한다고 하고 항상 $u_2 = \epsilon$라고 가정하자. Σ의 폐루프 시스템 $\Sigma_{c|s}$이 Σ'와 stably equivalent하는 동작을 보이도록 만드는 교정 제어기 C가 존재할 필요충분조건은 $S(\Sigma) \geq S(\Sigma')$이다. $S(\Sigma)$와 $S(\Sigma')$는 각각 Σ와 Σ'의 정상 입력에 대한 skeleton 행렬이다.

정리 3은 정리 2를 이용하여 쉽게 증명된다. Σ와 Σ'와의 모델 차이를 $x_1 \times U_1, \ldots x_q \times U_q$로 표현하면 정리 2의 결과를 바로 적용할 수 있을 것이다. 또 $S(\Sigma) \geq S(\Sigma')$인 관계가 성립하기 때문에 모델 Σ'에서 상태 x'_i가 x_i로부터 stably reachable하다면 머신 Σ에서도 stably reachable하기 때문에 모델 매칭이 가능하다(정리 2의 가정 부분). 지면 관계상 정리 3의 증명 과정은 생략한다. (기존 연구 [5] 참조)

4. 교정 제어기 설계

이번 장에서는 본 논문의 주요 목적인 입력 외란의 영향을 없애는 교정 제어기를 설계한다. 식 (2)와 (3)에서 입력 외란은 머신의 stable combination에서만 발생할 수 있다고 설정하였다. 따라서 교정 제어기를 통해서 얻어내야 할 stable equivalence는 다음과 같다. $(x, (v, \epsilon))$, $v \in U(x)$를 Σ의 임의의 stable combination이라고 하자. 입력 외란 u_2가 ϵ에서 w으로 변했을 때 아래 식을 만족시키는 제어기 C의 상태 $\xi \in \Xi$가 항상 존재한다.

$$\Pi_x \gamma(x, \xi, (v, w)) = x \tag{6}$$

위 식의 의미는 머신의 stable combination에서 입력 외란이 발생하여 원하지 않는 상태 천이가 일어나도 교정 제어기가 즉시 작동하여 머신을 원래의 상태로 되돌린다는 뜻이다. 이러한 교정 제어기가 존재하려면 입력 외란에 의해서 머신이 도달할 수 있는 상태들로부터 원래 상태가 stably reachable해야 한다. 만약 현재 상태에서 입력 외란이 발생하여 도달할 수 있는 상태가 둘 이상 있다면 그 상태들로부터 현재 상태까지 도달하는 경로가 모두 존재해야 한다. 이러한 존재 조건은 앞에서 정의한 skeleton 행렬로 표시될 수 있다. 다음 정리는 이러한 교정 제어기의 충분조건을 기술한 것이다.

정리 4: 입력 외란이 존재하는 비동기 순차 머신 $\Sigma = (A_1 \times A_2, Y, X, x_0, f, h)$가 있다. 입력 외란이 발생할 수 있는 모든 상태 집합 $Z(\subset X)$을 $Z = \{z_1, z_2, \ldots z_p\}$라고 하고 상태 z_i에서 입력 외란이 발생하여 머신이 다다를 수 있는 상태 집합을 $\chi(z_i)$라고 정의하자. 모든 i=1,...,p에 대하여 z_i가 $\chi(z_i)$로부터 stably reachable하다고 한다면 Σ의 폐루프 시스템 $\Sigma_{c|s}$가 다음과 같은 stable recursion 함수 s'를 가지는 모델 $\Sigma' = (A_1 \times A_2, X, X, x_0, s', h')$와 stably equivalent되도록 하는 제어기 C가 존재한다.

(i) $s'[z_i \times U(z_i) \times A_2] = x'_i$ for all i=1,...,q

(ii) $s'(x, (u, \epsilon)) = s(x, (u, \epsilon))$ otherwise.

