

이산사건 시스템의 실시간 모델링 및 응용

Real Time Modeling of Discrete Event Systems and Its Application

정용만 · 황형수*

Yong Man Jeong and Hyung Soo Hwang

(주)G.C.TECH 연구소, *원광대학교 제어계측공학과

ABSTRACT

A DEDS is a system whose states change in response to the occurrence of events from a predefined event set. A major difficulty in developing analytical results for the systems is the lack of appropriate modeling techniques. In this paper, we consider the modeling and control problem for Discrete Event Dynamic Systems(DEDS) in the Temporal Logic framework(TLF) which have been recently defined. The traditional TLF is enhanced with time functions for real time control of Discrete Event Dynamic Systems. A sequence of event which drive the system from a given initial state to a given final state is generated by pertinently operating the given plants. This paper proposes the use of Real-time Temporal Logic as a modeling tool for the analysis and control of DEDS. An given example of fixed-time traffic control problem is shown to illustrate our results with Real-time Temporal Logic Framework.

1. 서 론

많은 대규모의 동적 시스템들은 이산사건 시스템 구조(Discrete Event Systems Framework)[1,2]로 이루어져 있다. 이러한 시스템들은 기존의 제어 및 시스템 이론[6,7]으로 취급될 수 있는 연속변수 동적 시스템(CVDS: Continuous Variable Dynamical Systems)처럼 상미분 방정식이나 편미분 방정식에 의해 처리될 수 없으며 이산적인 상태와 이 상태들 사이에서 사건의 천이관계를 나타냄으로써 표현할 수 있다. 이러한 시스템으로 생산 시스템(manufacturing systems), 교통 시스템(traffic systems), 일괄처리(batch processing), 통신 시스템(communication systems)[8], 전문가 시스템(expert system)[9] 등을 예로 들 수 있다. 이런 시스템을 이산사건 동적 시스템(DEDS: Discrete Event Dynamical Systems)[1]이라고도 부른다. 대조적으로 상미분 방정식이나 편미분 방정식에 의해 표현될 수 있는 연속변수 동적 시스템(CVDS)의 체계는 연속적이다.

DEDS는 이산시간에 사건이 발생하고 그 사건에 의해 동작하며, CVDS는 일반적으로 연속적인 입력이나 클럭(clock)에 의해 동작한다. DEDS에서 상태는 송신을 기다리는 통신망에서 각 마디(node)에 대한 메시지의 수와 같이 물리적인 상태를 말하는 반면

CVDS에서의 상태는 추상적 의미에서 수학적 상태를 말한다.

최근 이산사건 시스템에 대한 많은 연구가 이루어지고 있으며, 모델링 방법으로는 Automata and formal language[7], Petri-net[1,8,9], Finitely Recursive Process, Mini-Max Algebra[1], Temporal Logic Framework[3,10]등이 제안되었다. 그러나 미분 방정식과 편미분 방정식 등에 의해 다루어 질 수 있는 선형 또는 비선형 연속 시스템들의 최적제어 이론, 안정도 이론, 가제어성과 가관측성 이론 등의 이론에 비해 이산사건 시스템을 위한 효과적인 이론은 아직까지 존재하지 않는다. 또한, 실시간 제어문제에 관련된 많은 연구가 이루어지지 않았으며 최근 이러한 문제들을 해결하기 위한 많은 연구가 활발히 진행되어지고 있다.

이러한 이론들 중에 시간논리구조는 일반 논리구조에 시간 개념을 부가한 구조로 시간관계를 논리적으로 표현한 이론으로 정의되었기 때문에 이산사건 시스템의 동작을 표현하기에 용이하고 시간관계를 논리적으로 표현함으로써 사건의 전후관계와 다음 상태, 최종 상태 등을 표현하기에 편리하다.

그러나 현재까지 주로 연구되어진 이론들은 실시간이라는 개념이 배제되어있다.

*이 논문은 '97년도 원광대학교 교비지원에 의해서 연구됨

실시간 문제를 고려하여 이산사건 시스템을 실시간으로 제어하기 위한 한 방법으로 시간논리구조에 실시간 개념을 정의하고 시스템을 모델링 하는 방법을 제안하였다. 이 이론의 실용성을 보이기 위해 이산사건 시스템의 실시간 제어문제 중에서 고정시간 문제를 다루었다. 고정시간 문제는 실시간 문제의 기본 함수인 시간함수로 표현되며, 한번 설계된 제어기는 시간에 따라 변하지 않는 특성을 갖고 있다. 이 이론을 다루기 위해 교차로의 교통 신호등문제를 예로 들어 시간논리구조로 모델링하였으며, 모델링 방법에 의한 모의실험결과와 이산사건 시스템의 제어를 위한 편리한 방법임을 확인하였다.

