이산사건시스템의 관리제어 개요

박 성 진, 조 광 현*
아주대학교 전자공학부, *서울대학교 의과대학 의학과 및 생명공학공동연구원(ckh-sb@snu.ac.kr)

1. 서론

불특정 시간에 발생하는 특정 사건(event)에 의해 상태천이가 유발되는 '이산사건시스템(discrete event dynamic system 또는 discrete event system)'은 그 동역학 특성이 기존의 연속변수시스템(continuous variable dynamic system)과 비교하여 여러 동질성을 지니면서 동시에 차별화된 특성을 지니고 있어서 이산수학, 컴퓨터알고리즘, 생산시스템자동화 등의 분야에서 꾸준히 연구되어져 왔다. 이산사건시스템의 '관리제어(supervisory control)'는 1987년 University of Toronto의 P. J. Ramadge와 W. M. Wonham이 SIAM J. of Control and Optimization에 발표한 3편의 논문[1-3]을 통해 처음 소개되었고 이후 많은 연구자들에 의해 지속적으로 발전되어 왔다[4-12]. 관리제어이론은 분산제어[13-15], 계층제어[16], 고장진단[17], 실시간시스템[18-19], 강인성[20-21] 주제 등을 포함한 시스템이론의 주요 문제들을 다루고, 실제시스템에 대한 응용연구도 함께 이루어지고 있다[22-28]. 현재 관리제어이론은 그동안의 광범위한 연구결과를 통해 이제는 시스템이론의 새로운 영역으로서 이미 그 입지가 확립되었다. 그러나 국내에서는 그동안 인식의 부족으로 인해 이 분야에 대한 연구가 외국에 비해 그다지 활발히 이루어지지 않고 있어 본 논문을 통해 이 분야의 연구를 시작하는 이들에게 분야의 소개와 이론적 토대, 응용범위, 그리고 주요 참고문헌 등을 제공하고자 한다. 관리제어이론은 그 응용에 따라 컴퓨터, 통신망, 집적회로, 생산시스템, 로봇 등의 첨단 시스템들에 널리 활용될 수 있다. 본 논문에서는 먼저 이산사건시스템을 소개하고, 이산사건시스템의 논리적 모델링 방법을 알아본 후, 이산사건모델 상에서 관리제어의 개념을 소개하고 나아가 관리제어기 설계방법을 소개한다. 본 논문은 관리제어이론에 대한 기초적 내용, 분산 및 최적 관리제어 등의 다양한 제어분야 주제들과 광범위한 참고문헌을 집대성하여 소개한 저서[29]를 바탕으로 하고 있다.

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 물리법칙을 따르는 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 연속변수시스템에 대비해 '이산사건시스템'이라고 칭한다.

여기서 사건이란 예를 들면, 생산시스템에 있어서 일(task)의 완성 또는 기계의 고장, 통신네트워크 상의 메시지의 전송, 컴퓨터 프로그램 수행의 종료, 서보제어시스템에 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 이산사건 시스템의 경우는 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 그림 1과 그림 2의 각 시스템의 상태궤적(state trajectory)을 생각해 보자. 그림 1은 '상승'과 '하강'이라는 2개의 버튼만으로 동작되는 간략화 된 3층 엘리베이터시스템에서 현재 엘리베이터의 위치를 상태변수로 하였을 때의 시간에 따른 상태궤적을 나타낸 것이다[4]. 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.

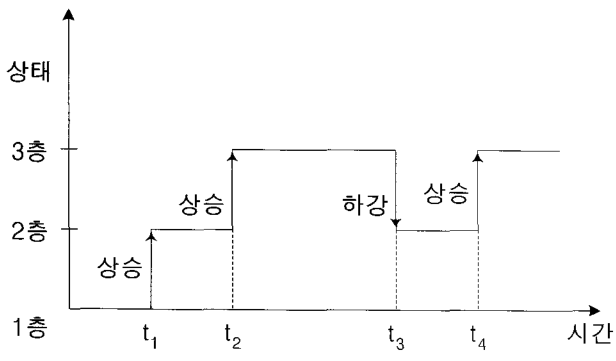


그림 1. 이산사건시스템의 상태궤적의 예[10].

반면, 그림 2에 나타나 있는 전형적인 연속변수시스템의 상태궤적을 보면 상태변수는 R^n상의 임의의 값을 취하며 연속적으로 변화하는 입력(u)과 현재 상태(x)에 따라 연속적으로 변화됨을 알 수 있다[5].

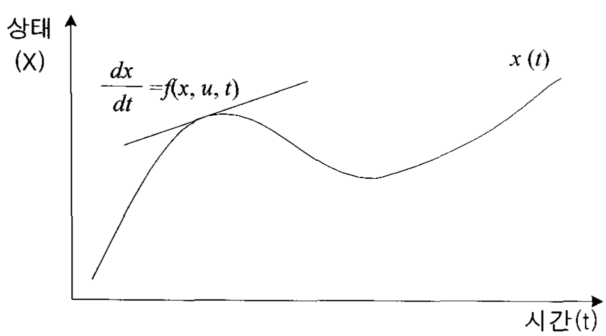


그림 2. 연속변수시스템의 상태궤적의 예[5].

3. 논리적 이산사건 모델

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[8][11][12].

본 논문에서는 대상시스템의 정성적인(qualitative) 동적특성을 잘 반영할 수 있는 논리적 이산사건모델을 이용해 대상 시스템을 모델링하고 동적특성을 기술하는 방법에 대해 다룬다.