증명: 조건 (i)의 의미는 상태 z_i에서 입력 외란이 발생한 것을 감지한 제어기 C는 머신이 상태 $x' \in \chi(z_i)$에서 z_i로 되돌아가도록 하는 정상 입력 스트링을 만들어내야 한다는 뜻이다. 이러한 교정 제어기의 설계 과정은 정리 2의 증명 부분과 유사하다. 정리 2와 마찬가지로 제어기 C의 recursion 함수와 출력 함수를 각각 $\phi(\xi, x_r, (x, v))$와 $\eta(\xi, x_r, (x, v))$라고 정의하고 $\xi \in \Xi$를 C의 임의의 상태라고 하자. 초기 상태 ξ_0에서 제어기는 머신이 $z_1, \ldots, z_p$ 중 하나와 stable combination을 이루는 순간 transition 상태 $\xi_0(x)$로 천이한다. ξ_0에서 C의 동작을 아래와 같이 정의하자.

$$\begin{aligned}
&\phi(\xi_0, x_r, (x, v)) := \xi_0, \\
&\quad \forall\, (x, v) \in X \times A_1 / \cup_{i=1,\ldots,p} z_i \times U(z_i) \\
&\phi(\xi_0, x_r, (z_i, v)) := \xi_0(x), \\
&\quad \forall\, (x, v) \in U(z_i), i = 1, \ldots, p \\
&\eta(\xi_0, x_r, (x, v)) := v, \forall\, (x, v) \in X \times A_1
\end{aligned} \tag{7}$$

$\xi_0(x)$에서 제어기는 머신의 상태를 계속 피드백 받으며 입력 외란이 발생했는지를 체크한다. 외부 입력 v의 값이 일정한데 상태 피드백 x에 모델 Σ'의 기준 상태 값 x'과 다른 값이 들어온다면 입력 외란이 발생했음을 의미한다. 즉 머신 Σ는 z_i에서 $\chi(z_i)$에 속하는 어떤 상태로 천이된다. 하지만 가정에서 $\chi(z_i)$의 모든 상태로부터 원래 상태 x_i까지 도달하는 정상 입력 스트링이 존재한다고 했으므로 교정 동작을 가능하게 하는 제어기를 설계할 수 있다.

x'_{ij}, $j = 1, \ldots, |\chi(z_i)|$를 $\chi(z_i)$의 원소라 하자. 가정으로부터 각 x'_{ij}에서 원래 상태 z_i까지 stably reachable

하는 정상 입력 스트링이 존재한다. x'_{ij}에 대한 입력 스트링을

$$\alpha_{ij} := (v^1_{ij}, \epsilon)(v^2_{ij}, \epsilon) \cdots (v^{m(i,j)}_{ij}, \epsilon) \in (A_1 \times \epsilon)^+$$

이라고 하면 Σ의 stable recursion 함수 s는 다음과 같은 중간 상태들을 거친다.

$$\begin{aligned} x^1_{ij} &:= s(x'_{ij}, (v^1_{ij}, \epsilon)), \\ x^2_{ij} &:= s(x^1_{ij}, (v^2_{ij}, \epsilon)), \\ &\vdots \\ x^{m(i,j)-1}_{ij} &:= s(x^{m(i,j)-2}_{ij}, (v^{m(i,j)-1}_{ij}, \epsilon)), \\ z_i &:= s(x^{m(i,j)-1}_{ij}, (v^{m(i,j)}_{ij}, \epsilon)) \end{aligned}$$

위 식에서 $m(i,j)$는 스트링 α_{ij}의 길이이다. 이와 같은 경로를 통해서 머신의 상태를 되돌리기 위하여 제어기의 새로운 상태 $\xi^k(x'_{ij}, z_i)$, $k = 1, \ldots, m(i,j)$를 정의한다. 이러한 입력 스트링과 제어기 상태는 모든 z_i에 대해서 필요하므로 제어기가 가지는 총 상태 집합은 아래와 같다.