2. 시간 논리 구조

시간논리(Temporal Logic)는 물리적 또는 프로그램에서의 상태가 시간에 따라 변하는 궤적을 수식적으로 표현하려는 방법으로 제안되었다. 시간논리구조는 시간개념을 포함한 술어논리의 확장된 형태이며, 시간 순차열에 따른 추론 지향적인 논리이다.

시간논리는 일반 논리를 이용하고 시제 표현식으로 표현 될 수 있으며 Allen's Properties-Event-Process[11]와 Mcdermott's Fact-Event[12]의 시제추론을 이용하였다.

이 논리는 주로 software 증명에 이용되었으며 최근에는 이산사건 시스템의 제어문제에 적용되었다.

2.1 시간논리구조의 기호와 표현

시간논리구조는 시간과 시간에 대한 추론을 할 수 있는 형식구조이다. 시간논리구조는 전형적인 Boolean 연결자인 \neg (not), \wedge (and), \vee (or), \rightarrow (implies), \leftrightarrow (if and only if)를 포함하며 시간의 변화량을 표현하기 위한 시간 연산자 \square (henceforth), \diamond (eventually), \circ (next), U (until), P (proceed) 등을 가진다[3,5,10]. 이들의 정의는 다음과 같다.

정의 2.1.1 : Boolean 연결자와 시간 연산자

• 부울 논리 연산자

- (1) $\neg(\phi)$: ϕ 는 참이 아니다.
- (2) $(\phi \wedge \psi)$: ϕ 가 참이고 ψ 도 참일 때 참이다.
- (3) $(\phi \vee \psi)$: ϕ 가 참이거나 ψ 가 참일 때 참이다.
- (4) $(\phi \rightarrow \psi)$: ϕ 가 참이면 ψ 가 참이다.
- (5) $(\phi \leftrightarrow \psi)$: ϕ 가 참이면 ψ 가 참이고 ψ 가 참이면 ϕ 가 참이다.

• Temporal 연산자

- (1) $(\square \phi)$: 지금부터 ϕ 는 참이다
- (2) $(\diamond \phi)$: 미래에 ϕ 가 참이 되는 점이 있다.

(3) $(\circ \phi)$: 다음 번에 ϕ 는 참이 된다.

(4) $(\phi U \psi)$: 현재 ϕ 가 참이고, 미래에 ψ 가 참이 되는 점이 존재한다면, 그때까지 ϕ 는 참일 것이다.

(5) $(\phi P \psi)$: ψ 가 참이기 전에 ϕ 는 참이어야 한다.

다음은 연산자의 전형적인 축약형을 나타낸다.

- (1) 함의(조건문) : $\phi \rightarrow \psi \equiv (\neg \phi \vee \psi)$
- (2) 등위 접속 연산자 : $\phi \wedge \psi \equiv (\neg(\neg \phi \vee \neg \psi))$
- (3) 쌍 조건문 : $\phi \leftrightarrow \psi \equiv (\phi \rightarrow \psi) \wedge (\psi \rightarrow \phi)$
- (4) eventually 연산자 : $(\diamond \psi) \equiv (\neg(\square(\neg \psi)))$
- (5) precedes 연산자 : $(\phi P \psi) \equiv (\neg(\neg(\neg U \psi)))$

2.2 공정 및 제어기의 시간논리구조식

Lin과 Ioncscu[3]에 의한 동적식(dynamic formula) $l(e,s)$ 은 식 (1)과 같은 형태를 가진다.

$$\square [\delta = e \wedge x = R \rightarrow (\circ x) = Q] \quad (1)$$

이것은 한 상태 $s=x$ 와 다음 상태 $s'=f(e,s)=\circ x$ 에 의한 천이를 나타낸다. 이 식은 사건 e 의 발생에 의해 원소 x 가 상태 R 에서 상태 Q 로 변화함을 의미한다.

공정에서 시간논리구조 모델은 식 (2)와 같이 6-변수를 가지는 모델 M 으로 정의된다.

$$M = (S, E, F^*, f, s_0, l) \quad (2)$$

여기서, S 는 상태 집합이고, E 는 사건의 집합이다. 각 사건은 한 상태에서 다음 상태로의 천이됨을 나타낸다. 또한, s_0 는 초기상태, f 는 발화함수(firing function), F^* 는 논리식의 집합이며 l 은 E 에서 F^* 로의 라벨링 함수(labeling function)이다. $\forall e \in S$ 와 $\forall e \in f(S)$ 에 대해 S 에서 E^* 로의 사상(mapping)을 나타내는 함수로써 $e(s) = f(s,e)$ 가 정의된다. 여기서 $f(s)$ 는 상태 s 에서의 사건 발화(event firing) 집합이다.

