'97 대한전기학회 창립 50주년 하계학술대회 논문집 (CIP25)

실시간 시간논리구조를 이용한 이산 사건 시스템의 모델링

정용만. 이원혁. 최정내. 황형수
원광대학교 제어계측공학과

Modeling of Discrete Event Systems with Real-time Temporal Logic Frameworks

Yong Man Jeong. Won Hyok Lee. Jeong Nae Choi. Hyung Soo Hwang
Department of Control and Instrumentation Engineering University of Wonkwang

Abstract - A Discrete Event Dynamic System 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. This paper proposes the use of Real-time Temporal Logic as a modeling tool for the analysis and control of DEDS. The Real-time Temporal Logic Frameworks is extended with a suitable structure of modeling hard real-time constraints. Modeling rules are developed for several specific situations. It is shown how the graphical model can be translated to a system of linear equations and constraints.

1. 서 론

많은 대규모의 동적 시스템들은 이산사건 시스템 구조로 이루어져 있다. 이러한 시스템들은 이산적인 수많은 상태와 이들 상태 사이의 천이관계를 나타냄으로써 표현할 수 있으며 생산 시스템, 교통 시스템, 일괄처리, 통신 시스템, 전문가 시스템 등을 예로 들 수 있다. 최근 이산사건 시스템에 대한 많은 연구가 이루어지고 있으며, 모델링 방법으로는 Automata and formal language, Petri-net[1], Finitely Recursive Process, Mini-Max Algebra, Temporal Logic Framework[2] 등이 제안되었다. 그러나 미분 방정식과 편미분 방정식 등에 의해 다루어 질 수 있는 선형 또는 비선형 연속 시스템들의 최적제어 이론, 안정도 이론, 가제어와 가관측 이론 등의 이론에 비해 이산사건 시스템을 위한 효과적인 이론은 아직까지 존재하지 않는다. 또한, 실시간 제어문제에 관련된 많은 연구가 없었기 때문에 최근에 이러한 문제를 해결하기 위한 많은 연구가 활발해지고 있다.

본 논문은 시간논리구조에 실시간 개념을 정의하고 시스템을 모델링 하는 것이다. 시간논리구조는 일반 논리구조에 시간 개념을 부가한 구조로 시간 관계를 논리로 표현한 이론이다. 그러나 이 논리는 실시간이라는 개념이 배제되어있다. 하지만 중요한 임계점에 놓여있는 시스템을 제어하고자 할 때 실시간 제어는 매우 중요한 문제이다.

2. 시간논리구조

시간논리구조는 시간개념을 포함된 술어논리(1차 논리)의 확장이며, 시간 순차열에 따른 추론 지향적인 논리이다. 논리를 이용하며 시제 표현식으로 표현될 수 있는 두 가지 장점을 가지는 시간논리는 Allen's Properties-Event-Process[3]와 Mcdermott's Fact-Event[4]의 시제추론을 이용하였다. 이 시간논리는 주로 software 증명에 이용되었으며 최근에는 이산사건 시스템의 제어문제에 적용되었다.[5]

2.1 시간논리구조의 기호 표현

시간논리구조는 시간과 시간에 대한 추론을 할 수 있는 형식구조이다. 시간논리구조는 전형적인 Boolean 연결자인 ¬(not), ∧(and), ∨(or), →(implies), ↔(if and only if)를 포함하며 시간의 변화량을 표현하기 위한 시간 연산자 □(henceforce), ◇(eventually), ○(next), U(until), P(proceed) 등을 가진다. 시간 연산자는 표 1에서 자세하게 보였다.

표 1 시간 연산자의 정의
Table 1 Definition of the temporal operators

$(\Box\varphi)$	henceforce φ is true.
$(\Diamond\varphi)$	there is a point in the future when φ will come true.
$(\bigcirc\varphi)$	φ will be true at the next time instant.
$(\varphi U \psi)$	φ is now true and there exists a point in the future when ψ will become true, φ will then cease to be true and φ will be true until then.
$(\varphi P \psi)$	φ must be true before ψ is true.

2.2 시간논리구조 모델

Lin과 Ionescu[2]에의한 동적식(dynamic formula) $l(e,s)$은 $\Box[\delta=e\wedge x=R\rightarrow(\bigcirc x)=Q]$의 형태를 가진다. 이것은 한 상태 s=x와 다음 상태 $s'=f(e,s)=\bigcirc x$에 의한 천이(transition)를 나타낸다. 이 식은 사건 e의 발생에 의해 원소 x가 상태 R에서 상태 Q로 변화함을 의미한다.

공정에서 시간논리구조 모델은 다음과 같이 6-변수를 가지는 식 M으로 정의된다.

$$M=(S,E,F^{*},f,s_{0},l) \qquad (1)$$

여기서, S는 상태 집합이고, E는 사건의 집합이다. 각 사건은 한 상태에서 다음 상태로의 천이와 같이 생각할 수 있다. 또한, s_0는 초기상태, f는 발화함수(firing function), F^{*}는 논리식의 집합이며 l은 E에서 F^{*}로의 labeling function이다. $\forall s\in S$와 $\forall e\in f(S)$에대해 S에서 E^{*}로의 사상으로 $e(s)=f(s,e)$가 정의된다. 여기서 $f(s)$는 상태 s에서의 사건발화 집합이다.

$e(s)=f(s,e)=s'$은 한 상태에서 다음 상태로 천이를 나타내므로 발화함수 f를 실시간 프로그래밍에서 동적 동작으로 주어지는 다음상태 함수라고도 한다.

3. 모델링 기법

3.1 플랜트와 제어기의 합성

플랜트와 제어기를 모델링하고 이들의 연관성을 이용하여 이들을 합성하는 문제는 매우 중요하다. 예를 들어 기차 건널목에서 플랜트인 기차와 제어기가 포함된 또 하나의 플랜트인 차단기는 분리되어 있지만 서로 연관성을 가지고 동작하게 된다.