$$\Xi := \xi_0 \cup \xi_0(x) \cup \left\{ \bigcup_{\substack{i=1,\ldots,p,\, j=1,\ldots,|\chi(z_i)| \\ k=1,\ldots,m(i,j)}} \xi^k(x'_{ij}, z_i) \right\} \tag{8}$$

transition 상태 $\xi_0(x)$에서 제어기가 x'_{ij}을 상태 피드백 값으로 받는다면 제어기는 $\xi^1(x'_{ij}, z_i)$로 상태 천이되며, 그렇지 않으면 초기 상태 ξ_0로 되돌아가거나 현재 상태를 유지한다. $\xi_0(x)$에서 C의 recursion 함수를 다음과 같이 설정한다.

$$\begin{aligned} &\phi(\xi_0(x), x_r, (x'_{ij}, v)) := \xi^1(x'_{ij}, z_i) \;\; x'_{ij} \in \chi(x_r) \\ &\phi(\xi_0(x), x_r, (z_i, v)) := \xi_0(x) \,\forall\, v \in U(z_i) \\ &\phi(\xi_0(x), x_r, (z_i, v)) := \xi_0 \,\forall\, v \in A_1/U(z_i) \\ &i = 1, \ldots, p, \;\; j = 1, \ldots, |\chi(z_i)| \end{aligned} \tag{9}$$

위 설정에서 주목해야 할 부분은 첫 번째 식에서 입력 외란에 의한 상태 천이를 감지할 때 $\chi(z_i)$ 대신 $\chi(x_r)$를 쓴다는 사실이다. 이것은 C의 transition 상태 $\xi_0(x)$에서 입력 외란에 의한 상태 천이가 발생했을 때 모델의 기준 입력 x_r를 보지 않고는 머신 Σ가 돌아가야 할 원래 상태가 어느 것인지 알 수 없기 때문이다. 제어기는 머신 Σ이 $z_1, \ldots, z_p$ 중 임의의 상태와 stable combination을 이루면 $\xi_0(x)$로 상태 천이되므로 원래 상태의 값을 저장하지 않는 한 $p \geq 2$일 때에는 그 값을 알 수가 없다. 따라서 그림 1의 교정 제어 시스템에서 모델 Σ'가 기준 상태를 제어기 C에게 알려줄 필요가 있다. 또 하나 중요한 사실은 C와 Σ가 기본 모드를 만족시키므로 변수 두 개가 동시에 변하는 상황은 고려하지 않아도 된다는 점이다. 식 (9)도 입력 외란 u_2만이 변하거나(첫 번째 식) 정상 입력 u_1만이 변하는 상황(두 번째, 세 번째 식)에 대해서만 설정되었다.

정리 3의 증명에서와 마찬가지로 C와 Σ의 연쇄적인 반응 형태를 공식화하면 아래와 같다.

$$\begin{aligned} &\phi(\xi^k(x'_{ij}, z_i), x_r, (x^k_{ij}, v)) := \xi^{k+1}(x'_{ij}, z_i), \\ &\quad \forall\, v \in U(z_i), \\ &\eta(\xi^k(x'_{ij}, z_i), x_r, (x, v)) := v^k_{ij}, \\ &\quad \forall\, (x, v) \in X \times A_1, \\ &k = 1, 2, \ldots, m(i,j) - 1, \;\; i = 1, 2, \ldots, p, \\ &j = 1, \ldots, |\chi(z_i)| \end{aligned} \tag{10}$$

위의 동작 설정은 C와 Σ가 모두 기본 모드를 만족하면서 동작하게끔 만든다. 제어기 C가 최종 상태 $\xi^{m(i,j)}(x'_{ij}, z_i)$에 도달하면 제어 입력으로 입력 스트링 α_{ij}의 최종 알파벳 $v^{m(i,j)}_{ij}$를 Σ에 보낸다. Σ는 원래 상태 z_i로 복귀하고 C는 다시 transition 상태로 천이된다.