$e(s) = f(s,e) = s'$ 은 한 상태에서 다음 상태로의 천이를 나타내므로 발화함수 f 를 실시간 프로그래밍에서 동적 동작으로 주어지는 다음상태 함수라고도 한다. 시간논리구조에서 한 상태를 정확하게 한 사건의 실행에 대응한다. 이 결과를 나타내기 위한 식은 식 (3)과 같다.

$$\square (s_i(x) \wedge e_{i+1}(x) \rightarrow s_{i+1}(x)) \quad (3)$$

여기서 $s_i(x)$ 는 x 의 현재상태, $e_{i+1}(x)$ 는 x 에 대응하는 사건의 발생을 나타내며, $s_{i+1}(x)$ 는 다음 상태를 표현한다.

3. 실시간 시간논리구조

이산사건 시스템을 모델링하고 제어하는 기존의 시간논리구조의 특징은 일반 이진논리구조에 시간

연산자를 도입하여 상태나 사건에 대한 시간의 전후 관계를 표현할 수 있다. 즉, 한 사건이 발생한 후 상태가 어떻게 변화될 지를 나타내고 결국 나타날 상태나 발생하게 될 사건을 알려줌으로써 유용하게 사용될 수 있다. 그러나 한 사건이 발생한 후 언제 상태가 변화할 지 또는 결국 나타나게 될 상태나 사건이 언제 나타나게 되는 지에 대한 표현이 불가능하다. 따라서 이런 단점을 보완하기 위하여 이장에서는 기존의 시간논리구조에 실시간 이론을 도입하여 실시간 시간논리구조식을 정의하였다.

3.1 실시간 시간논리구조식

이산사건 시스템을 제어하고자 할 때 한 사건 e_i 에 의해 한 상태에서 다음 상태로 상태가 천이 되는 동안 시간 t_i 가 흐르게 된다. 이 시간은 상황에 따라 다르다. 이 천이시간 t_i 동안 또 다른 사건 e_j 가 발생할 수 있다. 이때 현재의 상태 변화를 무시하고 사건 e_j 에 의해 상태가 변화할 수도 있으며, 새로운 사건 e_k 를 무시하고 현재의 상태 변화가 계속 될 수 있다. 또한 사건이 발생했을 때 상태의 천이 시간이 상한 시간 (T_u) 내에 종료 되어야 하는 경우가 있고, 또 다른 상황에서는 임의의 하한 시간 (T_l) 내에 종료되어서는 안되는 경우가 있다. 이러한 상황은 기존의 전형적인 시간논리구조로 표현할 수 없다. 하지만, 중요한 임계점에 놓여있는 시스템을 제어하고자 할 때 실시간 제어는 매우 중요한 문제이다. 이런 플랜트를 모델링하기 위하여 전형적인 시간논리구조에 시간함수 t 를 도입한 식(4)와 같은 실시간 시간논리구조식을 제안한다.

$$M_t = (S, E, F^*, f, s_0, l, t) \tag{4}$$

여기서 t 는 시간함수이고 다음과 같이 정의한다.

정의 3.1.1

$t(T_l; T_u)$: 사건 발생 후 상태 천이에 필요한 하한시간 T_l 과 상한시간 T_u 를 표시한다. 하한시간 또는 상한시간만 필요할 경우가 있다. 이 때는 각각 $t(T_l;)$ 과 $t(, T_u)$ 로 표시한다.

정의 3.1.2

$t(T)$: 상태의 천이시간이 정확하게 주기적이거나 그럴 필요가 있을 경우의 시간을 표시한다.

정의 3.1.3

정의 3.1.1과 3.1.2에서 정의한 시간함수는 시간논리구조식과 \wedge (and)로 연결될 수 있고, 예를 들어 $(\frac{t(T_l; T_u)}{\uparrow})$ 와 같이 함의(Implication)기호 위에 표시한다.

또한, 상태 천이 중에 다른 사건이 발생했을 경우

사건을 받아들여 그 사건에 의해 플랜트나 제어가 동작하는 경우와 그 사건을 무시할 경우를 구분할 필요가 있다. 이러한 필요성에 따라 다음을 정의한다.

정의 3.1.4

상태 천이 중에 다른 사건의 발생에 의해 플랜트나 제어가 동작하는 경우 ($\overline{\uparrow}$)으로 표기한다. 또한, 상태 천이 중 다른 사건의 발생을 무시하고 계속 상태 천이가 진행되는 경우는 함의기호(\rightarrow)를 사용한다.

다음 식 (5)는 이러한 기호들의 사용 예이다.

$$S_i(x) \wedge e_{i+1}(x) \rightarrow S_{i+1}(Ox) \wedge t(T_l; T_u)$$

$$S_i(x) \wedge e_{i+1}(x) \rightarrow S_{i+1}(Ox) \wedge t(T)$$

$$S_i(x) \wedge e_{i+1}(x) \overline{\uparrow} S_{i+1}(Ox) \wedge t(T)$$

$$S_i(x) \wedge e_{i+1}(x) \xrightarrow[\uparrow]{t(T_l; T_u)} S_{i+1}(Ox)$$

이러한 기호들을 이용하여 중요한 시간적 제약을 표현함으로써 이산사건 시스템을 실시간으로 모델링할 수 있다.