3.1. 형식언어

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 $\sigma_i \in \Sigma$ 를 사건이라고 하자. 그러면 $s = \sigma_1\sigma_2 \cdots \sigma_{10}$ 과 같은 유한 사건열을 그 시스템의 자국(trace) 또는 문자열(string)이라고 부르며 그러한 자국들의 모음(collection)을 형식언어(formal language) 또는 언어(language)라고 한다[30]. 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 $s \in \Sigma^*$에 대해 $|s|$는 string의 길이(구성하는 사건의 개수)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[11]. $K, K_1, K_2 \subseteq \Sigma^*$일 때,

- 언어의 합집합(union 또는 choice) :
  $K_1 \cup K_2 := \{ s \in \Sigma^* |\ s \in K_1$ 또는 $s \in K_2\}$
- 언어의 교집합(intersection) :
  $K_1 \cap K_2 := \{ s \in \Sigma^* |\ s \in K_1$ 그리고 $s \in K_2\}$
- 언어의 차집합(difference) :
  $K_1 - K_2 := \{ s \in \Sigma^* |\ s \in K_1$ 그리고 $s \notin K_2\}$
- 언어의 연결(concatenation) :
  $K_1 K_2 := \{ st \in \Sigma^* |\ s \in K_1$ 그리고 $t \in K_2\}$
- 언어의 여집합(complement) :
  $K^c := \Sigma^* - K$
- 언어의 접두닫힘(prefix closure) [또는 닫힘(closure)] :
  $\overline{K} := \{ s \in \Sigma^* | (\exists t \in \Sigma^*) st \in K\}$
- 언어의 클린네닫힘 :
  $K^* := \bigcup_{n \in N} K^n$
  ($K^0 := \{\varepsilon\}$, N은 0을 포함한 자연수 집합, $K^{n+1} = K^n K$)

$(K^*)^* = K^*$, $\emptyset^* = \varepsilon$, $\varepsilon^* = \varepsilon$의 멱등원(idempotent) 성립

- 언어의 몫(quotient) :

  $K_1/K_2 := \{s \in \Sigma^* \mid (\exists t \in K_2) st \in K_1\}$

- 언어의 후언어(post-language) :

  $K_1 \backslash K_2 := \{s \in \Sigma^* \mid (\exists t \in K_2) ts \in K_1\}$

- 언어의 정단힘(positive closure) :

  $K^+ := KK^*$

  ($\varepsilon \in K$이면, $K^+ = K^*$임)

- 언어의 비충돌성(nonconflicting) :

  $\overline{K_1 \cap K_2} = \overline{K_1} \cap \overline{K_2}$이면 K_1과 K_2는 비충돌적

  (예를 들어, $K_1 = \{a, ab\}$, $K_2 = \{a\}$일때 K_1과 K_2는 비충돌적임)

- 언어의 닫힘성(closedness) :

  K가 $\overline{K} \cap M = K$를 만족시키면 M-닫힘성(M-closed)

  (예를 들어, $K = \{aab\}$, $M = \{aab, aabb\}$이면 K는 M- 닫힘성이 있음)

3.2. 오토마타와 유한상태기계

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 유한상태기계(finite state machine)를 소개한다[30]. 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 Σ내의 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 $s \in \Sigma^*$에 의한 상태천이로 다음과 같이 확대 정의될 수 있다. 즉, 모든 $s \in \Sigma^*$에 대해 $\delta: Q \times \Sigma^* \mapsto Q$로 확장된다. 이러한 확장된 개념 위에서 오토마톤 A에 의해 생성된 언어 $L(A)$(생성언어, generated language)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 정의되어 있음을 의미), 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 오토마톤의 경우 $L(A) = \Sigma^*$임을 알 수 있다.

유한상태기계(또는 생성기(generator)) G도 마찬가지로

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

의 5개 원소로 구성된 순서쌍으로 표현되나, 이 경우 δ가 더 이상 전체함수일 필요가 없다는 점이 오토마톤과 다르다. G의 어떤 상태 $q \in Q$에서의 활성(active) 사건집합 $\Sigma_G(q)$는 다음과 같이 정의된다.

$$\Sigma_G(q) := \{\sigma \in \Sigma \mid \delta(q, \sigma)!\}.$$

유한상태기계 G의 경우 $L(G) \subseteq \Sigma^*$이며, 언제나 $L_m(G) \subseteq \overline{L_m(G)} \subseteq L(G)$의 관계를 만족시킨다.

임의의 2개의 유한상태기계 G_1과 G_2 (또는 오토마톤 A_1과 A_2)는 그들이 생성하는 언어와 표기언어가 모두 같을 때, 즉 $L(G_1) = L(G_2)$ 그리고 $L_m(G_1) = L_m(G_2)$ (또는 $L(A_1) = L(A_2)$ 그리고 $L_m(A_1) = L_m(A_2)$)일 때, 동등(equivalent)하다고 한다. 어떠한 오토마톤도 유한상태기계로 구분될 수 있으며, 그 역은 다음과 같이 임의의 상태를 추가함으로써 가능하다: 어떠한 유한상태기계 G가 주어졌을 때 동일한 표기언어를 생성하는 오토마톤 A로 바꾸기 위해서는 다음 조건을 만족하는 임의의 상태 q_{dead}를 추가한다. 즉, $Q_A = Q_G \cup q_{dead}$ 이며, $\delta_A(q, \sigma)$는 만일 $q \in Q_G$ 이고, $\sigma \in \Sigma_G(q)$이면 $\delta_G(q, \sigma)$이고 만일 $q \in Q_G$이면서 $\sigma \notin \Sigma_G(q)$이거나, 또는 $q = q_{dead}$이면 q_{dead}.