$$\begin{aligned} &\phi(\xi^{m(i,j)}(x'_{ij}, z_i), x_r, (z_i, v)) := \xi_0(x), \\ &\quad \forall\, v \in U(z_i), \\ &\eta(\xi^{m(i,j)}(x'_{ij}, z_i), x_r, (x, v)) := v^{m(i,j)}_{ij}, \\ &\quad \forall\, (x, v) \in X \times A_1, \\ &\quad i = 1, 2, \ldots, p, j = 1, \ldots, |\chi(z_i)| \end{aligned}$$

이러한 교정 제어를 수행하면 비동기 순차 머신의 피드백 시스템 $\Sigma_{c|s}$은 조건 (i)과 (ii)를 만족시키므로 증명이 완료된다. □

위 정리의 증명에서 설계한 교정 제어기가 가지는 상태의 개수는 아래와 같이 유도된다.

$$P = 2 + \sum_{i=1}^{p} \sum_{j=1}^{|\chi(z_i)|} m(i,j) \tag{11}$$

위 식에서 p는 입력 외란이 발생할 수 있는 Σ의 상태 개수, $|\chi(z_i)|$는 상태 z_i에서 입력 외란이 발생한 후 Σ가 다다를 수 있는 다음 안정 상태 개수이며, $m(i,j)$는 $\chi(z_i)$에 속하는 상태 x'_{ij}와 z_i를 연결하는 정상 입력 스트링의 길이를 말한다.

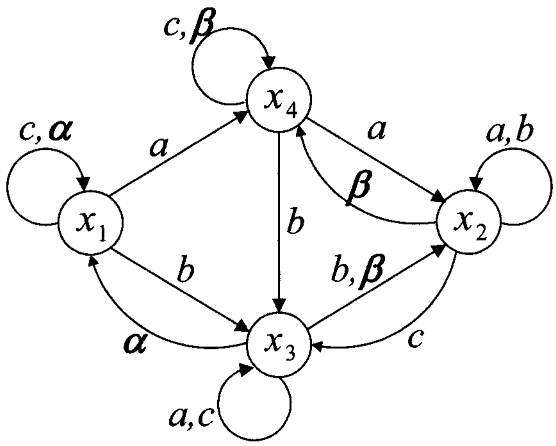
정리 4는 그림 1의 교정 제어기 C가 존재할 충분조건만을 기술하였다. 아래 정리는 교정 제어기 C의 필요조건을 추가하여 완성한 최종 정리이다.

정리 5: 입력 외란이 존재하는 비동기 순차 머신 $\Sigma = (A_1 \times A_2, Y, X, x_0, f, h)$가 있다. Σ의 폐루프 시스템 $\Sigma_{c|s}$이 입력 외란의 영향을 받지 않도록 하는 교정 제어기 C가 존재할 필요충분조건은 $S(\Sigma) \geq S^T_d(\Sigma)$이다. 이때 $S^T_d(\Sigma)$는 Σ의 외부 입력에 대한 skeleton 행렬 $S_d(\Sigma)$의 전치행렬을 말한다.

Σ의 폐루프 시스템 $\Sigma_{c|s}$이 입력 외란의 영향을 받지 않도록 하려면 stable equivalence식 (6)을 만족시키는 그림 1의 교정 제어기를 구하면 된다. 교정 제어 시스템을 구현할 때 사용되는 모델 Σ'는 비동기 순차 머신 Σ에서 u_2를 항상 $u_2 = \epsilon$로 설정하면 구해진다. 정리 4의 가정 조건이 정리 5에서 제시된 skeleton 행렬 부등식으로 표현될 수 있다는 사실을 보인다. $\chi(z_i)$는 z_i에서 입력 외란이 발생했을 때 Σ가 천이될 수 있는 다음 안정 상태 집합이라고 하였다. 정리 4의 가정은 원래 상태 z_i가 $\chi(z_i)$에 속한 모든 상태에서 stably reachable하다는 것을 의미한다. $\chi(z_i)$에 속한 임의의 상태를 x'_{ij}라고 하고 z_i와 x'_{ij}가 상태 집합 X에서 가지는 index를 각각 q와 r이라고 하자. 비동기 순차 머신 Σ에서 입력 외란에 의하여 z_i에서 x'_{ij}로 천이가 일어날 수 있으므로 정의 1에 따라서 $S_{dqr}(\Sigma) = 1$이다. 그런데 x'_{ij}에서 z_i를 연결해주는 정상 입력 스트링이 존재하므로 다시 정의 1에 따라서 $S_{rq}(\Sigma) = 1$이다. 즉 $S_{dqr}(\Sigma) = 1$인 모든 q와 r에 대해서 $S_{rq}(\Sigma) = 1$이므로 $S(\Sigma) \geq S_d^T(\Sigma)$인 관계가 성립한다. 정리 5의 필요조건에 대한 증명은 기존 연구 [5]의 결과를 이용하면 쉽게 유도되므로 여기서는 생략하기로 한다.

