이산사건모델링 및 관리제어이론

조 광 현*, 임 종 태**
*울산대학교 전기전자및자동화공학부, **한국과학기술원 전기및전자공학과

Abstract : 본 논문에서는 최근 10여 년간 세계적으로 많은 관심과 더불어 비약적인 연구발전을 이루어 온 이산사건시스템의 관리제어에 있어서 기본적인 이산사건모델링과 관리제어이론을 설명하고 최근의 연구동향 등을 소개한다. 또한 기존의 연속변수시스템과 대비해 이산사건시스템의 동적특성을 알아보고 구체적인 사례들을 살펴본다. 이러한 이산사건시스템의 동적특성을 기술하기 위해 형식언어이론에 기초한 논리적 이산사건모델을 소개하고 그 위에서 관리제어이론의 개념을 설명한다. 그리고 제어가능성 및 관측가능성의 관리제어기 존재조건을 토대로 관리제어기의 구현에 대해 알아본다. 마지막으로, 현재까지 수행되어 온 관리제어에 대한 다방면의 연구활동을 살펴보고 앞으로의 연구방향을 전망해 본다.

1. 서론

1-1. 개요

'이산사건(discrete event)'이란 용어가 '이산시간(discrete time)'과 혼동을 주곤 하던 1990년대 초와는 달리 최근 수년간 여러 제어 및 컴퓨터 관련 학술회의 등에서 별도의 분과회의(session)가 조직되어 개최되었고, 여러 IEEE Transaction등에서도 관련논문이 빈번히 발표되어 이제는 '이산사건시스템(discrete event dynamic system 또는 discrete event system)'이라는 영역이 많은 관심을 모으고 있어 어느덧 낯설지 않은 분야가 되었다. 이러한 관심과 더불어 이산사건시스템 부류의 모델링, 특성해석, 그리고 이에 바탕한 소위 '관리제어(supervisory control)' 등에 관한 다방면의 여러 연구가 이루어져 왔다. 특히, 제어분야에 있어서는 1987년 University of Toronto의 P. J. Ramadge와 W. M. Wonham이 SIAM J. of Control and Optimization에 발표한 3편의 논문[1]-[3]을 효시로 현재까지 많은 연구가 이루어져 시스템이론 측면에서의 골격을 어느 정도 갖추게 되었다. 그러나 대부분의 연구결과가 여러 저널에 파생적으로 발표되어 왔고, 이 분야를 소개하는 책 역시 현재로서는 특정분야에 치우치거나[4], [5] 다양한 부분을 소개한 가운데 일부[6]로 취급되고 있는 정도여서 이에 본 논문을 통해 이 분야의 연구를 시작하는 이들에게 분야의 소개와 이론적 토대, 응용범위, 그리고 광범위한 참고문헌 등을 제공하고자 한다.

본 논문은 P. J. Ramadge와 W. M. Wonham이 제안한 이산사건시스템의 관리제어에 관한 이론적 체제를 기초로 하여 그 후 저자를 포함한 많은 이들의 관련 연구결과들을 집대성한 것이다. 유의할 사항은 본 논문에서 소개하는 내용이 유독 제어분야에만 국한된 것이 아니라 그 응용에 따라 컴퓨터, 통신망, 집적회로, 생산시스템, 로보트 등에 걸쳐 널리 활용될 수 있다는 것이다.

본 논문의 구성은 다음과 같다. 먼저 이 절에서는 이산사건시스템의 소개와 사례들을 살펴보고, 2절에서는 이산사건시스템의 논리적 모델링 방법을 알아본다. 3절에서는 이산사건모델 상에서 관리제어의 개념을 소개하고 관리제어기의 특성을 알아본다. 마지막으로 4절에서는 이산사건시스템의 관리제어에 관한 여러 분야에서의 최근 연구동향들을 살펴보고 결론을 맺는다.

1-2. 이산사건시스템

고대의 과학자들로부터 뉴턴역학, 양자역학에 이르기까지 자연계에 존재하는 여러 물리현상 및 동적인 특성을 규명함에 있어서 기본적인 물리법칙에 근거한 미분방정식 또는 차분방정식 등은 그 동안 성공적인 역할을 해왔으며 이를 통한 시스템의 해석 및 제어를 가능케 했다. 그러나 현대에 이르러 등장하게 된 인공시스템들(man-made systems)—이를테면, 컴퓨터시스템, 통신네트워크, 생산시스템, 교통시스템 등—의 동적인 특성의 해석에 있어서는 또다른 어려움에 직면하게 되었다. 이는 그러한 시스템들의 동적특성의 변화가 일정한 물리법칙에 따라 시간의 함수로 귀결되지 않고, 불규칙한 시간간격에 비동기적(asynchronously)으로 발생된 '사건(event)'에 의해 이산(discrete)적으로 주어지는 '상태(state)' 상호간의 천이로 이루어지기 때문이다. 이와 같이 이산상태 상호간에 비동기적 사건에 의한 상태천이로 동적특성이 이루어지는 시스템 부류를 기존의 미분방정식(또는 차분방정식)에 의해 기술되는 연속변수시스템(continuous variable dynamic system)에 대비해 '이산사건시스템(discrete event dynamic system)'이라고 칭한다. 여기서 사건이란 예를 들면, 생산시스템에 있어서

일(task)의 완성 또는 기계의 고장, 통신네트워크 상의 메시지의 전송, 컴퓨터 프로그램 수행의 종료, 서보제어 시스템에 있어서 기준출력(set point)의 변경 등에 해당하며 그 특성상 일정한 시간의 함수가 아님을 알 수 있다. 그리고 이산사건시스템의 상태는 이러한 사건 발생에 의해 불규칙한 시간간격으로 천이되며 연속변수시스템에 있어서의 상태가 주어진 시점에서 시스템의 모든 과거정보(historical information)를 포함하는 수학적 의미를 지니는 반면 구체적인 물리적 상황을 표현하는 상징적 의미(symbolic meaning)를 가진다. 이를테면, 통신망의 각 노드에서 전송을 기다리는 메시지의 수, 생산시스템의 기계동작 상태(동작, 대기, 고장) 등이 이에 해당한다.

이산사건시스템과 연속변수시스템을 비교하기 위해 다음 그림 1과 그림 2의 각 시스템의 상태궤적(state trajectory)을 생각해 보자. 그림 1은 '상승'과 '하강'이라는 2개의 버턴만으로 동작되는 간략화된 3층 짜리 엘리베이터시스템에서 현재 엘리베이터의 위치를 상태변수로 하였을 때의 시간에 따른 상태궤적을 나타낸 것이다[4]. 즉, 이 엘리베이터시스템은 3개의 상태(1층, 2층, 3층)와 2개의 사건(상승, 하강)으로 표현되며 엘리베이터의 상태는 불규칙한 시간 (t_1, t_2, t_3, t_4)에 발생되는 사건에 의해 진행되는 구간별 상수(piece-wise constant) 형태의 궤적을 가진다. 이 때 각 상태는 주어진 이산집합(discrete set)내의 한 값을 가지며 전체 상태의 값은 시간에 대해 구간별로 연속적임을 알 수 있다. 반면, 그림 2에 나타나 있는 전형적인 연속변수시스템의 상태궤적을 보면 상태변수는 R^n 상의 임의의 값을 취하며 연속적으로 변화하는 입력(u)과 현재 상태(x)에 따라 연속적으로 변화됨을 알 수 있다[7].

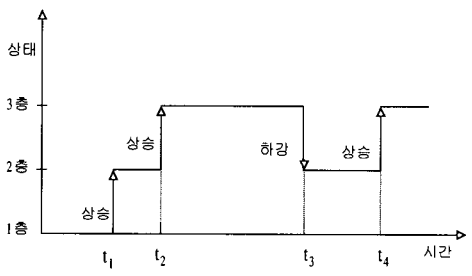

그림1. 이산사건시스템의 상태궤적의 예[4].

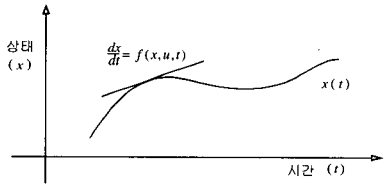

그림 2. 연속변수시스템의 상태궤적의 예[7].

이산사건시스템과 연속변수시스템의 특징을 비교·정리하면 표 1과 같다.

표 1. 이산사건시스템과 연속변수시스템의 비교.

	연속변수시스템	이산사건시스템
모델	미분방정식(차분방정식)	상태천이 규칙에 기반
상태변화	시간에 대해 연속적	시간에 대해 불연속적
동적변화	시간의 함수	사건발생의 함수
연구상황	수세기동안 이루어짐	1980년대 후반부터 주목되어짐
예	모터서보제어시스템, 연속화학공정시스템, 비행체 서보제어시스템 등	생산시스템, 컴퓨터시스템, 통신망, 교통시스템 등

이산사건시스템에 대한 연구는, 주로 인공시스템들을 대상으로 동작원리(rules of operation)를 밝혀내고 의사분석(decision analysis), 수학적 프로그래밍, Markov이론, 대기이론(queueing theory)등을 이용하는 산업공학의 OR(Operations Research)영역과 공통부분을 가지며, 동역학적 특성을 분석해 기존의 연속변수시스템에 있어서의 제어가능성(controllability), 관측가능성(observability)등의 개념을 확대 적용함으로써 시스템의 원하는 특성을 얻어내기 위한 제어이론을 도입할 수 있다는 점에서 제어 및 시스템이론 영역과 공통점을 지니고, 또한 대상시스템이 주로 인공시스템이므로 이를 운용하는 인간(human operator)과의 인터페이스를 위한 컴퓨터과학(computer science)의 인공지능(AI) 및 자연어처리(natural language processing) 연구영역과도 공통영역을 가지는 특성이 있다. 이러한 특징을 도표로 표현하면 그림 3과 같다[7]-[9].

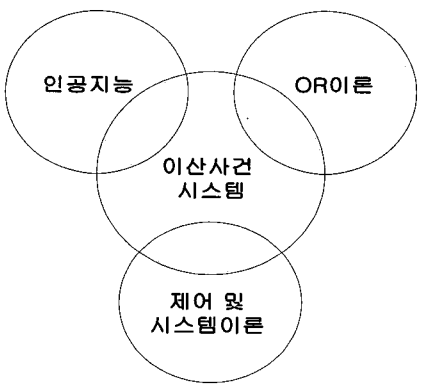

그림 3. 이산사건시스템의 연구영역[7].

1-3. 이산사건시스템의 사례

1-3-1. 생산시스템

다음 그림 4와 같이 직렬로 연결된 2대의 기계(M1, M2)와 사이에 놓여진 1개의 버퍼(B)로 이루어진 간략화된 생산시스템을 생각해보자[1].

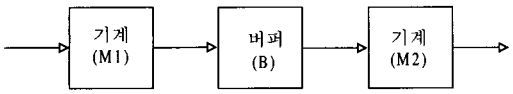

그림 4. 생산시스템의 예 : 2대의 기계와 1개의 버퍼[1].

이 때 각각의 기계(M1, M2)는 '대기(idle)', '동작(working)', '고장(breakdown)'의 3가지 상태를 가지며 동적특성의 변화는 사건에 의한 이들 상호간의 천이로 규명될 수 있다. 버퍼(B)의 경우는 각 슬롯(slot)마다 '비어있음(empty)', '차있음(full)'의 상태를 가지며 이들 상호간의 천이는 앞 뒤 기계에서 발생되는 사건에 의해 수동적으로 이루어짐을 알 수 있다. 즉, 전체시스템은 그 동적인 변화가 각 기계의 동작조건, 가공물품의 종류, 기계의 상태 등에 따라 비동기적으로 발생되는 사건에 의한 상태천이로 표현되는 이산사건시스템이다.