4. 관리제어이론

제어 이론의 관점에서 보았을 때 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 서에서 다루고 있는 이산사건시스템에 대해서도 이러한 관점에서 보았을 때 앞장에서 소개한 이산사건모델링을 기초로 시스템의 동적 변화를 원하는 방향으로 바꿀 수 있는 제어 메카니즘이 필요하게 된다. 이 장에서는 바로 이런 목적을 위한 관리제어이론(supervisory control theory) [1-4], [10-11]의 소개와 관리제어기의 존재조건에 따른 성질 및 이의 구현 등을 소개한다.

4.1. 관리제어의 개념

앞에서 소개한 바와 같이 이산사건시스템의 언어모델은 오토마타 또는 유한상태기계 $G=(Q, \Sigma, \delta, q_0, Q_m)$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_c와 제어불가능 (uncontrollable) 사건집합 Σ_{uc}의 서로소 합(disjoint union)으로 가정하자. 즉, $\Sigma=\Sigma_c \cup \Sigma_{uc}$이다. 제어불가능 사건은 물리적으로 시스템의 고장, 하드웨어적 한계, 타이머의 클럭 등을 의미한다. 그러면, 관리제어기(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 정의를 그림 3에 나타내었다.

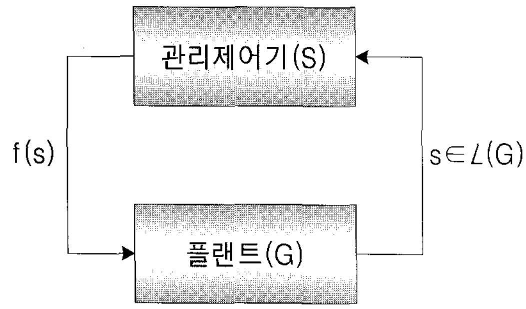


그림 3 관리제어의 개념적 구조[4].

주어진 플랜트 G에 대해 이와 같이 제어규칙에 기반한 관리제어기 S는 플랜트가 생성가능한 모든 언어로부터 제어가능사건 부분집합으로의 사상(map)

$$f: L(G) \mapsto 2^{\Sigma_c}$$

으로 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, 또는 관리제어시스템(supervised control system)의 언어 $L(S/G)$와 표기언어 $L_m(S/G)$는 각각 다음과 같다:

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

즉, 어떤 사건열 s가 관리제어시스템에 의해 생성되고($s \in L(S/G)$), 사건 σ가 플랜트에서 사건열 s다음에 발생하며($s\sigma \in L(G)$), 또한 사건 σ가 사건열 s뒤에 관리제어기에 의해 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

$$\emptyset \subseteq L_m(S/G) \subseteq \overline{L_m(S/G)} \subseteq L(S/G) \subseteq L(G)$$

의 관계식이 성립하며

$$L_m(S/G) \subseteq L(S/G) = \overline{L(S/G)} \neq \emptyset$$

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 확장되지 않을 수 있다. 이 경우 관리제어시스템은 막힘성(blocking)을 가진다고 한다. 그리고 이러한 경우를 항상 배제시킬 수 있는 관리제

어기를 비막힘성(nonblocking) 관리제어기라고 부르며 $\overline{L_m(S/G)} = L(S/G)$을 만족한다. 막힘성은 물리적으로 교착상태(deadlock)와 지속잠금(livelock) 등의 현상을 나타낸다.

앞서 언급한 내용의 관리제어 개념을 플랜트와 관리제어기 유한상태기계들의 동기합성(synchronous composition)이란 측면에서 고찰할 수 있다. $S := (Y, \Sigma, \beta, y_0, Y_m)$를 관리제어기의 유한상태기계이라 하고, 동기합성에 따른 관리제어시스템의 유한상태기계를 $G{\times}S := (Z, \Sigma, \gamma, z_0, Z_m)$이라 하자. 그러면

$$L(G\times S) = L(G) \cap L(S),$$

$$L_m(G\times S) = L_m(G) \cap L_m(S)$$

이므로 동기합성에 의해 플랜트의 동적특성이 제한됨을 알 수 있다. 이와 같은 동기합성에 기반한 관리제어시스템이 앞서 소개한 제어규칙기반 관리제어 시스템과 동일한 제어 결과를 낳기 위해선 관리제어기의 완결성과 $L_m(G\times S) = L(G\times S) \cap L_m(G)$의 두 가지 조건을 만족해야 한다. 첫 번째 조건은 $f: L(G) \mapsto 2^{\Sigma_c}$ 형태의 제어규칙기반 관리제어에서처럼 제어불가능한 사건은 억제시킬 수 없음을 의미하며 두 번째 조건은 관리제어시스템의 표기언어가 플랜트의 표기언어들 가운데 관리제어 후에도 유지되는 언어들의 집합과 같아야 함을 의미한다.