5. 사례 연구

사례 연구를 통해서 본 논문에서 제안한 교정 제어기의 성능을 검증하기로 한다. 그림 2(a)는 입력 외란이 존재하는 비동기 순차 머신 Σ의 유한 상태 머신이다. 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. Σ의 상태 천이 함수를 표로 표시하면 그림 2(b)와 같다. 그림에서 알 수 있듯이 Σ는 입력 외란 u_2가 아무 값을 가지지 않을 때에는($u_2 = \epsilon$) 정상적인 동작을 한다. 하지만 머신이 상태가 x_2나 x_3에서 stable combination을 가질 때에는 입력 외란이 발생하여 원하지 않는 상태 천이가 벌어질 수 있다. 한편 그림 2(a)에서 확인할 수 있듯이 상태 x_1과 x_4에서도 입력 외란 α와 β가 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 무방하므로 정리 4에서 정의한 집합 Z는 $Z = \{x_2, x_3\}$로 구해진다.



(a)

	(a,ϵ)	(b,ϵ)	(c,ϵ)	$(-,\alpha)$	$(-,\beta)$
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 2. 입력 외란이 존재하는 비동기 순차 머신 Σ: (a) 유한 상태 머신, (b) 상태 천이 함수 f.

Fig. 2. Asynchronous sequential machine Σ with input disturbance: (a) finite state machine and (b) state transition function f.

Σ의 stable state 머신 $\Sigma_{|s}$는 그림 3과 같이 유도되며 Σ의 skeleton 행렬들은 아래와 같이 구해진다. $\Sigma_{|s}$와 skeleton 행렬의 자세한 유도 과정은 앞 논문 [9]에 나와 있다.

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

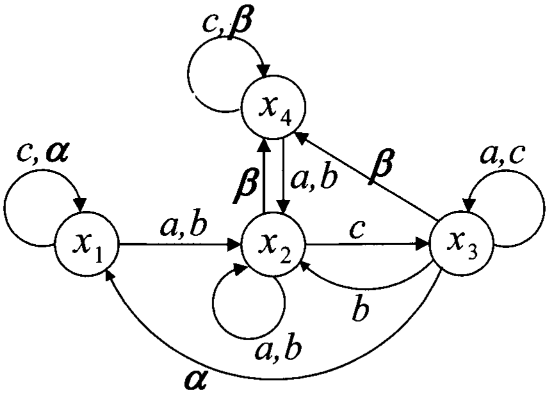
$$S_d(\Sigma) = \begin{Bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 \end{Bmatrix}$$

위 행렬들은 $S(\Sigma) \geq S_d^T(\Sigma)$ 관계를 만족시키므로 정리 5에 의해서 입력 외란의 영향을 없애는 교정 제어기 C가 존재함을 알 수 있다.