4. 교통제어 모델링

실시간제어의 예로서 교통제어를 다루었다. 교통신호기제어문제는 고정된 시간에 상태가 변화하는 이산사건시스템이므로 이 문제를 시간논리구조를 이용하여 모델링하고 제어하고자 한다. 즉, 이 시스템은 고정시간 이산사건 시스템의 제어문제으로써 시간함수를 고려한 실시간 시간논리구조를 이용하여 모델링한다.

교통제어문제를 모델링하기 위해서 먼저 신호등이 가지는 제약조건들을 표현해야 한다.

4.1 교통 신호의 모델링

우리가 고려하는 문제는 신호의 시간주기를 결정하는 것인데 각 신호의 한 주기는 적색과 녹색 신호의 시간에 의해서 결정된다. 미리 결정된 교통흐름의 양이나, 안전성 판단을 보장하기위하여 각 신호의 녹색지속시간의 하위 경계가 주어져야 한다. 신호의 계획이 결정되면 신호제어기는 계획된 주기 신호만을 엄격하게 반복한다. 하위경계는 또한 녹색지속시간의 어떤 목적함수에 영향을 미치는 경우 적절하게 조절하기 위한 목표로 사용되기도 한다. 실제적으로 신호의 한 주기와 더불어 녹색 지속시간은 교통의 연결량을 결정하는데 이것은 시스템의 성능평가에 관련된 중요한 요소이다. 더욱이 녹색지속시간의 하위 경계는 적색 지속시간의 상위 경계와 대응한다. 또한 녹색신호의 상위 경계는 보행자의 횡단이나, 하류의 밀집을 방지하기 위하여 흐름을 약간 감소시킬 경우등

을 적절하게 유지되어야 한다. 우리는 이러한 요구를 신호제약(signal constraints)이라 부른다.

또한 상반된 교통이동을 제어하는 신호는 안정한 상태에 따라야 하고 그들의 녹색시간들 사이에 적당한 여유시간이 삽입되어야 한다. 이러한 제약을 충돌 제약(conflict constraints)이라 부른다.

그리고 같은 흐름에 의해 순차적으로 교차되는 정지선의 신호는 흐름율이 충분히 높은경우에 교통흐름의 과잉에 의해 발생하는 전파 방해를 피하기 위하여 조합되어야 하며, 또한 효과적인 교통흐름을 보장하거나 정지하는 횡수를 감소시키는등의 최적화를 위하여 적절하게 사용되어야 한다. 이 조건을 조합제약(coordination constraints)이라 부른다.

교통신호등을 모델링하기 위해 먼저 신호등의 상태와 사건들을 정의해야 한다. 여기에서 정의할 수 있는 사건은 신호등이 변하는 그 때의 시간으로 정의한다. 한 신호등이 가질 수 있는 상태와 사건 그리고 시간함수를 다음과 같이 정의할 수 있다. 여기서 각 신호등은 방향 그래프에서 각 마디로 표현할 수 있다.

- $s(i)$: i 번째 마디의 녹색 상태
- $s(i+r)$: i 번째 마디의 적색 상태
- $e(i)$: i 번째 마디의 녹색 사건(또는 그 때의 시간)
- $e(i+r)$: i 번째 마디의 적색 사건(또는 그 때의 시간)
- $t_{ij}(T)$: 사건 $e(i)$ 의 발생에서 사건 $e(j)$ 가 발생할 때까지의 시간
- 즉, $t_{ij}(T) = e(j) - e(i)$ 이고 다음과 같이 나타낸다.
- $e(i) \xrightarrow{t_{ij}(T)} e(j)$
- $t_{ij}(T_i; T_u)$: 사건 $e(i)$ 가 발생한 후 사건 $e(j)$ 가 발생할 수 있는 최소 시간 (T_i)과 최대 시간 (T_u)

이와 같은 정의를 이용하여 다음과 같은 제약조건을 시간논리구조로 표현할 수 있다.