1-3-2. 공조시스템(HVAC System)

최근 지능형 빌딩(intelligent building)등에서 필수적으로 요구되는 공조시스템(HVAC: Heating, Ventilation, and Air Conditioning System)은 냉·난방 및 환기기능을 갖춘 것으로 그림 5에서와 같이 보일러, 펌프, 밸브, 밸브제어기, 팬 등으로 구성되어 있으며 이들 각 부분들은 '동작', '정지', '열림', '닫힘' 등의 이산상태 집합 내에서 동작모드의 사건발생에 따라 천이가 이루어지는 이산사건시스템이다[10].

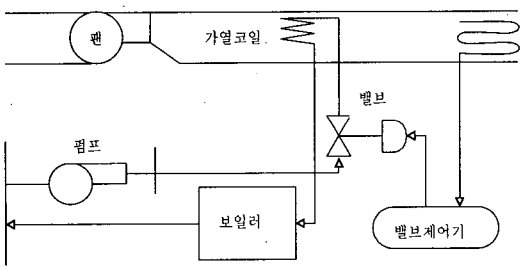

그림 5. 공조시스템의 예[10].

1-3-3. 통신망

그림 6의 간략화된 통신망의 프로토콜을 생각해보자[11].

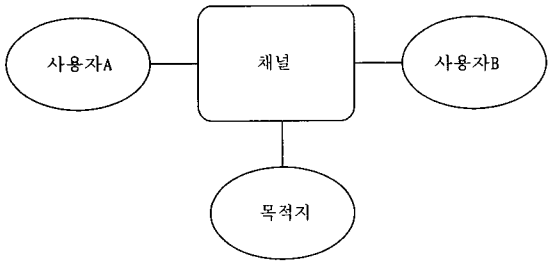

그림 6. 통신망 : 2명의 사용자와 1개의 채널[11].

이 경우 2명의 사용자는 1개의 채널을 공유하며 목적지(destination)로 메시지를 전송한다. 이 때 채널은 한번에 1개의 메시지만을 전송할 수 있으므로 '대기(idle)', '전송(transmitting)', '충돌(collision)'의 3가지 상태를 가지며, 각각의 사용자는 '대기(idle)', '전송(transmitting)', '전송대기(waiting to transmit)'의 상태를 각 사용자의 상황에 따라 비동기적으로 천이하고 그 결과에 따라 채널의 상태천이를 유발하는 이산사건시스템이다.

1-3-4. 전자회로 시스템

일반적인 전자회로의 논리적 동작특성 또는 송전선로의 보호계전시스템도 이산사건시스템으로 대변될 수 있다[12], [13]. 예를 들어 그림 7의 시지연(delay)을 포함한 NAND게이트의 입출력을 생각해보자[12].

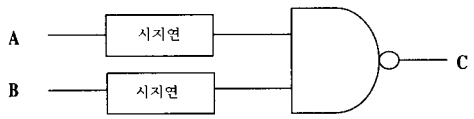

그림 7. 시지연을 포함한 NAND게이트의 입·출력[12].

그림 7에서 보는 바와 같이 시지연에 의해 2개의 입력 A, B가 NAND게이트에 도달하는 순서의 차이가 출력값 C의 변화를 유발하게 된다. 입출력 값을 상태변수로 보았을 때 이러한 시지연을 고려하면 각 신호(signal)의 0으로부터 1 또는 1로부터 0으로의 천이를 유발하는 6가지 사건에 의해 7가지 서로 다른 상태로 천이 되는 이산사건시스템임을 알 수 있다. 일반적인 논리회로는 이러한 NAND게이트의 조합으로 표현 가능하므로 이런 개념의 확장을 통해 이산사건시스템으로 볼 수 있다.

2. 논리적 이산사건모델

이산사건시스템의 동적인 특성을 모델링하기 위한 이산사건모델(discrete event model)의 종류에는 대상 시스템의 특성과 모델링의 목적에 따라 논리적 이산사건모델(logical discrete event model), 시간 이산사건모델(timed discrete event model), 비결정적 이산사건모델(nondeterministic discrete event model), 추계적 이산사건모델(stochastic discrete event model) 등이 있다[6], [9], [16]. 본 논문에서는 이 가운데 대상시스템의 정성적인(qualitative) 동적특성을 잘 반영할 수 있는 논리적 이산사건모델을 이용해 대상 시스템을 모델링하고 동적특성을 기술하는 방법에 대해 다룬다.

2-1. 형식언어(Formal language)

일반적으로 이산사건시스템의 정성적 또는 논리적인 측면에서의 동적특성은 앞 절에서 소개한 상태와 사건을 그 발생순서에 따라 모두 나열함으로써 기술할 수

있다. 더욱이, 대상 이산사건시스템이 주어진 상태에서 어떤 사건에 의해 천이되는 다음 상태가 항상 유일하게 결정되는 결정적(deterministic) 이산사건시스템일 경우에는 초기 상태로부터 발생 가능한 모든 사건열(sequence)들만의 나열을 통해 기술이 가능하다. Σ를 대상 이산사건시스템에서 발생 가능한 사건들의 유한 집합이라고 하고 $\sigma_i \in \Sigma$, i∈[1, n]를 각각의 사건이라고 하자. 그러면 $S = \sigma_1 \sigma_2 \cdots$ 와 같은 사건열을 그 시스템의 자국(trace) 또는 문자열(string)이라고 부르며 그러한 자국들의 모음(collection)을 형식언어(formal language) 또는 언어(language)라고 한다[14], [15]. Σ^*를 길이 0의 빈 문자열(null string) ε을 포함한 Σ의 사건들로 구성된 모든 유한 길이의 자국들의 집합이라고 하자[이를 Σ의 클린네닫힘(Kleene closure)이라고 한다]. 그러면 언어는 Σ^*의 부분집합이 된다. 흔히 L, K, H 등이 이러한 언어를 나타내는 부호(symbol)로 쓰인다. 주어진 문자열 s∈Σ^*에 대해 |s|는 string의 길이(구성하는 사건의 갯수)를 나타내는 기수(cardinality) 부호이며 정의로부터 $|\varepsilon| = 0$ 이다. 언어는 물리적으로 대상시스템의 동작순서(operating sequence), 상태변화, 일의 진행 등을 나타낸다. 다음은 복잡한 이산사건시스템의 동적인 특성을 이러한 언어를 이용해 단순하게 표현할 수 있는 연산(operation)들과 성질 등을 소개한다[4], [6], [15].

- 언어의 합집합(union 또는 choice) :
  $K_1, K_2 \subseteq \Sigma^*$라고 하자. 그러면 $K_1 \cup K_2 := \{s \in \Sigma^* \mid s \in K_1$ 또는 $s \in K_2\} = K_1 + K_2$로 정의된다.
- 언어의 교집합(intersection) :
  $K_1 \cap K_2 := \{s \in \Sigma^* \mid s \in K_1$ 그리고 $s \in K_2\}$로 정의된다.
- 언어의 차집합(difference) :
  $K_1 - K_2 := \{s \in \Sigma^* \mid s \in K_1$ 그리고 $s \notin K_2\}$로 정의된다.
- 언어의 연결(concatenation) :
  $K_1K_2 := \{st \in \Sigma^* \mid s \in K_1$ 그리고 $t \in K_2\}$로 정의된다.
- 언어의 여집합(complement) :
  $K \subseteq \Sigma^*$라고 하자. 그러면 $K^c \subseteq \Sigma^*$는 $K^c := \Sigma^* - K$로 정의된다.
- 언어의 접두닫힘(prefix closure)[또는 닫힘(closure)] :
  $\overline{K} := \{s \in \Sigma^* \mid (\exists t \in \Sigma^*)\ st \in K\}$로 정의된다.
- 언어의 클린네닫힘 :
  $K^* := \bigcup_{n \in N} K^n$ ($K^0 := \{\varepsilon\}$, N은 0을 포함한 자연수 집합, $K^{n+1} = K^nK$) 으로 정의된다. 이때, $(K^*)^* = K^*$, $\phi^* = \{\varepsilon\}$, $\varepsilon^* = \{\varepsilon\}$의 멱등원(idempotent) 관계가 있다.
- 언어의 몫(quotient) :
  K_1의 K_2에 대한 몫은 $K_1/K_2 := \{s \in \Sigma^* \mid \exists t \in K_2$ 다음과 같은 $st \in K_1\}$으로 정의된다.
- 언어의 후언어(post-language) :
  K_1의 K_2에 대한 후언어는 $K_1 \setminus K_2 := \{ s \in \Sigma^* \mid \exists t \in K_2$ 다음과 같은 $ts \in K_1\}$으로 정의된다.
- 언어의 정닫힘(positive closure) :
  K의 정닫힘은 $K^+ = KK^*$로 정의된다. 즉, $\varepsilon \in K$ 이면 $K^+ = K^*$ 임을 알 수 있다.
- 언어의 비충돌성 (nonconflicting) :
  K_1과 K_2가 $\overline{K_1 \cap K_2} = \overline{K_1} \cap \overline{K_2}$을 만족시키면 비충돌적이라고 한다. 예를 들어 $K_1 = \{ab\}$, $K_2 = \{a\}$이면 K_1과 K_2는 비충돌적임을 알 수 있다.
- 언어의 닫힘성(closedness) :
  K가 $\overline{K} \cap M = K$를 만족시키면 M-닫힘성(M-closed)이 있다고 한다. 예를 들어 $K=\{aab\}$, $M=\{aab, aabb\}$ 이면 K는 M-닫힘성이 있음을 알 수 있다.

2-2 오토마타(Automata)와 유한상태기계(FSM: Finite State Machine)

앞 절에서 소개한 언어 모델의 경우는 Σ가 유한집합이더라도 언어는 유한 또는 무한집합일 수 있으므로 이를 각각 원소나열법, 조건제시법 등의 집합이론을 통해 표현할 수 있다. 이러한 언어 모델의 또 다른 표현방식으로 이 절에서는 결정적 유한 오토마타(deterministic finite automata)—또는 간략히 오토마타—와 FSM(finite state machine)을 소개한다[1], [4], [15]. 오토마톤(automaton) A는 다음과 같은 5개의 원소로 구성된 순서쌍(tuple)이다.

$$A := (Q, \Sigma, \delta, q_0, Q_m)$$

여기서 Q는 A의 유한상태집합을 나타내며 Σ는 A의 상태상호간의 천이와 관계된 사건집합을 나타낸다. $\delta : Q \times \Sigma \mapsto Q$는 A의 전체천이함수(total transition function)이다. 이를테면, 상태 q_1으로부터 사건 σ에 의한 상태 q_2로의 천이는 $\delta(q_1, \sigma) = q_2$로 나타내어진다. 이때 δ가 함수이므로 주어진 오토마톤은 결정적(deterministic) 오토마톤이라고 불리며 또한 각각의 상태 $q \in Q$ 에서 Σ 내의 모든 사건에 해당하는 천이가 정의되어 있으므로 δ는 전체함수라고 한다. 그리고 q_0는 A의 초기상태를 나타내며 $Q_m \subseteq Q$ 는 A의 표기상태(marked states) 집합으로서 물리적으로 어떠한 일의 완성, 작업종료상태 등을 의미한다. 한편, 위의 상태천이함수 $\delta(\cdot, \cdot)$는 사건 $\sigma \in \Sigma$ 에 의한 상태천이로부터 사건열 s∈Σ^*에 의한 상태천이로 다음과 같이 확대 정의될 수 있다. 즉, 모든 $s \in s^*$, $\sigma \in \Sigma^*$에 대해 $\delta : Q \times \Sigma^* \mapsto Q$로 확장된다. 이러한 확장된 개념 위에서 오토마톤 A에 의해 생성된 언어 L(A)는

$$L(A) := \{s \in \Sigma^* \mid \delta(q_0, s)!\}$$

로 정의되며 (이때 $\delta(q_0, s)!$는 $\delta(q_0, s)$가 A에서 정의되어있음을 의미한다), 이 가운데 A에 의한 표기언어(marked language) L_m(A)는

$$L_m(A) := \{s \in L(A) \mid \delta(q_0, s) \in q_m\}$$

로 정의된다. 그러면 L(A)는 항상 접두닫힌 언어이며 오

토마톤의 경우 $L(A) = \Sigma^*$임을 알 수 있다.