첫 번째 조건은 동기합성기반 관리제어기의 경우, $\forall s \in \Sigma^*$, $\sigma \in \Sigma_{uc}$:

$$s \in L(G\times S),\ s\sigma \in L(G) \Rightarrow s\sigma \in L(G\times S)$$

의 조건식이 만족되면 성립된다. 즉, 관리제어시스템이 사건열 s를 수행한 뒤($s \in L(G\times S)$) 제어불가능한 사건 σ가 사건열 s에 후속해서 발생가능한 경우($s\sigma \in L(G)$), 관리제어기는 이를 허용(enable)해야함($s\sigma \in L(G\times S)$)을 의미한다. 이는 제어불가능한 사건의 발생을 관리제어기가 유발시키는 논리적 모순을 배제하기 위한 것이다[8].

그리고 두 번째 조건의 경우는

$$\begin{aligned} L(G\times S) \cap L_m(G) &= [L(G) \cap L(S)] \cap L_m(G) \\ &= L_m(G) \cap L(S) \end{aligned}$$

이므로, $L_m(G) \cap L(S) = L_m(G\times S)$이면 만족됨을 알 수 있다. 한편, 이러한 동기합성기반 관리제어시스템에 있어서의 비막힘성 관리제어기의 조건은

$$\overline{L_m(G\times S)} = L(G\times S)$$

로 주어짐을 같은 맥락에서 해석할 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 사건집합 Σ_{uo}의 서로소 합(disjoint union)으로 나타낼 수 있다. 즉, $\Sigma = \Sigma_o \cup \Sigma_{uo}$이다. 이 때, 발생 사건열로부터 관측가능사건열로의 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.

앞에서와 같이 관리제어기 S_P의 제어기능을 $f_P: P(L(G)) \mapsto 2^{\Sigma_c}$의 사상으로 정의하면 관리제어시스템의 언어와 표기언어는 각각 다음과 같다.

- $L(S_P/G)$:
  i) $\varepsilon \in L(S_P/G)$,
  ii) $\forall s \in \Sigma^*, \sigma \in \Sigma$,
  $s \in L(S_P/G), s\sigma \in L(G), \sigma \notin f_P(P(s))$
  $\Rightarrow s\sigma \in L(S_P/G)$

- $L_m(S_P/G) := L(S_P/G) \cap L_m(G)$

이와 같은 부분관측하에서의 관리제어개념을 그림 4에 나타내었다.

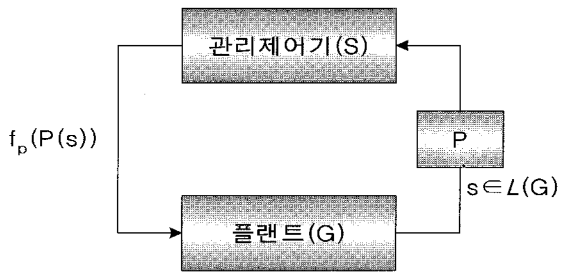


그림 4. 부분관측하에서 관리제어의 개념적구조[7].

지금까지 소개한 관리제어의 궁극적인 목적은 제어대상 이산사건시스템의 여러 가지 가능한 동작형태들로부터 원하는 동작만을 허용하고자 하는데 있다. 그러기 위해선 먼저 원하는 동

작형태, 즉 정상동작(legal behavior)을 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 표기언어만을 포함한 언어를 $L_{am}(G)$로 나타낸다. 주어진 각각의 사양(specification)을 K_{si}, $i \in [1, m]$라고 하면,

$$L_a(G) = L(G) \cap (\bigcap_{i=1}^{m} K_{si}),$$
$$L_{am}(G) = L_m(G) \cap (\bigcap_{i=1}^{m} K_{si})$$

로 각각 주어진다.
그림 5의 간단한 생산시스템의 예를 고려하자.

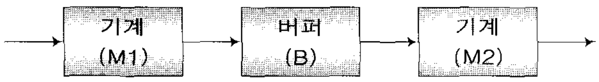


그림 5. 생산시스템의 예: 2대의 기계와 1개의 버퍼[1].

이 경우 버퍼는 1개의 슬롯(slot)만을 가진다고 가정한다. 그리고 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 기계2(M_2)가 먼저 수리되어야 하고, 기계2가 고장일 때는 기계1에 가공물이 투입되어서는 안 된다.

각 기계(M_i, $i=1,2$)와 버퍼의 부분이산사건모델은 그림 6과 같고, 사건 α_i는 'M_i에 가공물 투입', β_i는 'M_i의 가공물 처리완료', λ_i는 'M_i의 고장발생', μ_i는 'M_i의 수리완료'를 각각 나타낸다.

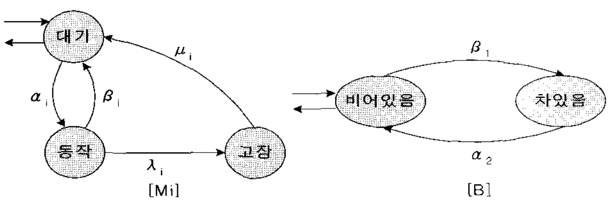


그림 6. 기계와 버퍼의 부분이산사건모델[1].

전체 시스템의 이산사건모델 G는 병렬합성을 통해 $G = M_1 \| M_2 \| B$로 구할 수 있다. 이 경우 사양1에 대한 언어를 K_{s_1}, 사양2에 대한 언어를 K_{s_2}라고 하면 $K_{s_1} = L(H_1)$, $K_{s_2} = L(H_2)$로 각각 주어지며, H_1, H_2는 그림 7과 같다.

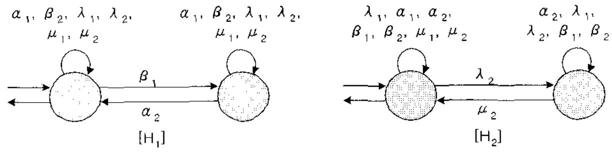


그림 7. 사양1, 2의 인식기[1].