4.1.1 신호 제약(signal constraints)

마디 i 와 $i+r$ 을 고려할 때 사건 $e(i)$ 와 사건 $e(i+r)$ 가 발생할 수 있는 사건사이의 시간 $t_{i,i+r}(T)$ 라하며 이를 녹색 지속시간이라 한다. 녹색 지속시간의 최소시간을 T_i 이라 하고 최대시간을 T_u 라 하면 다음 식(6)과 같이 신호제약을 나타낼 수 있다.

$$\text{조건 1: } \square \left(e(i) \xrightarrow{t_{i,i+r}(T_i; T_u)} e(i+r) \right) \quad (6)$$

4.1.2 충돌 제약(conflict constraints)

신호 i 와 j 가 충돌상태에 있을 때, $t_{i,r,j}(T)$ 을 신호 j 의 녹색 사건과 신호 i 의 적색 사건 사이 시간으로 정의하고 다음 식(7)과 같이 충돌제약을 나타낼 수 있다.

$$\text{조건 2: } \square \left(e(i+r) \xrightarrow{t_{i,r,j}(T_{\alpha}; \cdot)} e(j) \right) \quad (7)$$

여기에서 T_{α} 는 신호 i 와 j 사이에 삽입되어지는 최소 여유시간(minimum clearance time)이다. 최소 여유시간은 두 신호가 공유하는 영역을 서로 점유하지 않는 시간이고 보통 3초 정도 여유를 준다. 비슷하게 식(8)과 같은 다음 조건을 얻을 수 있다.

$$\text{조건 2': } \square \left(e(i+r) \xrightarrow{t_{i,r,o_i}(T_{\alpha}; \cdot)} e(o_i) \right) \quad (8)$$

교통문제를 시각적으로 표현하기 위하여 신호에 대응하는 절점과 관련된 신호들 사이의 충돌을 나타내는 무향 호와 상향 흐름으로 부터 하향 흐름에 대응하는 유향 호를 이용하여 상황을 표현하기 위하여 교차로의 축소된 그래프와 확장된 그래프를 이용한다[13]. 두 신호의 충돌 회로를 축소 그래프와 확장 그래프로 표현할 수 있다. 그림 1(b)의 확장 그래프는 각 신호등의 적색사건 후 다음 신호등의 녹색사건 전에 여유시간 (T_{α})이 포함됨을 보인다. 여기서 최소 여유시간은 일정 (T_{α}) 하다고 가정한다.

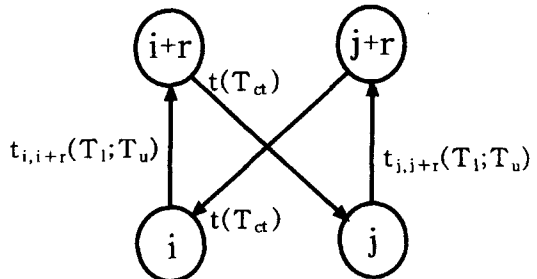
두 신호의 충돌 회로에서 사건의 순차적 발생과 조건은 식 (9)와 같다.

$$\square \left(e(i) \xrightarrow{t_{i,i+r}(T_i; T_u)} e(i+r) \xrightarrow{t_{i,r,j}(T_{\alpha})} e(j) \xrightarrow{t_{j,r,o_j}(T_{\alpha})} e(o_j) \right) \quad (9)$$

여기서 $e(i)$ 에서 $e(j)$ 까지의 순회시간은 주기 (T_f) 이고 식 (10)과 같이 나타낼 수 있다.



(a) 축소 그래프



(b) 확장 그래프

그림 1. 간단한 충돌 회로의 축소 그래프(a)와 확장 그래프(b).

$$\square \left(e(i) \xrightarrow{t_{oi}(T_f)} e(o_i) \right) \quad (10)$$

이 충돌 회로는 3거리 또는 그 이상의 교차로에 확대 적용할 수 있다.

4.1.3 조합 제약(coordination constraints)

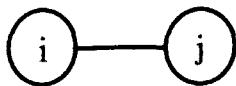
같은 흐름에 의해 순차적으로 교차된 정지선의 신호는 흐름 율이 충분히 높은 경우에 열의 과잉에 의해 생기는 차량 소통 방해를 피하기 위해 조합되어야 한다. 또한 조합 제약은 원활하고 효과적인 교통흐름을 보장하거나 정지하는 횟수를 감소시키기 위한 최적의 목적을 위해 사용된다.

하류신호의 녹색간격이 적절한 기준에 의해 전환된 상류신호의 간격에 대응하는 것을 요구함으로써 아주 일반적인 항에서 조합이 부가된다. 이런 조합은 하류신호에서 상류신호의 이동시간을 고려하고 이미 존재하는 열을 비우는 시간을 고려해야함으로 두 신호의 조합 시간은 이동시간과 같거나 작아야 한다. 이 조합 제약은 식(11)로 나타낼 수 있다.