2-3 FSM의 접근가능성(accessibility)과 상호접근가능성(coaccessibility)

주어진 FSM G의 한 상태 $q \in Q$로부터 도달 가능한(reachable) 상태집합 $Re_G(q)$는 $Re_G(q) := \{q' \in Q \mid \exists s \in \Sigma^*$ 다음과 같은 $\delta(q, s) = q'\}$로 정의된다. 만일 $Re_G(q) = Q$이면 G는 접근가능(accessible)이라고 하며, G는 모든 상태가 초기상태로부터 도달 가능함을 의미한다. 일반적으로 G의 접근가능한 부분(part)은 다음과 같이 접근가능하지 않는 상태를 모두 배제시킴으로써 구해낼 수 있다.

$$A_c(G)=(Q_{ac},\ \Sigma,\ \delta_{ac},\ q_o,\ Q_{ac,m}).$$

여기서 $Q_{ac}=\{q\in Q \mid (\exists w\in\Sigma^*)\ \ \delta(qo, w)=q\}$, $Q_{ac,m}= Q_m \cap Q_{ac}$, $\delta_{ac} = \delta \mid_{Qac\times\Sigma}$이며 $L(G)=L(A_c(G))$, $L_m(G)=L_m(A_c(G))$임을 알 수 있다[1].

한편, G의 각 상태로부터 적어도 한 개 이상의 표기상태가 도달 가능하면, 즉, 모든 $q\in Q$ 에 대해 $Re_G(q) \cap Q_m \neq \emptyset$ 일 때 G는 상호접근가능(coaccseeible)이라고 한다. 따라서 만일 모든 G가 접근가능이면, $\overline{L_m}(G) = L(G)$ 일 경우 G는 상호접근가능이 되며 그 역도 성립된다. 일반적으로 G의 상호접근가능한 부분은 다음과 같이 얻어진다.

$$C_{OAC}(G)=(Q_{coac}\ ,\ \Sigma\ ,\ \delta_{coac}\ ,\ q_{coac}\ ,\ Q_m).$$

여기서 $Q_{coac} = \{q \in Q \mid (\exists W \in \Sigma^*)\ \delta(q, w)\in Q_m\}$, 만약 $q_o \in Q_{coac}$ 이면 (아니면 정의되지 않은 상태로 남겨짐) $q_{coac} = q_o$, $\delta_{coac} = \delta \mid_{Qcoac\times\Sigma}$ 이다. 한편 G가 생성하는 언어가 $L(G) \neq \overline{L_m}(G)$ 이면 막힘(blocking)이라고 하는데 위의 정의를 따르면 비막힘(nonblocking)과 상호접근가능성(coaccessibility)은 동일한 의미를 지님을 알 수 있다. G가 접근가능이고 또한 상호접근가능이면 트림(trim)이라고 한다. 즉, $Trim(G) := C_{OAC}(A_c(G)) := A_c(C_{OAC}(G))$으로 정의한다.

2-4 정규언어(Regular language)

기본적으로 Σ^*가 무한집합이기 때문에 앞서 소개된 언어 또한 무한집합임을 알 수 있다. 반면 대상시스템의 동작특성을 모델링하고 분석하기 위해선 무한집합을 유한한 해석적(analytical)방법으로 다루길 원한다. 이러한 측면에서 오토마타와 FSM 등으로 무한 집합인 언어를 표현하는 것이 유용하리라 생각되어질 것이다. 그러나 항상 모든 언어를 FSM으로 동등하게 표현 할 수 있는 것은 아니다. 이 절에서는 그러한 표현 속성을 지니는 언어의 부분집합인 정규언어(regular language)에 대해 다룬다[4], [6], [15].

앞서 이산사건시스템의 동작을 발생 가능한 사건열의 나열을 통해 나타낸 것이 언어라 하였고, 이러한 언어를 생성하는 장치(device)로서 FSM을 소개하였다. 이를 다른 관점에서 볼 때, 즉, 언어가 먼저 주어지고 이와 동일한 언어를 생성하는 FSM을 그 언어의 인식기(recognizer)라고 한다. 정규언어의 정의는 문헌에 따라 조금씩 다르나 가장 직관적인 정의는 다음과 같다. 주어진 언어 K는 이를 표기언어로 인식하는 FSM G가 존재할 때, 즉 $L_m(G) = K$인 G가 존재하면 정규언어 L_R 이라고 한다($K \in L_R$). K가 정규언어이면, K를 인식하는 FSM은 유일하지 않다. 따라서 동일한 K를 인식하는 여러 FSM들을 생각 할 수 있다. $\| K \|$를, K를 인식하는 FSM G 가운데 $|Q_G|$의 최소값 이라고 할 때, 이러한 최소값을 가지는 인식기를 K의 표준인식기(canonical recognizer)라고 한다.

정규언어에 대한 또 다른 정의는 정규표현(regular expression)이라는 대수적(algebraic) 관계식에 의한 것이다. 즉, 정규언어 $L_R \subseteq 2^{\Sigma^*}$는 다음 3가지 조건식을 만족시키는 집합이다.

1) $\emptyset, \{\varepsilon\} \in L_R$,

2) $\forall \sigma \in \Sigma : \{\sigma\} \in L_R$,

3) $K, K_1, K_2 \in L_R \Rightarrow K_1 + K_2, K_1K_2, K^* \in L_R$.

정규언어는 위의 두 번째 정의로부터 합집합(union), 연결(concatenation), 그리고 클린네닫힘에 대해 닫혀 있음을 알 수 있다. 뿐만 아니라, 정규언어는 교집합(intersection)과 여집합 (complementation)등의 불연산(boolean operation)에 대해서도 닫혀 있음을 그 정의로부터 알 수 있다.

정규언어에 대한 정의들로부터 주어진 어떤 언어가 정규언어인지 판단하기 위해선 이를 인식하는 FSM이 존재하는지 또는 정규표현으로 나타낼 수 있는지를 검토하면 된다. 그럼에도 불구하고, 주어진 언어가 정규언어인지를 판단하기는 쉽지 않다. 이러한 판단을 돕기 위해 다음의 예비명제를 소개한다.

예비명제 2.1 (Pumping Lemma)[4]: $K\subseteq\Sigma^*$가 정규언어이면 어떤 $m \in N$이 존재해서 모든 $s \in K(|s| \geq m)$에 대해 $s=uvw$, $|uv| \leq m$, $|v| \geq 1$이고 각각의 $i \geq 0$에 대해 $uv^iw \in K$인 $u,v,w \in \Sigma^*$가 존재한다. 더욱이 m은 $m \leq \| K \|$을 만족시키도록 선택되어 질 수 있다.

[예] $K := \{a^ib^i \mid i \geq 0\}$, $\Sigma = \{a,b\}$를 생각해 보자. 그러면 우리는 위 예비명제 2.1로부터 K가 정규언어가 아님을 알 수 있다. 즉, $s=a^mb^m \in K$, $m \in N$를 생각해보자. 그러면 $|s|=2m \geq m$이고 명제 2.1에 따라 $uvw=s=a^mb^m$, $|uv| \leq m$, $|v| \geq 1$인 $u,v,w \in \Sigma^*$가 존재하고 각각의 $i \geq 0$ 에 대해 $uv^iw \in K$이어야 한다. $|uv| \leq m$이므로 $uv \leq a^m$이고, 따라서 $u=a^j, v=a^K$이어야 한다. 그러면 $w=a^{m-(j+K)}b^m$ 이 된다. $i=2m$이라고 할 때, $uv^mw=a^j(a^K)^{2m}a^{m-(j+K)}b^m = a^{m+2mK-K}b^m$ 이므로 모순이 된다. 따라서 K는 정규언어가 아님을 알 수 있다.

2-5. 곱(Product)과 병렬합성(Parallel composition)

일반적으로 대상 이산사건시스템의 이산사건모델을 구하기 위해서는 먼저 시스템의 각 구성요소들에 대한 부분 이산사건모델들(component discrete event models)을 구한 뒤 각각의 동작조건을 고려하여 합성을 통해 시스템 전체에 대한 이산사건모델을 구한다[1], [16]. 이 때 관건은 동기화된(synchronous)사건과 비동기화된(asynchronous)사건에 의한 전체시스템의 상태변화를 어떻게 고려할 것인가에 있다.

먼저 동기화된 사건에 의한 상태천이만을 고려한 곱(product)연산에 따른 합성을 소개한다. 다음의 두 FSM G_1,G_2를 생각해 보자.

$$G_1=(Q_1,\ \Sigma_1,\ \delta_1,\ q_{01},\ Q_{m1}),$$
$$G_2=(Q_2,\ \Sigma_2,\ \delta_2,\ q_{02},\ Q_{m2}).$$

이 때 두 FSM들의 곱 $G_1 \times G_2$는 다음과 같이 정의된다.
$G_1 \times G_2 := A_c(Q_1 \times Q_2,\ \Sigma_1 \cap \Sigma_2,\ \delta,\ (q_{01}, q_{02}),\ Q_{m1} \times Q_{m2})$,
여기서, $\delta((q_1, q_2), \sigma)$는

1) 만일 $\sigma \in \Sigma_{G1}(q_1) \cap \Sigma_{G2}(q_2)$이면 $(\delta_1(q_1, \sigma), \delta_2(q_2, \sigma))$이고
2) 그 외의 경우는 정의되지 않는다.

위 정의로부터 $L(G_1 \times G_2)=L(G_1) \cap L(G_2)$, $L_m(G_1 \times G_2)=L_m(G_1) \cap L_m(G_2)$임을 알 수 있다. 이와 같은 곱연산에 따른 합성을 동기합성(synchronous composiotion)이라고도 한다.

다음은 비동기화된 사건에 의한 상태천이까지 고려한 보다 일반적인 병렬합성(parallel composition)을 생각해보자. 앞서와 같이 두 FSM G_1, G_2가 주어졌다고 하자. 그러면 G_1, G_2의 병렬합성 $G_1 \| G_2$는 다음과 같이 정의된다.
$G_1 \| G_2 := A_c(Q_1 \times Q_2,\ \Sigma_1 \cup \Sigma_2,\ \delta,\ (q_{01}, q_{02}),\ Q_{m1} \times Q_{m2})$,
여기서 $\delta((q_1,q_2), \sigma)$는

1) 만일 $\sigma \in \Sigma_{G1}(q_1) \cap \Sigma_{G2}(q_2)$이면 $(\delta_1(q_1, \sigma), \delta_2(q_2, \sigma))$,
2) 만일 $\sigma \in \Sigma_{G1}(q_1)$-$\Sigma_{G1}(q_2)$이면 $(\delta_1(q_1, \sigma), q_2)$,
3) 만일 $\sigma \in \Sigma_{G2}(q_2)$-$\Sigma_{G1}(q_1)$이면 $(q_1, \delta_2(q_2, \sigma))$,
4) 그 외의 경우는 정의되지 않는다.