4.2. 관리제어기의 존재조건 I : 제어가능성

지금까지 관리제어의 개념과 관리제어시스템에 관해 언급하였다. 이 절에서는 플랜트와 관리제어스시템의 사양에 대한 언어가 주어졌을 때 이를 만족시키는 관리제어기가 존재할 조건에 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 1) 제어가능성[1]
K와 $M(=\overline{M})$을 사건집합 Σ로부터 정의된 언어일 때

$$\overline{K}\Sigma_{uc} \cap M \subseteq \overline{K}$$

이면 K는 M과 Σ_{uc}에 대해 제어가능이라고 한다.

다음 정리 1은 관리제어시스템의 사양에 대한 언어 K가 제어가능(controllable)인 경우 이를 만족시키는 관리제어기 S가 존재함을 보여준다.

(정리 1) 제어가능성 정리[1]
제어불가능 사건집합 $\Sigma_{uc} \subseteq \Sigma$를 가진 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 성립한다:
i) $\overline{K}\Sigma_{uc} \cap L(G) \subseteq \overline{K}$
ii) K는 $L_m(G)$-닫힘성을 지닌다 (즉, $K = \overline{K} \cap L_m(G)$).

앞의 정리 1은 이산사건시스템 G와 관리제어시스템의 사양 K에 대해 K가 제어가능일 경우 관리제어기 S의 존재여부에 대한 정보를 제공한다. 정리 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.

(보조정리 1)[1]

제어불가능 사건집합 $\Sigma_{uc} \subseteq \Sigma$를 가진 이산사건시스템 G에 대해

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

ii) $K \subseteq L_m(G)$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.

그림 6의 부분이산사건 모델을 가진 생산시스템을 다시 고려하자. 사건집합 Σ가

$$\Sigma = \Sigma_c \cup \Sigma_{uc} = \{\alpha_1, \alpha_2, \mu_1, \mu_2\} \cup \{\beta_1, \beta_2, \lambda_1, \lambda_2\}$$

와 같이 주어졌을 때, 다음의 세 가지 부분 사양들을 모두 만족하는 사양 K는 그림 8의 유한상태기계에서 인식하는 언어가 된다.

- 사양 1: 버퍼(B)의 과잉(overflow) 또는 결핍(underflow)이 없어야 한다.
- 사양 2: 만일 2대의 기계가 모두 고장나면 기계2(M_2)가 먼저 수리되어야 하고, 기계2가 고장일 때는 기계1에 가공물이 투입되어서는 안 된다.
- 사양 3: 두 기계가 모두 동작하고 있을 때 작업이 끝나는 순서는 M_2, M_1 순이어야 한다.

K가 Σ_{uc}와 $L(G)$에 대해 제어가능인지를 살펴보자. 이 경우 $s=\alpha_1\beta_1\alpha_2\alpha_1 \in L(G)$에 대해 $\sigma=\beta_1 \in \Sigma_{uc}$가 후속해서 발생되면 $[s \in \overline{K}] \wedge [\sigma \in \Sigma_{uc}] \wedge [s\sigma \in L(G)]$이지만 $s\sigma \notin \overline{K}$이므로 정의 1로부터 K는 제어불가능이며 따라서 정리 1에 의해 $L_m(S/G)=K$를 만족시키는 비막힘성 관리제어기 S가 존재하지 않음을 알 수 있다.

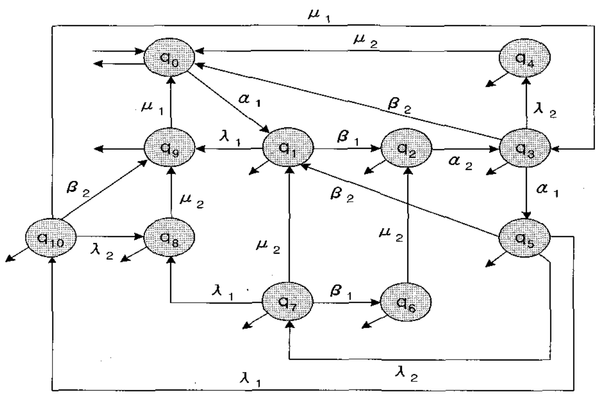


그림 8. 사양 K의 인식기[1].

4.3. 관리제어기의 존재조건 II : 관측가능성

앞 절에서는 플랜트에서 발생하는 모든 사건들이 관측가능한(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 관측불가능한(unobservable) 사건들이 존재하므로, 보다 현실적 상황하에서 관리제어기의 존재조건을 유도하기 위해 K의 관측가능성(observability) [6][7]을 고려할 필요가 있다.

일반적인 관측가능성에 대한 정의를 기술하기 이전에 다음의 $NEXTACT$[13]라는 삼원관계(ternary relation)를 먼저 생각해보자.

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 2) 관측가능성[6][13]

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_o에 대해 관측가능이라고 한다.

그림 9에 보이는 플랜트 모델을 가진 시스템을 고려하자. 사건집합은

$$\Sigma = \{\alpha, \beta, \gamma, \mu_{01}, \mu_{02}\},\ \Sigma_c = \Sigma,\ \Sigma_o = \{\alpha, \beta, \gamma\},$$

플랜트의 생성언어가 $M=\overline{\mu_{01}\alpha(\beta+\gamma)}+\overline{\mu_{02}\alpha\gamma}$, 사양 언어는 $K=\mu_{01}\alpha(\beta+\gamma)+\mu_{02}\alpha$로 각각 주어졌을 때 K가 관측가능인지 살펴보자.