정리 4의 증명 과정에서 제시한 방법으로 제어기 C를 설계한다. 먼저 그림 3(a)의 유한 상태 머신으로부터 $U(x_2) = \{a, b\}$, $U(x_3) = \{a, c\}$를 구한다. Σ가 입력 외란 발생에 의해서 상태 집합 $Z = \{x_2, x_3\}$의 각 원소에서 도달할 수 있는 다음 안정 상태를 모두 구하면 $\chi(x_2) = \{x_4\}$, $\chi(x_3) = \{x_1, x_4\}$이다 (그림 3(b) 참

조). 또 그림 3(a)에서 Σ를 $\chi(x_2)$, $\chi(x_3)$에서 원래 상태 x_2, x_3로 되돌아가도록 하는 정상 입력 스트링을 하나씩 찾으면 아래와 같다.

$$x_4 \rightarrow x_2 : (a, \epsilon)$$
$$x_1 \rightarrow x_3 : (a, \epsilon)(c, \epsilon)$$
$$x_4 \rightarrow x_3 : (b, \epsilon)(c, \epsilon)$$



(a)

	(a,ϵ)	(b,ϵ)	(c,ϵ)	$(U(x_i),\alpha)$	$(U(x_i),\beta)$
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 3. Stable state 머신 $\Sigma_{|s}$: (a) 유한 상태 머신, (b) stable recursion 함수 s.

Fig. 3. Stable state machine $\Sigma_{|s}$: (a) finite state machine and (b) stable recursion function s.

위 정보를 가지고 교정 제어기 C의 설계를 시작한다. p=2, $|\chi(x_2)|=1$, $|\chi(x_3)|=2$이며 입력 스트링의 길이는 각각 1, 2, 2이므로 제어기의 상태 개수를 식 (11)에서 구하면 P=2+1+2+2=7이다. 또 정리 4의 증명에서 사용한 표기 방법을 그대로 이용하면 Ξ는 아래와 같이 구해진다 (식 (8) 참조).

$$\Xi = \{\xi_0, \xi_0(x), \xi^1(x_4, x_2), \xi^1(x_1, x_3), \xi^2(x_1, x_3), \xi^1(x_4, x_3), \xi^2(x_4, x_3)\}$$

초기 상태 ξ_0에서의 동작은 식 (7)을 이용하면 아래와 같이 결정된다.

$$\phi(\xi_0, x_r, (x, v)) = \xi_0,$$
$$\forall\ (x, v) \in X \times A_1 / (x_2 \times U(x_2) \cup x_3 \times U(x_3))$$
$$\phi(\xi_0, x_r, (x_2, v)) = \xi_0(x), \quad v \in \{a, b\}$$
$$\phi(\xi_0, x_r, (x_3, v)) = \xi_0(x), \quad v \in \{a, c\}$$
$$\eta(\xi_0, x_r, (x, v)) = v, \ \forall\ (x, v) \in X \times A_1$$

transition 상태 $\xi_0(x)$에서 제어기 C는 입력 외란의 발생을 감지하고 교정 제어를 위한 적절한 정상 입력 스트링을 만들어낸다. 예를 들어 Σ가 현재 x_3과 stable combination을 이루고 있을 때 입력 외란 α가 발생하여 머신이 x_4으로 천이되었다고 가정하자. 식 (9)로부터 제어기의 동작을 결정하면 아래와 같다.

$$\phi(\xi_0(x), x_3, (x_4, v)) = \xi^1(x_4, x_3), \ v \in \{a, c\}$$

앞에서도 설명했듯이 제어기가 이동해야 하는 상태는 기준 입력 x_r를 보고 결정해야 한다. x_4는 상태 x_3에서 입력 외란 α가 발생하면 얻어질 뿐 아니라 상태 x_2에서 입력 외란 β가 발생해도 얻어질 수 있다. 즉 머신이 현재 상태를 벗어나면 그 상태가 어떤 값인지 피드백만을 보고는 알 수가 없게 된다. 하지만 $x_r = x_3$이므로 제어기 C는 원래 상태가 x_3라는 사실을 알아서 $\xi^1(x_4, x_3)$으로 이동한다. 위에서 찾은 정상 입력 스트링 $(b, \epsilon)(c, \epsilon)$를 이용하여 교정 제어 동작을 완성하면 아래와 같다(식 (10), (11) 참조).