병렬합성 모델의 사건집합으로부터 각 부분 모델의 사건집합으로의 투영(projection) 사상(map)을 $P_i := (\Sigma_1 \cup \Sigma_2)^* \mapsto \Sigma_i^*$, i=1,2로 정의하자. 그러면 $P_i(\cdot)$로부터 $P_i(\varepsilon)=\varepsilon$, $P_i(\sigma)$는 만일 $\sigma \in \Sigma_i$이면 σ이고, 그렇지 않으면 ε임을 알 수 있으며 $s \in (\Sigma_1 \cup \Sigma_2)^*$, $\sigma \in (\Sigma_1 \cup \Sigma_2)$에 대해 $P_i(s\sigma) = P_i(s)P_i(\sigma)$로 확대 정의됨을 알 수 있다. 더욱이, 역투영(inverse projection)은 $P_i^{-1}: \Sigma_i^* \mapsto (\Sigma_1 \cup \Sigma_2)^*$ 즉, $P_i^{-1}(t)=\{s \in (\Sigma_1 \cup \Sigma_2)^* \mid P_i(s) = t\}$로 정의된다. 이러한 투영사상과 병렬합성연산에 따른 성질들(properties)을 정리하면 다음과 같다.

- $L(G_1 \| G_2) = P_1^{-1}(L(G_1)) \cap P_2^{-1}(L(G_2))$,
- $L_m(G_1 \| G_2) = P_1^{-1}(L_m(G_1)) \cap P_2^{-1}(L_m(G_2))$,
- $G_1 \| G_2 = G_2 \| G_1$,
- $G_1 \| (G_2 \| G_3) = (G_1 \| G_2) \| G_3$.

위 성질들로부터 병렬합성 연산 $\|$를 언어에 대해 확대 정의 할 수 있음을 짐작하게 된다. 즉, $L_i \subseteq \Sigma_i^*$에 대해 $L_1 \| L_2$는 $L_1 \| L_2 := P_1^{-1}(L_1) \cap P_2^{-1}(L_2)$로 정의가능하다. 또한 병렬합성에 있어서 $\Sigma_1=\Sigma_2$이면 $G_1 \| G_2 = G_1 \times G_2$이고, $\Sigma_1 \cap \Sigma_2 = \varnothing$이면 동기화된 사건에 의한 상태천이가 전혀 없는 경우이며 이를 특히 뒤섞임(shuffle)이라고 부르기도 한다.

2-6. 페트리네트(Petri net)

이 절에서는 지금까지 다룬 오토마타와 FSM과 별도로 또 다른 논리적 이산사건모델로서 널리 쓰이고 있는 페트리네트(Petri net)를 소개한다. 페트리네트는 1962년 독일의 Carl A. Petri 가 그의 박사학위 논문에서 처음 소개한 모델링 형식(formalism)으로, 주로 모델링 및 시뮬레이션 등을 위해 유럽에서 특히 활발한 연구가 이루어져 왔다[17]-[21].

페트리네트 구조(또는 그래프) C 는 가중치를 가지는 이중엽 그래프(weighted bipartite graph)이다. 즉,

$$C := (P, T, A, W) \text{ 이고,}$$

여기서 P는 자리(place)들의 유한집합을, T는 천이(transition)들의 유한집합을, $A \subseteq (P \times T) \cup (T \times P)$는 그래프 상에 자리로부터 천이로 또는 천이로부터 자리로의 원호(arcs)들의 집합을, $W : A \mapsto \{1, 2, \cdots\}$는 각 원호들에 주어진 가중치 함수를 각각 나타낸다. 그러면, 페트리네트 N은 다음과 같이 5개의 원소로 구성된 순서쌍으로 정의된다.

$$N := \{C, \Sigma, l, x_0, X_m\}.$$

이 때 C는 페트리네트 구조이고, Σ는 천이에 따른 사건집합이며, $l : T \mapsto \Sigma$는 천이명명함수(transition labeling function)이다(일반적으로 $l(\cdot)$은 단사함수(injective function)가 아니다). 또한, $x_0 \in N^{|P|}$는 초기상태 또는 네트표기(marking of net)라 불리며 각 자리의 토큰(token) 초기값을 나타내고, $X_m \in N^{|P|}$는 표기상태들의 집합 또는 네트의 최종표기(final marking)를 나타낸다. 네트의 상태천이함수는 $\delta : N^{|P|} \times T \mapsto N^{|P|}$로 정의되며, 네트의 허용 가능한(admissible) 점화(firing)를 의미한다(점화란 사건의 실행을 나타내는 용어이다). 그리고 $X \subseteq N^{|P|}$는 이러한 x_0와 X_m을 포함하는 전체상태 집합이다. $\delta(\cdot, \cdot)$를 보다 자세히 기술하기 위해 다음의 일련의 함수들을 먼저 정의한다.

1) $In_t(t) := \{p \mid (p, t) \in A\}$,
2) $Out_t(t) := \{p \mid (t, p) \in A\}$,
3) $In_p(p) := \{t \mid (t, p) \in A\}$,
4) $Out_p(p) := \{t \mid (p, t) \in A\}$.

그러면, 네트의 동적특성은 다음과 같다. 모든 $p \in$

$In_t(t)$에 대해 $x(p) \geq W(p, t)$이면 $\delta(x, t)$가 정의되며 이 경우 $\delta(x, t) = x'$이 된다. 여기서 $x'(p)$는 $p \notin In_t(t)$이고 $p \notin Out_t(t)$이면 $x(p)$이고, $p \in In_t(t)$이나 $p \notin Out_t(t)$이면 $x(p) - W(p, t)$이고, $p \in In_t(t)$이면서 $p \in Out_t(t)$이면 $x(p) - W(p, t) + W(t, p)$이고, $p \notin In_t(t)$이나 $p \in Out_t(t)$이면 $x(p) + W(p, t)$이다. 그리고, 이와 같이 $\delta(x, t)$가 정의될 때, 상태 x에서 t가 가능(enable)하다고 한다. 이러한 상태천이함수는 또한 다음과 같이 확대 정의될 수 있다.

$$\delta : N^{|P|} \times T^* \mapsto N^{|P|}.$$

그러면 초기상태로부터 위에서 정의된 상태천이함수에 의해 상태천이가 이루어지며 이에 따라 시스템의 동적특성이 기술된다. 네트의 초기상태로부터 도달 가능한 상태집합 (set of reachable states) R(N)은 $R(N) := \{ x \in N^{|P|} \mid (\exists s \in T^*)$ 다음과 같은 $\delta(x_0, s) = x\}$로 정의된다.

[예] 다음 그림 8의 페트리네트를 생각해 보자. 자리 P_i 내의 검은 점은 토큰을 의미한다. 이 경우 앞서 정의된 상태천이함수에 따른 상태변화는 다음과 같다. 초기상태 $x_0 = [2, 0, 0, 1]^T$로부터 t_1에 의해 $x_1 = [1, 1, 1, 1]^T$로 천이되며, 다음 t_2에 의해 $[1, 1, 0, 2]^T$로 천이되거나 t_3에 의해 $[0, 1, 0, 0]^T$으로 천이된다. 그리고 이와 같은 식으로 계속 천이가 이루어진다.

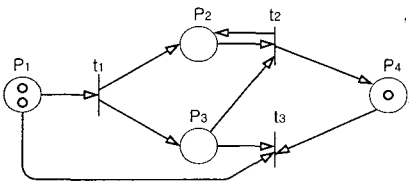

그림 8. 페트리네트의 예.

다음은 앞서 FSM이 언어의 인식기 또는 생성기로 쓰였던 것처럼 페트리네트에서도 각 천이에 사건명(label)을 인가하여 언어를 표현해보자. 먼저 앞서 정의한 천이명명함수 l을 다음과 같이 확대 정의한다.

$$l : T^* \mapsto \Sigma^*.$$

그러면 $L(N) := \{l(s) \in \Sigma^* \mid s \in T^*$ 그리고 $\delta(x_0, s)!\}$, $L_m(N) := \{l(s) \in L(N) \mid \delta(x_0, s) \in X_m\}$으로 각각 나타낼 수 있으며, 페트리네트에 의해 표현될 수 있는 언어의 클래스(class) PNL은 $PNL := \{ K \in \Sigma^* \mid (\exists N) K = L_m(N)\}$이 된다.

이러한 페트리네트는 주로 이산사건시스템의 충돌성(conflictness), 동시동작성(concurrency), 혼돈성(confusion), 한계성(boundedness)등의 정성적 동적특성을 시뮬레이션 등을 통해 분석하기 위한 모델링 도구로 연구 개발되어져 왔으며 그 사용목적에 따라 제한적(restricted) 페트리네트, FSM 페트리네트, 표기(marked) 페트리네트, 칼라(colored) 페트리네트 등의 다양한 형태로 변형되어 쓰여지기도 한다. 페트리네트는 대상 시스템의 상태를 내재적으로(implicitly) 표현하고 있어 오토마타나 FSM보다 모델링 측면에서는 더 우수한 면이 있으나 동적특성을 분석하기 위해서는 네트로부터 또다시 상태수목도(state tree)를 구성해야 하므로 해석적인 측면에서는 오히려 떨어진다고 볼 수 있다.

3. 관리제어이론

제어이론의 관점에서 보았을 때 주어진 시스템(또는 플랜트)에 대한 동적특성의 모델링과 이에 따른 해석은 궁극적으로 모델링을 토대로 대상 시스템의 출력(또는 동적특성의 변화)을 원하는 값이나 궤적으로 제어해 나가고자하는 능동적 취지에 있다. 본 논문에서 다루고 있는 이산사건시스템에 대해서도 이러한 관점에서 보았을 때 앞장에서 소개한 이산사건모델링을 기초로 시스템의 동적 변화를 원하는 방향으로 바꿀 수 있는 제어메카니즘이 필요하게 된다. 이 절에서는 바로 이런 목적을 위한 관리제어이론(supervisory control theory)[1], [4], [6], [22]의 소개와 관리제어기의 존재조건에 따른 성질 및 이의 구현 등을 다룬다.

3-1. 관리제어의 개념

2-2절에서 소개한 것과 같이 이산사건시스템의 언어모델은 오토마타 또는 FSM $G=(Q, \Sigma, \delta, q_o, Q_m)$로 표현할 수 있다(앞으로 다루고자하는 언어는 모두 정규언어로 가정한다. 이는 단순히 추후에 전개될 이론의 완결성을 위한 것이며, 일반적인 언어로도 확장 가능하므로 제한적 의미를 가지는 것은 아니다). 이렇게 모델링된 시스템은 제어이론의 관점에서 플랜트(plant)라 불린다. 이때 사건집합 Σ를 제어가능(controllable) 사건집합 Σ_c과 제어불가능(uncontrollable) 사건집합 Σ_{uc}의 합(disjoint union)으로 가정하자. 즉, $\Sigma=\Sigma_c \cup \Sigma_{uc}$이다. 제어불가능 사건은 물리적으로 시스템의 고장, 하드웨어

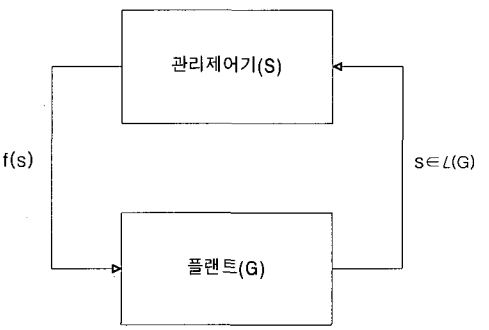

그림 9. 관리제어의 개념적 구조.