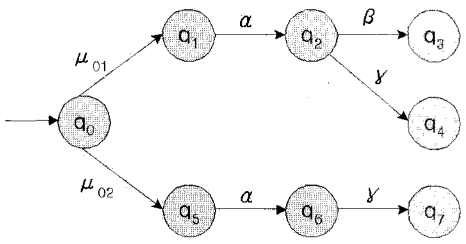


그림 9. 플랜트 모델.

이 경우 예를 들어 $s=\mu_{01}\alpha$, $s'=\mu_{02}\alpha$에 대해 $P(s)=P(s')=\alpha$ 이지만 $\sigma=\gamma$일 때

$$[\mu_{01}\alpha\gamma\in\overline{K}]\wedge[\mu_{02}\alpha\in\overline{K}]\wedge[\mu_{02}\alpha\gamma\in M]$$

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

$$P(s)=P(s')\Rightarrow\forall\sigma\in\Sigma_c,\quad(s,\sigma,s')\in NEXTACT_{K,M}$$

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_P라 하고, $s\in\Sigma^*$ 관측 뒤의 관리제어기능을

$$S_P(P(s)):=\Sigma_G(\delta(q_0,s))-f_P(P(s))$$

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 나타낼 수 있다.

앞의 예에서 보면 K'의 경우 $\mu_{02}\in\Sigma_{uo}\cap\Sigma_c$를 관리제어기가 초기상태에서 억제(disable)함으로써, 즉 $f_P(P(\varepsilon))=\mu_{02}$, 관리제어시스템이 K'를 만족시키게 되는 것이다. 다음은 지금까지 소개한 K의 제어가능성과 관측가능성의 정의로부터 일반적인 상황하에서 관리제어시스템이 K를 만족시키도록 하는 관리제어기가 존재할 필요충분조건을 살펴보자.

(정리 2) 제어가능성 및 관측가능성 정리[6][7]

제어불가능 사건집합 $\Sigma_{uc}\subseteq\Sigma$와 관측가능 사건집합 $\Sigma_o\subseteq\Sigma$를 포함한 이산사건시스템 G에 대해 사양 $K\subseteq L_m(G)$, $K\neq\emptyset$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:

i) K는 $L(G)$와 Σ_{uc}에 대해 제어가능이고
ii) K는 $L_m(G)$-닫힘성을 지니며
iii) K는 $L(G)$, P 그리고 Σ_o에 대해 관측가능이다.

(보조정리 2)[7]

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,

i) 사양 $K\subseteq L(G)$, $K\neq\emptyset$가 주어졌을 때 K가 접두닫힌 언어이고 $L(G)$와 Σ_{uc}에 대해 제어가능이며, $L(G)$, P 그리고 Σ_o에 대해 관측가능이면 $L(S_P/G)=K$인 P-관리제어기 S_P가 존재하고,
ii) 사양 $K\subseteq L_m(G)$, $K\neq\emptyset$가 주어진 경우 K가 $L(G)$와 Σ_{uc}에 대해 제어가능이며, $L(G)$, P, 그리고 Σ_o에 대해 관측가능이면 $L(S_P/G)=\overline{K}$인 P-관리제어기 S_P가 존재하며, 각각 그 역도 성립된다.

4.4. 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 유한상태기계를 이용한 표준구현(standard realization)방식이다. 먼저 모든 사건들은 관측가능하다고 가정하고 $L(S/G)=\overline{K}$, $L_m(S/G)=K$을 만족시키는 관리제어기 S를 생각해보자 (이 때 $K=\emptyset$ 또는 $K=L(G)$의 사소한 경우는 배제하기로 한다). 이러한 K에 대해 $L_m(R)=$ $=L(R)=\overline{K}$인 트림(trim) 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,

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

이고,

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

이 성립한다. 이때 관리제어기능(supervisory control action)은

$$
\begin{aligned}
S(s)\cap\Sigma_G(\delta(q_0,\ s)) &= \Sigma_{R\times G}(\xi\times\delta(x_0, q_0), s)\\
&= \Sigma_R(\xi(x_0, s))
\end{aligned}
$$

이 되어 S의 역할이 인식기 R의 활성(active) 사건집합으로 표현되며 그 결과 S/G의 동적특성이 동기합성된 유한상태기계 $R\times G$의 동적특성과 같게 되는 것이다. 이러한 인식기 R에 의한 관리제어기 구현방식을 S의 표준구현이라고 한다.
그림 6의 플랜트 G에 대해

$$\Sigma=\Sigma_c\cup\Sigma_{uc}=\{\alpha_1, \alpha_2, \mu_1, \mu_2\}\cup\{\beta_1, \beta_2, \lambda_1, \lambda_2\}$$

와 사양1, 사양2를 모두 만족시키는 언어 K가 주어졌을 때 $L(S/G)=\overline{K}$, $L_m(S/G)=K$를 만족시키는 관리제어기 S의 구현을 생각해보자. K는 그림 10의 유한상태기계가 인식하는 언어가 됨을 알 수 있다. 이 때 K는 제어가능이고 $K=\overline{K}\cap L_m(G)$, 즉 $L_m(G)$-닫힘성을 지님을 쉽게 확인할 수 있으며, 정리 1로부터 $L_m(S/G)=K$인 비막힘성 관리제어기 S가 존재함을 알 수 있다. 이 경우 $L_m(R)=L(R)=\overline{K}$인 그림 10의 인식기 R로부터 $R\times G$에 의해 앞서와 같이 S/G의 동적특성을 보일 수 있다.