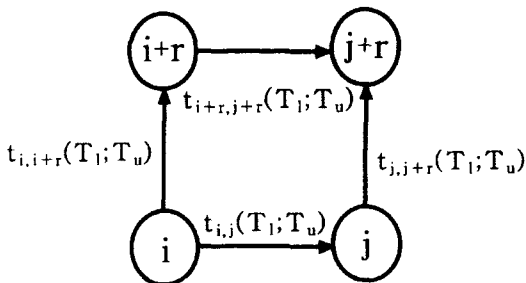
$$\text{조건 3: } \square \left(e(i) \xrightarrow{t_{ij}(T_i; T_u)} e(j) \right) \\ \square \left(e(i+r) \xrightarrow{t_{i+r,j+r}(T_i; T_u)} e(j+r) \right) \quad (11)$$

조합 제약의 축소 그래프와 확장 그래프는 각각 그림 2(a)와 그림 2(b)에 보였다. 그림 2(a)의 축소 그래프는 교통 신호의 흐름 방향을 나타낸다. 그림 2(b)의 확장 그래프를 한 신호등의 녹색사건 후 다음 신호등의 녹색사건 사이에 일정한 시간이 필요함을 보인다.

4.2 고정시간 교통 신호등 모델링 및 제어



(a) 축소 그래프



(b) 확장 그래프

그림 2. 조합 회로의 축소 그래프(a)와 확장 그래프(b).

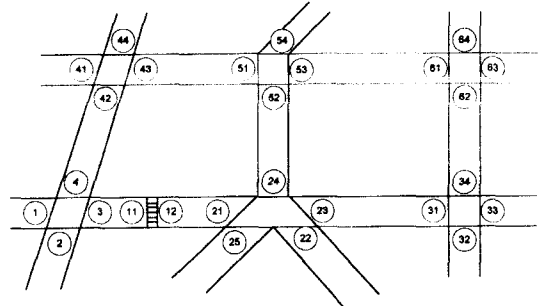


그림 3. 도로 교통 설계도.

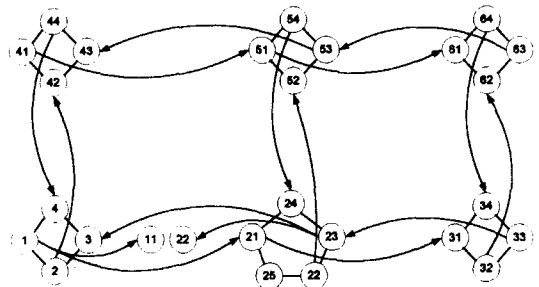


그림 4. 도로 교통 설계도의 축소 그래프.

앞 절에서 설명한 제약 조건을 이용하여 교차로를 부분적으로 모델링 한다. 모든 신호등은 동시 신호를 가진다고 가정하자. 그리고 신호제약은 모두 만족함을 기본으로 한다. 그러면 사거리의 4개 신호등은 충돌신호를 가진다. 또한 두 사거리 사이는 직진 신호등이 서로 조합되어야 한다. 그림 3과 같은 실제 교통 문제와 비슷한 복잡한 시스템을 모델링 다루었다. 축소 그래프는 그림 4와 같다.

4.2.1 각 제약조건에 의한 모델링

그림 4을 보면 각 교차로간의 신호등은 충돌 회로로 보이고 직진 방향의 신호등간은 조합 회로로 보인다. 그리고 신호등 11과 22는 충돌하지 않으며 신호등 3의 녹색사건이 발생한 후 일정시간이 지난 후에 신호 11과 12는 동시에 녹색사건이 발생하면 된다. 또한 신호등 21의 녹색사건 발생 후에도 동일하다. 그러므로 프로그램에서 제외하여 생각할 수 있다. 각 충돌 회로의 점등 순서는 신호등 제어에서 결정되어지며 각 회로에서 보이는 16개의 조합 조건은 다음과 같이 실시간 시간논리구조로 표현할 수 있다. 이 조합 조건들이 가지고 있는 시간들은 교통 이동시간에 관련되어 있다.

$$\square \left(e(1) \xrightarrow{t_{1,21}(T_u; T_m)} e(21) \xrightarrow{t_{21,31}(T_u; T_m)} e(31) \right) \\ \square \left(e(33) \xrightarrow{t_{33,23}(T_u; T_m)} e(23) \xrightarrow{t_{23,3}(T_u; T_m)} e(3) \right)$$

$$\begin{aligned}
 & \square \left(e(41) \xrightarrow{t_{41,51}(T_u; T_w)} e(51) \xrightarrow{t_{51,61}(T_u; T_w)} e(61) \right) \\
 & \square \left(e(63) \xrightarrow{t_{63,53}(T_u; T_w)} e(53) \xrightarrow{t_{53,43}(T_u; T_w)} e(43) \right) \\
 & \square \left(e(2) \xrightarrow{t_{2,42}(T_u; T_w)} e(42) \right) \tag{12} \\
 & \square \left(e(44) \xrightarrow{t_{44,4}(T_u; T_w)} e(4) \right) \\
 & \square \left(e(22) \xrightarrow{t_{22,52}(T_u; T_w)} e(52) \right) \\
 & \square \left(e(54) \xrightarrow{t_{54,24}(T_u; T_w)} e(24) \right) \\
 & \square \left(e(32) \xrightarrow{t_{32,62}(T_u; T_w)} e(62) \right) \\
 & \square \left(e(64) \xrightarrow{t_{64,34}(T_u; T_w)} e(34) \right) \\
 & \square \left(e(3) \xrightarrow{t_{3,11}(T_u; T_w)} e(11+r) \wedge e(12+r) \right) \\
 & \square \left(e(21) \xrightarrow{t_{21,12}(T_u; T_w)} e(11+r) \wedge e(12+r) \right)
 \end{aligned}$$