적 한계, 타이머의 클럭 등을 의미한다. 그러면, 관리제

어기(supervisory controller 또는 supervisor) S는 플랜트가 생성하는 사건열을 관측해가며 다음 생성될 사건들 가운데 제어 가능한 사건들을 선택적으로 억제(disable)함으로써 플랜트의 동적인 변화를 원하는 방향으로 바꿔나가는 외부 대리자(external agent)의 역할을 수행하게 된다. 이러한 개념적 정의가 그림 9에 도식화되어 있다.

주어진 플랜트 G에 대해 이와 같이 제어규칙에 기반한 관리제어기 S는 플랜트가 생성가능한 모든 언어로부터 제어가능사건 부분집합으로의 사상(map) $f : L(G) \mapsto 2^{\Sigma_c}$로 정의될 수 있다. 각각의 사건열 $s \in L(G)$에 대해 $f(s) \subseteq \Sigma_c \cap \Sigma_G(\delta(q_0, s))$는 플랜트가 s를 수행(execution)한 다음 관리제어기가 억제시키는 사건집합을 나타낸다. 관리제어기를 포함한 폐루프시스템, 관리제어시스템(supervised control system)의 언어 $L(S/G)$와 표기언어 $L_m(S/G)$는 각각 다음과 같다:

- $L(S/G)$: $\varepsilon \in L(S/G)$, $\forall s \in \Sigma^*$, $\sigma \in \Sigma$: $s \in L(S/G)$, $s\sigma \in L(G)$, $\sigma \notin f(s) \Rightarrow s\sigma \in L(S/G)$.
- $L_m(S/G) := L(S/G) \cap L_m(G)$.

즉, 어떤 사건열 s가 관리제어시스템에 의해 생성되고($s \in L(S/G)$), 사건 σ가 플랜트에 있어서 사건열 s다음에 발생하며($s\sigma \in L(G)$), 또한 사건 σ가 사건열 s뒤에 관리제어기에 위해 억제되지 않으면($\sigma \notin f(s)$), $s\sigma$는 관리제어시스템에서 생성된 언어(generated language)에 포함되어 있음[$s\sigma \in L(S/G)$]을 의미한다. 문헌에 따라[1] 이를 관리제어기의 완결성(completeness)이라는 성질로 규명하고 있다. 관리제어시스템의 표기언어는 플랜트의 표기언어들 가운데 관리제어 후에도 남아 있는 것들의 집합[$L_m(S/G) = L(S/G) \cap L_m(G)$]으로 간주될 수 있다. 한편, 이러한 언어들 상호간에는 $\phi \subseteq L_m(S/G) \subseteq \overline{L_m}(S/G) \subseteq L(S/G) \subseteq L(G)$의 관계식이 성립하며 $L_m(S/G) \subseteq L(S/G) = \overline{L}(S/G) \neq \phi$이므로 $\overline{L_m}(S/G) \subseteq L(S/G)$임을 알 수 있다. 그러나 그 역은 항상 성립되지는 않는다. 그러므로 관리 제어시스템이 생성하는 언어들 가운데는 경우에 따라 표기언어로 확장시킬 수 없을 수 있다. 이 경우 관리제어시스템은 막힘(blocking) 성질을 가진다고 한다. 그리고 이러한 경우를 항상 배제시킬 수 있는 관리제어기를 비막힘성(nonblocking)관리제어기라고 부르며 $\overline{L_m}(S/G) = L(S/G)$일 때 가능하다. 막힘성질은 물리적으로 교착상태(deadlock)와 지속잠금(livelock) 등의 현상을 나타낸다.

앞서 언급한 내용의 관리제어 개념을 플랜트와 관리제어기 FSM들의 동기합성(synchronous composition)이란 측면에서 생각해 볼 수 있다. $S := (Y, \Sigma, \beta, y_0, Y_m)$를 관리제어기의 FSM이라 하고, 동기합성에 따른 관리제어시스템의 FSM을 $G \parallel S := (Z, \Sigma, \gamma, z_0, Z_m)$이라 하자. 그러면 $L(G \parallel S) = L(G) \cap L(S)$이고 $L_m(G \parallel S) = L_m(G) \cap L_m(S)$이므로 동기합성에 의해 플랜트의 동적특성이 제한됨을 알 수 있다. 이와 같은 동기 합성에 기반한 관리제어기에 따른 관리제어시스템이 앞서 소개한 제어규칙기반 관리제어 시스템과 동일한 제어 결과를 낳기 위해선 관리제어기의 완결성과 $L_m(G \parallel S) = L(G \parallel S) \cap L_m(G)$의 두 가지 조건을 만족시켜야 한다. 먼저 첫 번째 조건은 $f : L(G) \rightarrow 2^{\Sigma_c}$ 형태의 제어규칙기반 관리제어에서처럼 제어불가능한 사건은 억제시킬 수 없음을 의미하며 두 번째 조건은 관리제어시스템의 표기언어가 플랜트의 표기언어들 가운데 관리제어 후에도 유지되는 언어들의 집합과 같음을 의미한다. 첫번째 조건은 동기합성기반 관리제어기의 경우, $\forall s \in \Sigma^*$, $\sigma \in \Sigma_u$: $s \in L(G \parallel S)$, $s\sigma \in L(G) \Rightarrow s\sigma \in L(G \parallel S)$의 조건식이 만족되면 성립된다. 즉, 관리제어시스템이 사건열 s를 수행한 뒤($s \in L(G \parallel S)$) 제어불가능한 사건 σ가 사건열 s에 후속해서 발생가능한 경우($s\sigma \in L(G)$), 관리제어기는 이를 허용(enable)해야함($s\sigma \in L(G \parallel S)$)을 의미한다. 이는 또한 제어불가능한 사건의 발생을 관리제어기가 유발시키는 논리적 모순을 배제하기 위한 것이다[7]. 그리고, 두번째 조건의 경우는 $L(G \parallel S) \cap L_m(G) = [L(G) \cap L(S)] \cap L_m(G) = L_m(G) \cap L(S)$이므로, $L_m(G) \cap L(S) = L_m(G \parallel S)$이면 만족됨을 알 수 있다. 한편, 이러한 동기합성기반 관리제어시스템에 있어서의 비막힘성 관리제어기의 조건은 $\overline{L_m}(G \parallel S) = L(G \parallel S)$로 주어짐을 같은 맥락에서 확대해석할 수 있다.

지금까지는 플랜트의 사건집합이 제어가능 사건들과 제어불가능 사건들로만 구성되어 있으며 모두 관측가능하다고 가정하였다. 그러나 실제로 플랜트에서 발생하는 사건들 가운데는 관리제어기가 감지할 수 없는 내부에 국한된 사건들도 있으며 이를 관측불가능(unobservable)사건으로 분류한다. 그러면 전체 사건집합은 관측가능(observable) 사건집합 Σ_o과 관측불가능 사건집합Σ_{uo}의 합(disjoint union)으로 나타낼 수 있다. 즉, $\Sigma = \Sigma_o \cup \Sigma_{uo}$이다. 이 때, 발생 사건열로부터 관측가능 사건열로의 투영사상(projection map)을 $P : \Sigma^* \mapsto \Sigma_o^*$라고 하면 $P(s_1) = P(s_2)$인 $s_1, s_2 \in L(G)$에 대해 관리제어기 S_p는 동일한 제어명령을 내리게 된다. 앞서와 같이 관리제어기 S_p의 제어기능을 $f_p : P(L(G)) \mapsto 2^{\Sigma_c}$의 사상으로 정의하면 관리제어시스템의 언어와 표기언어는 각각 다음과 같다.

- $L(S_p/G)$: $\varepsilon \in L(S_p/G)$, $\forall s \in \Sigma^*$, $\sigma \in \Sigma$: $s \in L(S_p/G)$, $s\sigma \in L(G)$, $\sigma \notin f_p(P(s)) \Rightarrow s\sigma \in L(S_p/G)$.
- $L_m(S_p/G) := L(S_p/G) \cap L_m(G)$.

이와 같은 부분관측하에서의 관리제어개념이 그림 10에 도식화되어 있다.

지금까지 소개한 관리제어의 궁극적인 목적은 제어대상 이산사건시스템의 여러 가능한 동작형태들로부터 원하는 동작만을 허용하고자 하는데 있다. 그러기 위해선 먼저 원하는 동작형태, 즉 정상동작(legal behavior)을

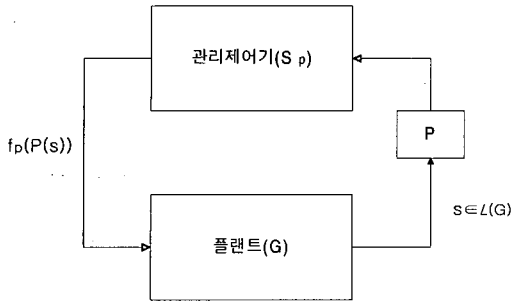

그림 10. 부분관측하에서 관리제어의 개념적 구조.

언어로 표현해야 한다. 또는 반대로 비정상동작(illegal behavior)을 배제시키고자 하는 언어로 표현해야 한다. 정상동작만을 포함한 언어를 허용가능(admissible) 언어 $L_a(G)$로 나타내고 이 가운데 표기상태를 포함한 언어를 $L_{am}(G)$로 나타낸다. 주어진 각각의 사양(specification)을 Ks_i, i∈[1,m]라고 하면

$$L_a(G) = L(G) \cap \left(\bigcap_{i=1}^{m} Ks_i \right) \text{로,}$$

$$L_{am}(G) = L_m(G) \cap \left(\bigcap_{i=1}^{m} Ks_i \right) \text{로 각각 주어진다.}$$

[예] 앞의 그림 4의 간단한 생산시스템의 예를 생각해 보자. 이 경우 버퍼는 1개의 슬롯(slot)만을 가진다고 가정한다. 그리고 이 시스템의 정상동작에 대한 사양이 다음과 같이 주어졌다고 가정하자.

- 사양 1 : 버퍼(B)의 과잉(overflow) 또는 결핍(underflow)이 없어야 한다.
- 사양 2 : 만일 2대의 기계가 모두 고장나면, 기계 2(M_2)가 먼저 수리되어야 한다.

각 기계(M_i, i=1, 2)와 버퍼의 부분이산사건모델은 그림 11과 같다.

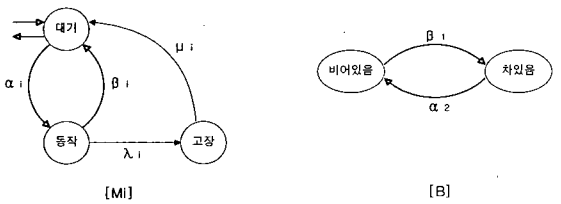

그림 11. 기계와 버퍼의 부분이산사건모델.

전체 시스템의 이산사건모델 G는 병렬합성을 통해 $G=M_1 \parallel M_2$로 구할 수 있다. 이 경우 사양 1에 대한 언어를 Ks_1, 사양2에 대한 언어를 Ks_2라고 하면 $Ks_1=L(H_1)$, $Ks_2=L(H_2)$로 각각 주어지며, H_1, H_2는 그림 12와 같다.