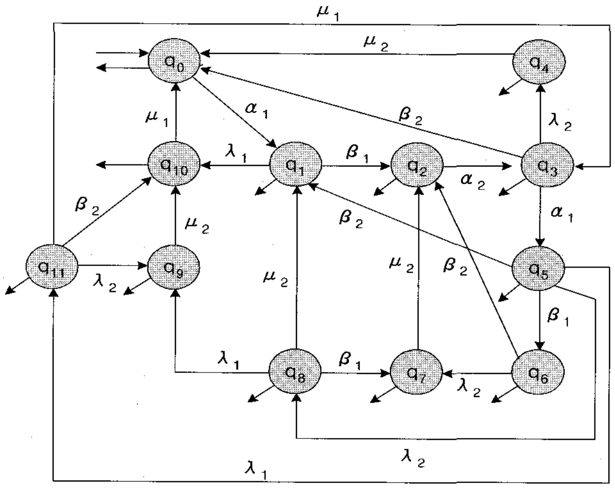


그림 10. 인식기 R : $L_m(R)=L(R)=\overline{K}$[1].

인식기 R의 각 상태에서 활성 사건집합내의 사건들에 대한 관리제어기능은 표 1과 같다.

표 1. 관리제어기능 : $S(s)\cap\Sigma_G(\delta(q_0, s))$.

상태	허용사건	억제사건	상태	허용사건	억제사건
q_0	α_1	α_2	q_6	β_1, λ_2	α_1
α_1	β_1, λ_1	α_2	q_7	μ_2	α_1
q_2	α_2	α_1	q_8	β_1, λ_1, μ_2	$\emptyset$
q_3	α_1, β_2, λ_2	$\emptyset$	q_9	μ_2	μ_1
q_4	μ_2	α_1	q_{10}	μ_1	α_2
q_5	β_1, β_2, λ_1, λ_2	$\emptyset$	q_{11}	μ_1, β_2, λ_2	$\emptyset$

그러면 $R\times G$로부터 $L(S/G)=\overline{K}$이고 $L_m(S/G)=K$임을 알 수 있다.

그림 11의 플랜트 G의 인식기와 사건집합 Σ가 $\Sigma=\Sigma_c\cup\Sigma_{uc}=\Sigma_c\cup\Sigma_{uc}=\{\beta\}\cup\{\alpha, \gamma\}$와 같이 주어졌을 때, $K=\overline{(\alpha\beta)^*+(\alpha\beta)^*\alpha\gamma^*}$에 대해 $L(S/G)=K$를 만족시키는 관리제어기 S의 구현을 생각해보자[1]. 이 경우

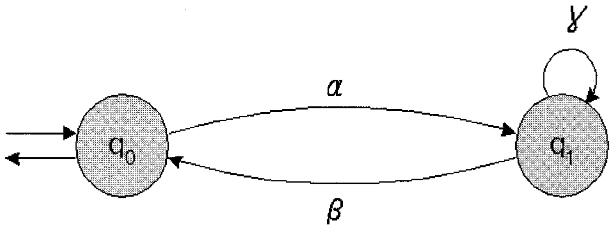


그림 11. 플랜트 G의 인식기[1].

$$L(G)=\overline{(\alpha\gamma^*\beta)^*}=\overline{L_m(G)}$$

이고, $K=\overline{K}\subseteq L(G)$이며 K가 $L(G)$와 Σ_{uc}에 대해 제어가능이므로 보조정리 1로부터 $L(S/G)=K$를 만족시키는 S가 존재함을 알 수 있다.
그러한 관리제어기 S의 구현을 위해

$$L_m(R)=L(R)=\overline{K}$$

을 만족하는 그림 12의 인식기 R과 표 2에 나타나 있는 R의 각 상태에서 활성 사건집합내의 사건들에 대한 관리제어기능을 생각해 보자.

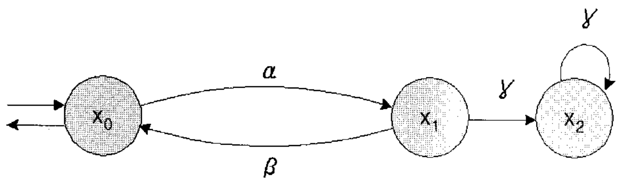


그림 12. 인식기 R[1].

표 2. 관리제어기능

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

그러면 $R\times G$에 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, $\alpha\in\overline{K'}$, $\gamma\in\Sigma_{uc}$에 대해 $\alpha\gamma\in L(G)$이지만 $\alpha\gamma\notin\overline{K'}$이므로 K'는 제어불가능이어서 보조정리 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.

5. 결론

본 논문에서는 이산사건시스템 모델링과 관리제어에 대한 기본적인 개념과 이론을 소개하였다. 주지해야할 사항은 이산사건시스템의 관리제어가 전통적으로 연구되어져 온 연속변수 시스템의 제어기법과 매우 흡사하다는 것이다. 이러한 이산사건시스템의 관리제어는 현재 급부상하고 있는 임베디드 시스템, 로봇제어, 유비쿼터스 제어시스템, 네트워크제어 등의 측면에서 매우 유용하게 이용될 수 있다.