$$\phi(\xi^1(x_4, x_3), x_r, (x_2, v)) := \xi^2(x_4, x_3), \ \forall\, v \in U(x_3)$$
$$\eta(\xi^1(x_4, x_3), x_r, (x, v)) := b, \ \forall\ (x, v) \in X \times A_1$$
$$\phi(\xi^2(x_4, x_3), x_r, (x_3, v)) := \xi_0(x), \ \forall\, v \in U(x_3)$$
$$\eta(\xi^2(x_4, x_3), x_r, (x, v)) := c, \ \forall\ (x, v) \in X \times A_1$$

C가 $(b, \epsilon)(c, \epsilon)$를 출력할 때 머신 Σ가 거치는 상태 시퀀스는 $x_4 \rightarrow x_2 \rightarrow x_3$이다. 입력 외란이 발생하는 다른 상태에 대해서도 위와 유사하게 제어기를 설계하면 교정 동작이 완성된다. 앞에서도 강조했듯이 이러한 교정 제어를 가능하게 하는 요인은 머신 Σ와 제어기 C가 모두 비동기 머신라는 데 있다. 위의 예에서 제어기는 상태 피드백 값 x_4를 감지하는 즉시 교정 제어 동작을 실행하고 또 그에 따른 머신 Σ의 움직임도 비동기적으로 수행된다. 따라서 Σ가 겪는 상태 복원 경로 $x_4 \rightarrow x_2 \rightarrow x_3$에 걸리는 시간은 극히 적으므로 stable equivalence가 완성된다고 말할 수 있다.

6. 결 론

본 논문이 제안한 연구 결과를 요약하면 아래와 같다.

- 앞 논문 [9]에서 제안한 입력 외란이 존재하는 비동기 순차 머신 모델과 교정 제어 문제를 해결하는 상태 피드

백 제어기의 구조를 설명하였다.

- 교정 제어기가 존재할 필요충분조건을 제시하였으며 설계 과정을 명시하였다.
- 사례 연구를 통해서 제시한 교정 제어기의 적용 예를 보였다.

본 논문은 전통적인 자동 제어기의 틀을 바탕으로 비동기 순차 머신의 행동을 보상하는 새로운 제어 개념을 구현하였다는 데 그 의의가 있다. 본 논문의 이론적 결과를 적용하면 오동작을 보이는 기존의 비동기 순차 머신을 재설계하지 않고도 제어 목적을 구현할 수 있을 것이다. 또한 ATM(Asynchronous Transfer Mode) 방식의 통신 네트워크, 비동기 머신으로 작동되는 임베디드 시스템 등 대규모 제어 시스템 등에도 본 논문의 결과를 이용할 수 있을 것이다.

참 고 문 헌

[1] N. Venkatraman, "On the control of asynchronous machines with infinite cycles," Ph.D. Dissertation, Department of Electrical and Computer Engineering, University of Florida, USA, 1996.

[2] S. H. Unger, "Hazards, critical races, and metastability," IEEE Transactions on Computers, vol. 44, no. 6, pp. 754-768, 1995.

[3] J. Hammer, "On the control of incompletely described sequential machines," International Journal of Control, vol. 63, no. 6, pp. 1005-1028, 1996.

[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] 조경연, 이재훈, 민형복, "기본 모드에서 동작하는 비동기 순차 회로의 시험 벡터 생성," 전자공학회논문지, 제35권 C편, 제9호, pp. 716-726, 1998.

[8] 류광기, 정정화, "비동기 회로 합성을 위한 평션 해저드 제거 알고리듬," 전자공학회논문지, 제36권 C편, 제10호, pp. 47-55, 1999.

[9] 양정민, "입력 외란이 존재하는 비동기 순차 머신의 모델 매칭 제어 I: 모델링," 전기학회논문지, 제57D권, 2007.

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

[11] 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.

[12] X. J. Geng, "Model matching for asynchronous sequential machines," Ph.D. Dissertation, Department of Electrical and Computer Engineering, University of Florida, USA, 2003.

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, 걸음새 연구 등.

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

E-mail : jmyang@cu.ac.kr