3-2 관리제어기의 존재조건 I : 제어가능성

지금까지 관리제어의 개념과 관리제어시스템에 관해 언급하였다. 이 절에서는 주어진 플랜트와 관리제어스

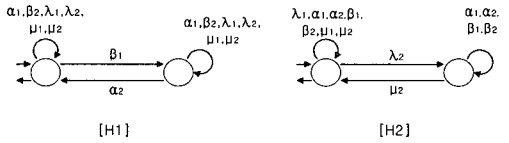

그림 12. 사양1, 2의 인식기.

시템의 사양에 대한 언어가 주어졌을 때 이를 만족시키는 관리제어기가 존재할 존재조건에 대해 다룬다. 먼저 대상 이산시스템에서 발생하는 모든 사건들이 관측가능하다고 가정하자. 일반적으로 어떤 언어 K의 제어불가능 사건집합 Σ_{uc}과 참조언어(reference language) M에 대한 제어가능성(controllability)은 다음과 같이 정의된다.

정의 3.1(제어가능성) : K와 $M(=\overline{M})$을 사건집합 Σ로부터 정의된 언어라 하고 Σ_{uc}를 Σ의 제어불가능 사건 부분집합이라 할 때 $\overline{K}\Sigma_{uc} \cap M \subseteq \overline{K}$이면 K는 M과 Σ_{uc}에 대해 제어가능이라고 한다. 앞의 정리 3.1로부터 K가 제어가능일 경우 $L_m(S/G)=K$가 아닌 $L(S/G)=K$ 또는 $L(S/G)=\overline{K}$를 만족시키는 관리제어기 S는 보다 간략화된 조건만을 만족시키면 존재함을 알 수 있다.

다음 정리 3.1은 관리제어시스템의 사양에 대한 언어 K가 제어가능(controllable)인 경우 이를 만족시키는 관리제어기 S가 존재함을 보여준다.

정리 3.1(제어가능성 정리)[1], [22] : 제어불가능 사건집합 $\Sigma_{uc} \subseteq \Sigma$를 포함한 이산사건시스템 G에 대해 사양 $K \subseteq L_m(G)$가 주어졌을 때 K가 다음 2가지 조건을 만족시키면 $L_m(S/G)=K$인 비막힘성 관리제어기 S가 존재하며 그 역도 성립된다:

(1) $\overline{K}\Sigma_{uc} \cap L(G) \subseteq \overline{K}$,

(2) K는 $L_m(G)$-닫힘성을 지닌다(즉, $K=\overline{K} \cap L_m(G)$).

위 정리 3.1은 주어진 이산사건시스템 G와 관리제어시스템의 사양 K에 대해 K가 제어가능일 경우 관리제어기 S의 존재여부에 대한 정보를 제공한다. 즉, 사양 K가 초기 상태로부터 발생가능한 모든 제어불가능 사건들에 의한 후속 사건열들을 포함하고 있어야만 관리제어기를 통해 이를 만족시킬 수 있음을 의미한다.

보조정리 3.1[1] : 제어불가능 사건집합 $\Sigma_{uc} \subseteq \Sigma$를 포함한 이산사건시스템 G에 대해

(1) $K \subseteq L(G)$로 주어진 경우 K가 접두닫힌 언어이고 $L(G)$와 Σ_{uc}에 대해 제어가능이면 $L(S/G)=K$를 만족시키는 S가 존재하며,

(2) $K \subseteq L_m(G)$로 주어진 경우 K가 $L(G)$와 Σ_{uc}에 대해 제어가능이면 $L(S/G)=\overline{K}$를 만족시키는 S가 존재하고,

각각 그 역도 성립된다.

3-3 관리제어기의 존재조건 II : 관측가능성

앞 절에서는 플랜트에서 발생하는 모든 사건들이 관

측가능한(observable) 경우 관리제어시스템의 사양에 대한 언어 K가 제어가능이면 이를 만족시키는 관리제어기 S가 존재함을 보였다. 그러나 많은 경우 플랜트 내부에서 발생되는 사건들 가운데는 외부, 즉 관리제어기에서 감지할 수 없는 관측불가능(unobservable)사건들이 존재하므로, 보다 현실적 상황하에서 관리제어기의 존재조건을 유도하기 위해 K의 관측가능성(observability)[23], [24]을 고려할 필요가 있다. 일반적인 관측가능성에 대한 정의를 기술하기 이전에 다음의 NEXTACT라는 삼원관계(ternary relation)를 먼저 생각해 보자. Σ^*에 대해 $M=\overline{M}$, $K\subseteq M$인 언어 M과 K가 주어졌을 때 $\Sigma^*\times\Sigma\times\Sigma^*$상에서의 삼원관계 $NEXTACT_{K,M}$은 $(s,\sigma,s')\in\Sigma^*\times\Sigma\times\Sigma^*$이 $[s\,\sigma\in\overline{K}]\wedge[s'\in\overline{K}]\wedge[s'\,\sigma\in M]\Rightarrow s'\,\sigma\in\overline{K}$을 만족시키면 성립되며 $(s,\sigma,s')\in NEXTACT_{K,M}$으로 표기한다. 즉, 삼원관계 $NEXTACT_{K,M}$은 관리제어기가 사건열 s, s'을 각각 감지하였을 때 모두 동일하게 후속사건 σ를 허용하는 제어기능(control action)을 수행함을 의미한다. 그러면 관리제어시스템의 사양으로 주어지는 언어 K의 관측가능성에 대한 정의는 다음과 같다.

정의 3.2(관측가능성) : K와 $M(=\overline{M})$을 사건집합 Σ로부터 정의된 언어라 하고, Σ_c를 Σ의 제어가능사건 부분집합, 그리고 Σ_o를 Σ의 관측가능사건 부분집합이라 하며 P를 Σ^*로부터 Σ_o^*로의 투영사상이라 할 때 모든 $s,s'\in\Sigma^*$에 대해 $P(s)=P(s')\Rightarrow{}^{\forall}\sigma\in\Sigma_c$, $(s,\sigma,s')\in NEXTACT_{K,M}$ 이면 K는 M, P 그리고 Σ_c에 대해 관측가능이라고 한다.

[예] $\Sigma=\{\alpha,\beta,\gamma,\mu_{01},\mu_{02}\}$, $\Sigma_c=\Sigma$, $\Sigma_o=\{\alpha,\beta,\gamma\}$, $M=\overline{\mu_{01}\alpha(\beta+\gamma)+\mu_{02}\alpha\gamma}$, $K=\mu_{01}\alpha(\beta+\gamma)+\mu_{02}\alpha$로 각각 주어졌을 때 K가 관측가능인지 살펴보자. 이 경우 예를 들어 $s=\mu_{01}\alpha$, $s'=\mu_{02}\alpha$에 대해 $P(s)=P(s')=\alpha$이지만 $\sigma=\gamma$일 때 $[\mu_{01}\alpha\gamma\in\overline{K}]\wedge[\mu_{02}\alpha\gamma\in M]$이나 $\mu_{02}\alpha\gamma\notin\overline{K}$ 이므로 $(s,\sigma,s')\notin NEXTACT_{K,M}$이 되어 K는 관측불가능이다. 한편 $K'=\mu_{01}\alpha(\beta+\gamma)$의 경우는 모든 $s,s'\in\Sigma^*$에 대해 $P(s)=P(s')\Rightarrow\forall\sigma\in\Sigma_c$, $(s,\sigma,s')\in NEXTACT_{K,M}$ 이므로 관측가능임을 알 수 있다.

지금까지 다룬 $NEXTACT_{K,L(G)}$ 관계와 관측가능성의 정의를 관리제어기의 역할로부터 재조명해보자. 부분관측(partial observation)하에서의 관리제어기를 P-관리제어기(partial supervisor) S_p라하고, $s\in\Sigma^*$관측뒤의 관리제어기능을 $S_P(P(s)) := \Sigma_G(\delta(q_0,s))\text{-}f_p(P(s))$로 정의하면, $NEXTACT_{K,L(G)}$의 조건식$[s\,\sigma\in\overline{K}]\wedge[s'\in\overline{K}]\wedge[s'\,\sigma\in L(G)]\Rightarrow s'\,\sigma\in\overline{K}$는 $[\sigma\in S_P(P(s))]\wedge[s'\in L(S_p/G)]\wedge[s'\,\sigma\in L(G)]\Rightarrow s'\,\sigma\in L(S_p/G)$ 로 나타내어질 수 있다. 앞의 예에서 보면 K'의 경우 $\mu_{02}\in\Sigma_{uc}\cap\Sigma_c$ 를 관리제어기가 초기상태에서 억제(disable)함으로써, 즉 $f_p(P(\varepsilon))=\{\mu_{02}\}$, 관리제어시스템이 K'를 만족시키게되는 것이다. 다음은 지금까지 소개한 K의 제어가능성과 관측가능성의 정의로부터 일반적인 상황하에서 관리제어시스템이 K를 만족시키도록 하는 관리제어기가 존재할 필요충분조건을 살펴보자.

정리 3.2(제어가능성 및 관측가능성 정리)[23], [24] : 제어불가능 사건집합 $\Sigma_{uc}\subseteq\Sigma$와 관측가능 사건집합$\Sigma_o\subseteq\Sigma$를 포함한 이산사건시스템 G에 대해 사양 $K\subseteq L_m(G)$, $K\neq\phi$와 Σ^*로부터 Σ_o^*로의 투영사상 P가 주어졌을 때, K가 다음 3가지 조건을 만족시키면 $L_m(S_p/G)=K$인 비막힘성 P-관리제어기 S_p가 존재하며 그 역도 성립한다 :

(1) K는 $L(G)$와 Σ_{uc}에 대해 제어가능이고,
(2) K는 $L(G)$, P 그리고 Σ_c에 대해 관측가능이며,
(3) K는 $L_m(G)$-닫힘성을 지닌다.

보조정리 3.2[24] : 제어불가능 사건집합 $\Sigma_{uc}\subseteq\Sigma$와 관측가능 사건집합 $\Sigma_o\subseteq\Sigma$를 포함한 이산사건시스템 G에 대해 Σ^*로부터 Σ_o^*로의 투영사상을 P라고 정의하면

(1) 사양 $K\subseteq L(G)$, $K\neq\phi$가 주어졌을 때 K가 접두닫힌 언어이고 $L(G)$와 Σ_{uc}에 대해 제어가능이며, $L(G)$, P 그리고 Σ_c에 대해 관측가능이면 $L(S_p/G)=K$인 P-관리제어기 S_p가 존재하고,
(2) 사양 $K\subseteq L_m(G)$, $K\neq\phi$가 주어진 경우 K가 $L(G)$와 Σ_{uc}에 대해 제어가능이며, $L(G)$, P, 그리고 Σ_c에 대해 관측가능이면 $L(S_p/G)=\overline{K}$인 P-관리제어기 Sp가 존재하며,

각각 그 역도 성립된다.

3-4. 관리제어기의 구현

앞 절에서는 관리제어시스템의 사양 K가 주어졌을 때 이를 만족시키는 관리제어기의 존재조건에 관해 다루었다. 이 절에서는 그러한 존재조건이 성립될 때 관리제어기의 구현(realization)방식을 소개한다.