4.2.2 제어

그림 3와 4에 보여진 회로의 각 시간이 표 1과 같이 주어졌다고 하자. 표 1은 앞 절에서 표현한 모델링식에 맞추어 표현하였다.

여기에서 이동시간이란 현재 상태에서 다음 상태로 이동하는데 소요되는 시간을 말하며, 비움 시간이란 차량 집중에 의해 이미 존재하는 차량을 통과시키

표 1. coordination에 의한 이동시간과 비움 시간

Present event	Next event	Transition time	Evacuation time
e(1)	e(21)	4 : 40 s	10 s
e(21)	e(31)	3 : 50 s	8 s
e(33)	e(23)	3 : 50 s	7 s
e(23)	e(3)	4 : 40 s	7 s
e(2)	e(42)	4 : 05 s	3 s
e(44)	e(4)	4 : 05 s	7 s
e(41)	e(51)	3 : 40 s	6 s
e(51)	e(61)	4 : 05 s	2 s
e(63)	e(53)	4 : 04 s	6 s
e(53)	e(43)	3 : 40 s	5 s
e(22)	e(52)	3 : 30 s	5 s
e(54)	e(24)	3 : 30 s	7 s
e(32)	e(62)	3 : 50 s	3 s
e(64)	e(34)	3 : 50 s	4 s
e(3)	e(11+r)	1 : 50 s	0 s
e(21)	e(12+r)	2 : 50 s	0 s

는데 소요되는 시간이다. 조합 제약(coordination conflicts)에 의한 최적의 조합 시간(coordination time)은 이동시간에 대한 비움 시간의 차로써 식 (13)과 같이 얻을 수 있다.

$$T_{optimal} = \text{transition time} - \text{evacuation time} \tag{13}$$

충돌 제약을 만족하고 모든 조합 제약을 만족하게 하는 것이 최적의 신호등 조합이나 이를 만족하도록 하는 것은 쉽지 않은 일이다. 따라서 최적에 가깝도록 신호 시간을 배열하는 것이 이 절의 목적이다. 이로 인하여 식 (13)에서 얻은 $T_{optimal}$ 근처에서 오차에 의한 상한값과 하한값을 설정한다. 모의실험 결과는 표 2와 같다.

표 2. 각 신호등의 녹색 시간

충돌 신호 교차로 번호	01	02	03	04	05
00	0 s	30 s	90 s	60 s	
10	80+r s	80+r s			
20	29 s	101 s	53 s	5 s	77 s
30	5 s	95 s	65 s	35 s	
40	8 s	38 s	98 s	65 s	
50	99 s	69 s	9 s	39 s	
60	106 s	76 s	16 s	46 s	

표 3. 이상적인 조합 회로와 모의 실험에 의한 조합 회로의 비교

Present event	Next event	Ideal time	Simulation time
e(1)	e(21)	30 s	29 s
e(21)	e(31)	102 s	96 s
e(33)	e(23)	103 s	108 s
e(23)	e(3)	33 s	37 s
e(2)	e(42)	2 s	8 s
e(44)	e(4)	118 s	112 s
e(41)	e(51)	94 s	91 s
e(51)	e(61)	3 s	7 s
e(63)	e(53)	119 s	113 s
e(53)	e(43)	95 s	89 s
e(22)	e(52)	85 s	88 s
e(54)	e(24)	83 s	86 s
e(32)	e(62)	107 s	101 s
e(64)	e(34)	106 s	109 s
e(3)	e(11+r)	110 s	110 s
e(21)	e(12+r)	50 s	50 s

여기서 각 시간은 녹색사건의 발생 시간을 나타내며 $i+t$ 로 나타낸 시간은 적색시간을 나타낸다. 이 모의 실험에서 주기는 2분(120초)으로 하였고 오차 한계 시간은 6초로 하였다.

이상적인 조합 제약의 경우와 모의 실험에 의한 조합 제약의 차이를 표 3에 보였다. 그러나 이상적인 조합 제약은 중요한 충돌 제약을 만족하지 못하므로 바로 사용할 수 없다. 그러므로 충돌 제약을 만족하며 조합 회로를 이루는 모의 실험의 결과는 이상적인 경우와 최대 6초의 오차를 가지고 이루어 졌다.