이산사건시스템의 관리제어에 관한 연구논문은 IEEE Trans. Automatic Control을 비롯하여 Automatica, Systems & Control Letters, IEE Proc. Control Theory and Applications, 그리고 Discrete Event Dynamic Systems: Theory and Applications 등의 저널에 주로 수록되고 있으며 관련 학술대회로는 WoDES, CDC, ACC, MTNS 등이 있다.

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

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

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

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

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

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

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

[11] C. G. Cassandras and S. Lafortune, Introduction *to Discrete Event Systems*, Kluwer Academic Pub., 1999.

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

[13] K. Rudie and W. M. Wonham, "Think Globally, Act Locally : Decentralized Supervisory Control," *IEEE Trans. on Automatic Control*, vol. 37, pp. 1692-1708, 1992.

[14] K.-H. Cho and J.-T. Lim, "Mixed centralized/decentralized supervisory control of discrete event dynamic systems," *Automatica*, vol. 35, no. 1, pp. 121-128, 1999.

[15] T.-S. Yoo and S. Lafortune, "A general architecture for decentralized supervisory control of discrete event systems," *Discrete Event Dynamic Systems: Theory and Applications*, vol. 12, no. 3, pp. 335-377, 2002.

[16] R. J. Leduc, B. A. Brandin, M. Lawford, and W. M. Wonham, "Hierarchical interface-based supervisory control-part I: serial case," *IEEE Trans. on Automatic Control*, vol. 50, no. 9, pp. 1322-1335, 2005.

[17] M. Sampath, R. Sengupta, S. Lafortune, K. Sinnamohideen, and D. Teneketzis, "Failure diagnoesis using discrete-event models," *IEEE Trans. on Control Systems Technology.*, vol. 4, pp. 105-124, 1996.

[18] B. A. Brandin and W. M. Wonham, "Supervisory control fo timed discrete event systems," *IEEE Trans. on Automatic Control*, vol. 39, no. 2, pp. 329-342, 1994.

[19] S.-J. Park and K.-H. Cho, "Supervisory control of timed disctete event systems under partial observation based on activity models and eligible time bounds," *Systems and Control Letters*, In Press, 2006.

[20] S.-J. Park and J.-T. Lim, "Robust and non-blocking supervisor for discrete event systems with model uncertainty under partial observation," *IEEE Trans. on Automatic Control*, vol. 45, no. 12, pp. 2393-2396, 2000.

[21] S.-J. Park and J.-T. Lim, "Robust and non-blocking supervisory control of nondeterministic discrete event systems using trajectory models," *IEEE Trans. on Automatic Control*, vol. 47, no. 4, pp. 655-658, 2002.

[22] 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.

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

[24] 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.

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

[26] S.-J. Park and J.-T. Lim, "Fault-tolerant robust supervisor for discrete event systems with model uncertainty and its application to a workcell," *IEEE Trans. on Robotics and Automation*, vol. 15, no. 2, pp. 386-391, 1999.

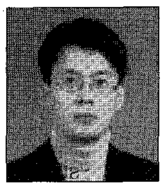
[27] K.-H. Cho and J.-T. Lim, "Multiagent supervisory control for anti-fault-propagation in serial production systems," *IEEE Trans. on Industrial Electronics*, vol. 48, no. 2, pp. 460-466, 2001.

[28] V. V. Phoha, A. U. Nadgar, A. Ray, and S. Phoha, "Supervisory control of software systems," *IEEE Trans. on Computers*, vol. 53, no. 9, pp. 1187-1199, 2004.

[29] 임종태, 조광현, 박성진, 이산사건시스템의 관리제어, KAIST Press & Brain Korea, 2003.

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

저자약력



《박 성 진》

- 1994년 KAIST 전기및전자공학과 졸업.
- 동대학원 석사(1997).
- 동대학원 박사(2001).
- 2001년~2004년 삼성전자 네트워크 사업부 책임연구원.
- 2004년~현재 아주대학교 전자공학부 조교수.
- 관심분야 : 이산사건시스템의 관리제어, 임베디드시스템, 반도체공정자동화, 통신네트워크제어.

《조 광 현》

• 1993년 KAIST 전기 및 전자공학과 졸업. 동대학원 석사(1995).
• 동대학원 박사(1998) 및 박사 후 연구원 (1998).
• 1999년~2004년 8월 울산대학교 전기전자정보시스템공학부 전임강사 및 조교수.
• 2002년 6월~2003년 8월 영국 UMIST 제어시스템센터 초빙교수.
• 2003년 12월~2004년 3월 스웨덴 Royal Institute of Technology 자동제어그룹 초빙교수.
• 2004년 6월~2004년 8월 아일랜드 Hamilton Institute 초빙교수.
• 2004년 9월~현재 서울대학교 의과대학 의학과 조교수 및 서울대학교 생명공학공동연구원(Bio-MAX Institute, 관악캠퍼스 소재) 겸임 - 시스템생물학연구실 운영.
• 2004년~현재 국제저널 'Systems Biology (IEE, 영국런던)'의 Editor-in-Chief 역임.
• 2006년부터 국제저널 'Synthetic and Systems Biology (BMC, 영국런던)'의 Editor 위임.
• 관심분야 : 비선형동역학, 이산사건시스템제어이론, 제어공학의 생명과학 응용 (시스템생물학) 및 생명과학에서 발견되는 자연현상의 공학적 응용 (생명모사공학).
• E-Mail : ckh-sb@snu.ac.kr