관리제어기의 구현에 있어서 가장 기본적이고 직관적인 방법은 K의 인식기 FSM을 이용한 표준구현(standard realization)방식이다. 먼저 모든 발생하는 사건들은 관측가능으로 가정하고 $L(S/G)=\overline{K}$, $L_m(S/G)=K$을 만족시키는 관리제어기 S를 생각해보자(이 때 $K=\phi$ 또는 $K=L(G)$의 사소한 경우는 배제시키기로 한다). 이러한 K에 대해 $L_m(R)=L(R)=\overline{K}$ 인 트림 인식기 $R=(X,\Sigma,\xi,x_0,X)$을 생각할 수 있다. 그러면 플랜트 G와 R의 동기합성 $R\times G$는 우리가 원하는 폐루프(closed-loop) 관리제어시스템 S/G의 동적특성을 보이게 된다. 즉,

$$\begin{aligned}L(R\times G) &= L(R)\cap L(G)\\ &= \overline{K}\cap L(G)\\ &= \overline{K}\\ &= L(S/G)\text{이고,}\\ L_m(R\times G) &= L_m(R)\cap L_m(G)\end{aligned}$$

$= \overline{K} \cap L_m(G)$
$= L(S/G) \cap L_m(G)$
$= L_m(S/G)$이 된다.

이때 관리제어기능(supervisory control action)은 $S(s) \cap \sum_G(\delta(q_0, s)) = \sum_{R\times G}(\xi \times \delta(x_0, q_0), s) = \sum_R(\xi(x_0, s))$이 되어 $S(\cdot)$의 역할이 인식기 R의 활성(active) 사건집합으로 표현되며 그 결과 S/G의 동적특성이 동기합성된 FSM R×G의 동적특성과 같게 되는 것이다. 이러한 인식기 R에 의한 관리제어기 구현방식을 S의 표준구현이라고 한다.

[예] 그림 13의 플랜트 G의 인식기와 사건집합 $\sum$가 $\sum = \sum_c \cup \sum_{uc} = \{\beta\} \cup \{\alpha, \gamma\}$와 같이 주어졌을 때, $K = \overline{(\alpha\beta)^* + (\alpha\beta)^*\alpha\gamma^*}$에 대해 $L(S/G)=K$를 만족시키는 관리제어기 S의 구현을 생각해보자.

이 경우 $L(G) = \overline{(\alpha\gamma^*\beta)^*} = \overline{L_m}(G)$이고, $K = \overline{K} \subseteq L(G)$이며 K가 $L(G)$와 $\sum_{uc}$에 대해 제어가능이므로 보조정리 3.1로부터 $L(S/G)=K$를 만족시키는 S가 존재함을

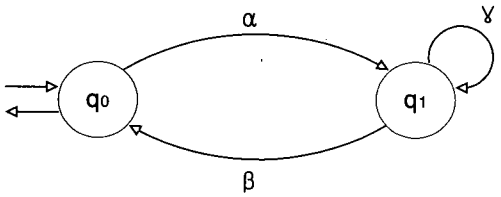

그림 13. 플랜트 G의 인식기.

알 수 있다. 그러한 관리제어기 S의 구현을 위해 $L_m(R)=L(R)=\overline{K}$인 그림 14의 인식기 R과 표 2에 나타나 있는 R의 각 상태에서 활성 사건집합내의 사건들에 대한 관리제어기능을 생각해 보자.

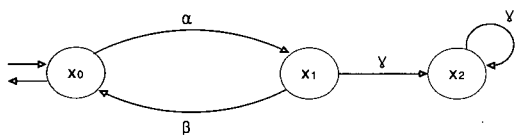

그림 14. 인식기 R.

표 2. 관리제어기능.

상 태	허용사건	억제사건
x_0	α	ϕ
x_1	$\beta\gamma$	ϕ
x_2	γ	β

그러면 R×G에 의해 $L(S/G)=K$임을 알 수 있다.

만일 $K' = (\alpha\beta)^*$가 주어진 경우 $L(S/G)=\overline{K'}$를 만족시키는 관리제어기 S의 구현을 생각해 보자. 이 때 $K' \subseteq L_m(G)$이고, $\alpha \in \overline{K'}$, $\gamma \in \sum_{uc}$에 대해 $\alpha\gamma \in L(G)$ 이지만 $\alpha\gamma \notin \overline{K}$이므로 K'는 제어불가능이어서 보조정리 3.1로부터 $L(S/G)=\overline{K'}$를 만족시키는 S는 존재하지 않음을 알 수 있다. 한편 $L(S/G)=\overline{K}$, $L_m(S/G)=K'$를 만족시키는 관리제어기 S는 존재하지만 막힘성을 지니게 된다. 그러나 만일 이 경우 플랜트 G가 그림 15와 같이 주어졌다면 관리제어기 S는 비막힘성이 된다.

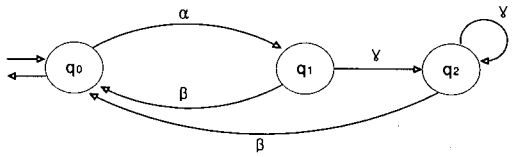

그림 15. 플랜트 G의 인식기.

앞서 관리제어기의 표준구현 방식에 있어서는 K의 인식기 R로부터 직접적인 관리제어기능을 도출해 내었다. 이러한 개념을 확장하여 K의 인식기가 아닌 일반적인 트림 FSM M을 도입해 M×G로부터 관리제어시스템 S/G의 동적특성을 보일 수 있는데 이 방식을 유도구현(induced realization)이라고 한다. $M := (Y, \sum, \xi, y_0, Y)$을 트림 FSM이라고 하자. 그러면 플랜트 G로부터 사건열 $s \in L(G)$가 관측되었을 때 M으로부터 유도되는 관리제어기능 $S_i^M(s)$는

1) 만일 $s \in L(G) \cap L(M)$이면 $\sum_{uc} \cup \{\sigma \in \sum_c \mid s\sigma \in L(M)\}$ 으로 정의되고,
2) 그 외의 경우는 $\sum_{uc}$로 정의된다.

따라서 $L(S_i^M/G) = L(M\times G) = L(M) \cap L(G)$이고, $L_m(S_i^M/G) = L_m(M\times G) = L_m(M) \cap L_m(G) = L(M) \cap L_m(G) = L(M) \cap L(G) \cap L_m(G) = L(S_i^M/G) \cap L_m(G)$이므로, $L(M)$이 $L(G)$와 $\sum_{uc}$에 대해 제어가능이면 $L(S_i^M/G) = L(M\times G)$이 되고 그 역도 성립됨을 알 수 있다.

문헌에 따라서는[1], [22], 앞서 소개한 동기합성에 의한 관리제어기의 구현방식을 비슷한 개념이지만 다른 방법으로 기술하기도 한다. 즉, 관리제어기 S는 플랜트가 생성하는 언어를 인식하는 인식기 S와 피드백사상(feedback map) ϕ의 쌍(pair) (S, ϕ)로 표현된다. 여기서 피드백사상 ϕ는 인식기 S의 사건과 상태들로부터 집합 {1(허용), 0(억제)}로의 사상을 의미한다. 즉, X를 S의 상태집합이라고 하면, $\phi : \sum \times X \mapsto \{1,0\}$는

1) 만일 $x \in X$에 대해 $\sigma \in \sum_{uc}$이면 $\phi(\sigma, x)=1$이고,
2) 만일 $x \in X$에 대해 $\sigma \in \sum_c$이면 $\phi(\sigma, x) \in \{1,0\}$으로 정의된다.

그러면 인식기 S는 G의 동적특성을 모사해가며 $\phi(\cdot, \cdot)$를 통해 관리제어해 나간다. 플랜트 G로부터 생성된 사건에 의해 인식기 S의 상태도 천이되며 또한 S의 각 상태에서 정의되는 $\phi(\sigma, x)$에 의해 플랜트 G의 다음 발생가능 사건 σ를 허용 또는 억제함으로써 관리제어 시스템의 동적특성이 이루어 진다.

부분관측하에서 P-관리제어기의 구현은 먼저 사양 K

에 대해 $\overline{K}$ 의 인식기를 구한 뒤 관측불가능 사건에 의한 천이로부터 형성되는 비결정적(nondeterministic) FSM을 동등한(equivalent) 결정적(deterministic) FSM으로 변환한 뒤 마찬가지로 앞서와 같은 방식을 통해 이루어진다[23], [24].

4. 결론

본 논문에서는 기존의 연속변수시스템과 대비해 이산사건시스템을 설명하고 아울러 논리적 이산사건모델링과 관리제어이론 등을 소개하였다. 이러한 이산사건시스템의 관리제어 및 이의 활용에 대한 연구는 지금까지 다방면에 걸쳐 이루어져 왔으며 지금도 세계 도처에서 활발히 진행되고 있다. [25]-[29]에서는 이산사건모델링에 기반한 고장검출 및 진단기법, 그리고 관리제어이론을 이용한 내고장성 관리제어기법에 관한 연구가 이루어졌으며, [30]-[35]에서는 대규모(large scale)의 복잡한 시스템에 대한 관리제어기 설계시 계산상의 복잡도(computational complexity) 문제를 해결하고 부분모델의 변화에 유연성(flexibility)있게 대처하고자 하는 분산 관리제어(decentralized supervisory control)에 관한 연구가 다루어졌고, [36]-[41]에서는 관리제어시스템(supervisory control system)의 동적특성을 허용언어(admissible language) 범위 이내에서 최적화시키는 최적 관리제어(optimal supervisory control)에 관한 연구가 각각 이루어졌다. 또한 [42]-[44]에서는 대상 이산사건시스템의 전체 이산사건모델이 아닌 부분 모델을 이용해 온라인(on-line) 형태로 관리제어기능을 수행함으로써 실시간 제어(real-time control)에 응용하려는 연구가 이루어졌고, [45]-[47]에서는 사건 발생시 시간이라는 또 다른 제약조건(constraint)이 인가되었을 경우 시간 이산사건모델 상에서의 관리제어에 관한 연구가 다루어졌다. 그리고 [48]-[52]에서는 이산사건시스템의 안정도(stability) 해석 및 안정화(stabilization)를 위한 관리제어의 연구가 이루어졌다. 최근에는 추계적 이산사건모델[53], [54] 및 비결정적 이산사건모델[55], [56] 상에서의 관리제어에 관한 연구 등 보다 복잡한 실제 시스템의 다양한 동적특성을 고려하려는 시도가 이루어지고 있으며, 연속변수시스템과 이산사건시스템의 동적특성이 혼용된 하이브리드(hybrid) 시스템의 해석 및 제어에 관한 연구로 확장되고 있다[57]-[59]. 이러한 연구는 앞으로도 컴퓨터 시스템, 통신망, 생산시스템 등의 제어 및 자동화라는 실제적 문제의 해(solution)를 제공할 수 있는 이론적 · 실험적 토대를 마련하기 위한 방향으로 꾸준히 이어질 것으로 사료된다.

참고문헌

[1] P. J. Ramadge and W. M. Wonham, "Supervisory control of a class of discrete event processes", *SIAM J. of Control and Optimization*, vol. 25, pp. 206-230, 1987.

[2] W. M. Wonham and P. J. Ramadge, "On the supremal controllable sublanguage of a given language", *SIAM J. of Control and Optimization*, vol. 25. pp. 637-659, 1987.

[3] P. J. Ramadge and W. M. Wonham, "Modular feedback logic for discrete event systems", *SIAM J. of Control and Optimization*, vol. 25, pp. 1202-1218, 1987.