5. 결 론

이산사건 시스템을 위한 형식화된 해석적 방법들의 개발은 프로세스 모델링과 제어를 위한 중요한 분야이나 이런 이론들이 충분히 연구되어지지 않았다.

본 논문은 이런 기법들 중에서 이산사건 시스템의 동작을 표현하기 위해 일반 논리에 시간의 전후관계를 표현할 수 있는 시간논리를 포함하는 시간논리구조를 소개하였다.

이산 사건 시스템을 실시간으로 제어하기 위해 기존의 시간논리구조에 실시간 이론을 추가한 실시간 시간논리구조를 제안하였고 이산 사건 시스템의 실시간 제어문제 중에서 고정시간 제어 문제에 적용하기 위하여 교통 신호등을 설계할 때 발생하는 문제 등을 실시간 시간논리구조로 모델링 하였다. 모의실험 결과 주어진 플랜트는 조합제약의 일정 오차 범위 안에서 만족할 만한 결과를 얻을 수 있었다. 이로써 실시간 시간논리구조가 이산사건 시스템을 모델링하고 제어를 설계하는데 유용하게 이용될 수 있음을 확인하였다.

참고문헌

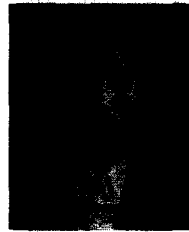
- [1] E. C. Yamalidou, E. P. Patsidou and J. C. Kantor, "Modeling Discrete- Event Dynamical Systems for Chemical Process Control-A Survey of Several New Techniques", *Computers Chem. Engng*, **14**(3), 281-299, 1990.
- [2] R. Sengupta and S. Lafortune, "Optimal control of a class of Discrete Event System", IFAC Symposium on Distributed Intelligence Systems, Arlington, Virginia, August 13-15, 25-30, 1991.
- [3] Dan Ionescu and Jing-Yue Lin, "Optimal Supervision of Discrete Event Systems in A Temporal Logic Framework", *IEEE Trans. on Systems. Man. and Cybernetics*, **25**(12), Dec., 1595-1605, 1995.
- [4] Benjamin C. Kuo, "Automatic Control Systems", Prentice-Hall, Inc., Englewood Cliffs, New Jersey, U.S.A, 1995.
- [5] A. Ganz, W. B. Gong, C. M. Krishna and W. Zhai, "Reconfiguration Algorithm for Ring Networks", *IEEE Proceedings of the 31st Conference on Decision and Control Tucson, Arizona, December, 3221-3226*, 1992.
- [6] George F. Luger and William A. Stubblefield, "Artificial Intelligence and the Design of Expert Systems", The Benjamin/Cummings Publishing Company, Inc., California, 1989.
- [7] P. J. Ramadge and W. M. Wonham, "Supervisory control of a class of discrete event process", *SIAM J. Control. Optimiz.*, **25**(1), 206-230, 1987.
- [8] MengChu Zhou, Kevin McDermott and Pareshe A. Patel, "Petri Net Synthesis and Analysis of a Flexible Manufacturing System Cell", *IEEE trans. on systems, Man. and Cybernetics*, **23**(2), March, 523-531, 1993.
- [9] E. C. Yamalidou and J. C. Kantor, "Modeling and Optimal Control of Discrete-Event Chemical Processes using Petri Nets", *Computers chem. Engng*, **15**(7), 503-519, 1991.
- [10] H. S. Hwang, S. C. Joo and D. Ionescu, "The Controller modeling using the temporal logic model in Discrete Event Dynamic Systems", *Journal of the KISS*, **21**(9), 1665-1674, 1994
- [11] J. F. Allen, "Toward a general theory of action and time", *Artificial Intelligence*, **23**, 123-154, 1984.
- [12] D. McDermott, "A temporal logic for reasoning about processes and plans", *Cognitive Science*, **6**, 101-155, 1982.
- [13] Paolo Serafini and Walter Ukovich, "A mathematical model for the fixed-time traffic control problem", *European Journal of Operational Research* **42**, 152-165, 1989.



황철수(Hyung Soo Hwang)정회원

1981년 : 전북대학교 전기공학과 졸업
1983년 : 동 대학원 전기공학과 졸업(석사)
1987년 : 동 대학원 전기공학과 졸업(박사)
1991년~1992년 : University of Ottawa,
Canada (Post-Doc.)
1988년~현재 : 원광대학교 제어계측공
학과 교수

주관심분야 : 퍼지 및 지능제어, 이산사건시스템제어, 대규모
시스템제어 등임



정용만(Yong Man Jeong)

1996년 : 원광대학교 제어계측공학과
졸업
1998년 : 동 대학원 졸업(석사)
1997년~1998년 : 한국 원자력 연구소 연
구생
1998년~현재 : (주)G.C.TECH 연구소
연구원

주관심분야 : 퍼지 및 지능제어, 이산사건시스템제어 등임