[4] R. Kumar and V. K. Garg, *Modeling and Control of Logical Discrete Event Systems*, MA : Kluwer Academic Publishers, 1995.

[5] A. Tornambe, *Discrete-Event System Theory*, NJ: World Scientific Publishers, 1995.

[6] C. G. Cassandras, *Discrete Event Systems: Modeling and Performance Analysis*, Aksen Associates, Inc., 1993.

[7] Y.-C. Ho, "Dynamics of discrete event systems", *Proc. of IEEE*, pp. 3-6, Jan, 1989.

[8] C. G. Cassandras and P. J. Ramadge, "Toward a control theory for discrete event systems", *IEEE Control Systems Magazine*, pp. 66-68, June 1990.

[9] X.-R. Cao and Y.-C. Ho, "Models of discrete event dynamic systems", *IEEE Control Systems Magazine*, pp. 69-76, June 1990.

[10] M. Sampath, R. Sengupta, S. Lafortune, K. Sinnamohideen, and D. Teneketzis, "Failure diagnoesis using discrete-event models", *IEEE Trans. Contr. Syst. Technol.*, vol. 4, pp. 105-124, 1996.

[11] K. Rudie and W. M. Wonham, "Protocol verification using discrete-event systems", *Proc. of IEEE Conf. on Decision and Control, Tucson, AZ, pp. 3770*-3777, 1992.

[12] G. Westerman, R. Kumar, C. stroud, and J. R. Heath, "Discrete event system approach for delay fault analysis in digital circuits", *Proc. of American Contr. Conf.*, Philadelphia, PE, pp. 239-243, 1998.

[13] J. Prosser, J. Selinsky, H. Kwatny, and M. Kam, "Supervisory control of electric power transmission networks", *IEEE Trans. Power Systems*, vol. 10, pp. 1104-1110, 1995.

[14] J. C. Martin, *Introduction to Languages and the Theory of Computation*, Singapore : McGraw-Hill, 1991.

[15] J. E. Hopcroft and J. D. Ullman, *Introduction to Automata Theory, Languages, and Computation*, MA : Addison-Wesley, 1979.

[16] M. Heymann, "Concurrency and discrete event control", *IEEE Control Systems*, pp. 103-112, June 1990.

[17] T. Murata, "Petri nets : properties, analysis and applications", *Proc. IEEE*, vol. 77, pp. 541-580, April 1989.

[18] J. L. Peterson, *Petri Net Theory and the Modeling of*

Systems, NJ : Prentice Hall, 1981.

[19] L. Holloway and B. H. Krogh, "Synthesis of feedback control logic for a class of controlled Petri nets", *IEEE Trans. Automat. Contr.*, vol. 35, pp. 514-523, 1990.

[20] C. Haoxun, "Net structure and control logic synthesis of controlled Petri nets", *IEEE Trans. Automat. Contr.*, vol. 43, pp. 1465-1468, 1998.

[21] R. S. Sreenivas, "On supervisory policies that enforce liveness in a class of completely controlled Petri nets obtained via refinement", *IEEE Trans. Automat. Contr.*, vol. 44, pp. 173-177, 1999.

[22] P. J. Ramadge and W. M. Wonhan, "The control of discrete event systems", *Proc. IEEE*, vol. 77, pp. 81-98, 1989.

[23] F. Lin and W. M. Wonhan, "On observability of discrete-event systems", *Information Sciences*, vol. 44, pp. 173-198, 1988.

[24] R. Cieslark, C. Desclaux, A. S. Fawaz, and P. Varaiya, "Supervisory control of discrete-event processes with partial observations", *IEEE Trans. Automat. Contr.*, vol. 33, pp. 249-260, 1988.

[25] F. Lin, "Diagnosability of discrete event systems and its applications", *Discrete Event Dynamic Systems*, vol. 4, pp. 197-212, 1994.

[26] M. Sampath, R. Sengupta, S. Lafortune, K. Sinnamohideen, and D. Teneketzis, "Diagnosability of discrete-event systems", *IEEE Trans. Automat. Contr.*, vol. 40, no. 9, pp. 1555-1575, 1995.

[27] M. Sampath, R. Sengupta, S. Lafortune, K. Sinnamohideen, and D. Teneketzis, "Failure diagnosis using discrete-event models", *IEEE Trans. Contr. Syst. Technol.*, vol. 4, no. 2, pp. 105-124, 1996.

[28] M. Sampath, S. Lafortune, and D. Teneketzis, "Active diagnosis of discrete-event systems", *IEEE Trans. Automat. Contr.*, vol. 43, no. 7, pp. 908-929, 1998.

[29] K.-H. Cho and J.-T. Lim, "Synthesis of fault-tolerant supervisor for automated manufacturing systems: A case study on photolithographic process", *IEEE Trans. Robot. Automat.*, vol. 14, no. 2, pp. 348-351, 1998.

[30] W. M. Wonham and P. J. Ramadge, "Modular supervisory control of discrete event systems", *Math. of Control, Signals, and Systems*, vol. 1, pp. 13-30, 1998.

[31] F. Lin and W. M. Wonham, "Decentralized control and coordination of discrete event systems with partial observation", *IEEE Trans. Automat. Contr.*, vol. 35, pp. 1330-1337, 1990.

[32] F. Lin, "Control of large scale discrete event systems: Task allocation and coordination", *Systems and Control Letters*, vol. 17, pp. 169-175, 1991.

[33] K. Rudie and W. M. Wonham, "Think globally, act locally: Decentralized supervisory control", *IEEE Trans. Automat. Contr.*, vol. 37, pp. 1692-1708, 1992.

[34] Y.-L. Chen and S. Lafortune, "Modular supervisory control with priorities for discrete event systems", *Proc. IEEE Conf. on Decision and Control*, New Orleans, LA, pp. 409-415, 1995.

[35] K.-H. Cho and J.-T. Lim, "Mixed centralized/decentralized supervisory control of discrete event dynamic systems", *Automatica*, vol. 35, pp. 121-128, 1999.

[36] K. M. Passino and P. J. Antsaklis, "On the optimal control of discrete event systems", *Proc. IEEE Conf. on Decision and Control*, Tampa, FL, pp. 2713-2718, 1989.

[37] A. M. J. Skulimowski, "Optimal control of asynchronous discrete event systems", *Proc. IFAC World Congress*, Tallinn, Estonia, USSR, pp. 243-249, 1990.

[38] F. Lin, "A note on optimal supervisory control", *Proc. IEEE Int. Symp. on Intelligent Control*, Arlington, Virginia, pp. 227-232, 1991.

[39] O. R. Boissel and J. C. Kantor, "Optimal feedback control design for discrete event systems using simulated annealing", *Computers and Chemical Engineering*, vol. 19, pp. 253-266, 1995.

[40] R. Kumar and V. K. Garg, "Optimal supervisory control of discrete event dynamical systems", *SIAM J. of Control and Optimization*, vol. 33, pp. 419-439, 1995.

[41] K.-H. Cho and J.-T. Lim, "Layered optimal supervisory control of discrete event dynamic systems", *Int. J. of Systems Science*, vol. 30, no. 4, pp. 395-405, 1999.

[42] S.-L. Chung, S. Lafortune, and F. Lin, "Limited lookahead policies in supervisory control of discrete event systems", *IEEE Trans. Automat. Contr.*, vol. 37, pp. 1921-1935, 1992.

[43] R. Kumar, H. K. Cheung, and S. I. Marcus, "Extension based limited lookahead control for discrete event systems", *Proc. IEEE Conf. on Decision and Control*, vol. 2, pp. 2225-2230, 1996.

[44] K.-H. Cho and J.-T. Lim, "On-line tracing supervisory control of discrete event dynamic systems based on outlooking", *Automatica*, vol. 35, no. 10, pp. 1725-1729, 1999.

[45] J. S. Ostroff and W. M. Wonham, "A framework for real-time discrete event control", *IEEE Trans. Automat. Contr.*, vol. 35, no. 4, pp. 386-397, 1990.

[46] B. A. Brandin and W. M.Wonham, "Supervisory control of timed discrete-event systems", *IEEE Trans. Automat. Contr.*, vol. 39, no. 2, pp. 329-342, 1994.

[47] T.-J. Ho, "Controller synthesis for some control problems in timed discrete-event systems", *Proc. IEEE Conf. on Decision and Control*, pp. 4613-4618, 1997.

[48] C. M. özveren, A. S. Willsky, and P. J. Antsaklis, "Stability and stabilizability of discrete event dynamic systems", *J. of the Association for Computing Machinery*, vol. 38, pp. 730-752, 1991.

[49] K. M. Passino, A. N. Michel, and P. J. Antsaklis, "Lyapunov stability of a class of discrete event systems", *IEEE Trans. Automat. Contr.*, vol. 39, pp. 269-279, 1994.

[50] S. Takai, T. Ushio, and S. Kodama, "Stabilization and blocking in state feedback control of discrete event systems", *Discrete Event Dynamic Systems: Theory and Application*, vol. 5, pp. 33-57, 1995.

[51] K.-H. Cho and J.-T. Lim, "A study on stability analysis of discrete event dynamic systems", *IEICE Trans. Information and Systems*, vol. E80-D, pp. 132-137, 1997.

[52] K.-H. Cho and J.-T. Lim, "Stability and robustness of discrete event dynamic systems", *Int. J. of Systems Science*, vol. 28, no. 7, pp. 691-703, 1997.

[53] R. Kumar and V. K. Garg, "Control of stochastic discrete event systems: Synthesis", *Proc. IEEE Conf. on Decision and Control*, pp. 3299-3304, 1998.

[54] V. K. Garg, R. Kumar, and S. I. Marcus, "A probabilistic language formalism for stochastic discrete-event systems", *IEEE Trans. Automat. Contr.*, vol. 44, no. 2, pp. 280-293, 1999.

[55] R. Kumar and M. A. Shayman, "Nonblocking supervisory control of nondeterministic systems via prioritized synchronization", *IEEE Trans. Automat. Contr.*, vol. 41, no. 8, pp. 1160-1175, 1996.

[56] M. Heyman and F. Lin, "Discrete-event control of nondeterministic systems", *IEEE Trans. Automat. Contr.*, vol. 43, no. 1, pp. 3-17, 1998.

[57] M. S. Branicky, "Studies in hybrid systems: Modeling, analysis, and control", *Ph.D. Dissertation*, MIT, 1995.

[58] H. Ye, A. N. Michel, and L. Hou, "Stability theory for hybrid dynamical systems", *IEEE Trans. Automat. Contr.*, vol. 43, no. 4, pp. 461-474, 1998.

[59] M. D. Lemmon, K. X. He, and I. Markovsky, "Supervisory hybrid systems", *IEEE Control Systems Magaine*, pp. 42-55, Aug. 1999.

조 광 현

1993년 한국과학기술원 전기및전자공학과 졸업. 동대학원 석사(1995), 동대학원 박사(1998). 1998.~1999. 동대학원 위촉연구원 및 연수연구원, 1999. 3.~현재 울산대학교, 전기전자및자동화공학부, 전임강사.

관심분야는 supervisory control of discrete event systems, nonlinear control, analysis and synthesis of hybrid systems 등의 이론과 control and automation of manufacturing systems, synthesis of industrial control networks, supervisory control of semiconductor manufacturing systems, congestion control of communication networks, traffic management in IVHS 등의 응용분야를 포함하는 시스템과학 및 제어공학.

임 종 태

제어 · 자동화 · 시스템 공학회지 제4권 제4호 참조. 현재 한국과학기술원 전기및전자공학과 교수.