앞 절에서 소개한 시간논리구조식에서 플랜트의 초기상태가 s_0이고 제어기의 초기상태를 c_0라 하면 플랜트와 제어기의 구조식은 각각 $M_p=(S_p,E_p,F^{*}_p,f_p,s_0,l_p)$와 $M_c=(S_c,E_c,F^{*}_c,f_c,c_0,l_c)$로 모델링 되어 질 수 있다. 우리는 플랜트의 사건과 제어기의 사건을 동기화 시킴으로써 구성한 폐루프 이산사건 시스템을 정의한다. 폐루프 시스템 M_{cls}는 다음과 같이 정의할 수 있다.

$$M_p\|M_c=(S_p\times S_c,E_p,F^{*}_p\cup F^{*}_c,f_{cls},(s_0,c_0),(l_p,l_c))$$

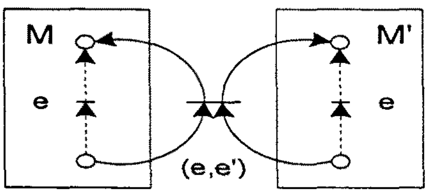

그림 1. 두 사건의 동기화

Fig. 1 synchronization of two events

일반적으로 폐루프 시스템 M_{cls}는 시스템의 요구동작을 지정하기 위해 주어졌다. 그림 1은 두 사건 M_p와 M_c의 동기화를 나타낸다.

3.2 실시간 시간논리구조

이산사건 시스템을 제어하고자 할 때 한 사건 e_i에 의해 한 상태에서 다음 상태로 상태가 천이 되는 동안 시간 t_i가 흐르게 된다. 이 천이시간 t_i동안 다른 사건 e_j가 발생할 수 있다. 이 사건 e_j에 의한 상태의 변화는 플랜트의 특성에 따라 받아들일 수도 있고 무시할 수도 있다. 또한 사건이 발생했을 때 상태의 천이가 어떤 시간(t_u)내에 종료되어야하는 경우가 있고 또한 어떤 시간(t_l)내에 종료되어서는 안되는 경우가 있다. 이러한 상황을 모델링 하기 위하여 전형적인 시간논리구조에 시간함수 t를 도입하여 식 (2)와 같은 실시간 시간논리구조식을 제안한다.

$$M_t=(S,E,F^{*},f,s_0,l,t) \qquad (2)$$

여기서 t는 시간함수이고 다음과 같이 사용한다.

- $t(T_l;T_u)$: 사건 발생 후 상태 천이의 하한시간 T_l과 상한시간 T_u을 표시한다. 하한시간만 또는 상한시간만 필요할 경우는 각각 $t(T_l;)$과 $t(;T_u)$으로 표기할 수 있다.
- $t(T)$: 상태의 천이시간이 정확하거나 그럴 필요가 있을 경우 사용한다.

이 시간함수는 차기 상태식과 ∧(and)로 연결될 수 있고, implies 기호 위에 표시할 수 있다. 또한 상태 천이 중에 다른 사건의 입력이 있을 경우 그 사건을 받아들이는 경우는 ($\underset{\uparrow e}{\longrightarrow}$)으로 표기하고 무시할 경우에는 일반적인 implies($\rightarrow$)만을 사용하여 구분하였다. 다음 식에 이러한 기호들의 사용 예를 보였다.

$$S_i(x)\wedge e_{i+1}(x)\underset{\uparrow e}{\longrightarrow}S_{i+1}(\bigcirc x)\wedge t(T_l;T_u)$$

$$S_i(x)\wedge e_{i+1}(x)\underset{\uparrow e}{\xrightarrow{t(T)}}S_{i+1}(\bigcirc x)$$

이러한 기호들을 이용하여 중요한 시간적 제약을 표현함으로써 이산사건 시스템을 실시간으로 모델링 할 수 있다. 이 기호들을 이용한 간단한 예를 다음 장에서 보였다.

4. 예 제

건널목의 기차와 차단기의 간단한 실례를 들어 보인다. 제어대상은 차단기이다. 그러나 차단기는 기차의 동작에 의존하므로 두 플랜트를 동기화 해야 한다. 먼저 기차의 상태에는 건널목에 접근상태(s_a), 건널목에 있는 상태(s_i), 건널목을 벗어나 달리는 상태(s_t)로 나타낼 수 있고, 차단기의 상태

에는 개방상태(s_u)와 차단상태(s_d)가 있다. 그리고 이 플랜트에서 발생할 수 있는 사건에는 접근사건(e_a), 건널목 안에 들어오는 사건(e_i), 건널목을 지나 달리는 사건(e_t)으로 표현 할 수 있다. 먼저 기차의 상태와 사건의 관계를 식 (3)과 같이 시간논리로 표현할 수 있다.

$$\begin{aligned}&\Box(s_t(x)\wedge e_a \rightarrow s_a(\bigcirc x))\\&\Box(s_a(x)\wedge e_i \rightarrow s_i(\bigcirc x))\\&\Box(s_i(x)\wedge e_t \rightarrow s_t(\bigcirc x))\\&\Box(s_t(x)\wedge e_t \rightarrow s_t(\bigcirc x))\end{aligned} \qquad (3)$$

이 식에서 상태의 천이시간은 존재하지 않으므로 시간을 표기하지 않는다. 그러나 차단기의 상태 천이에는 시간이 존재한다. 그러므로 천이 중에 다른 사건의 입력을 수락할 것인지를 함께 표기해야 한다. 차단기의 시간논리는 식 (4)와 같다.

$$\Box(s_u(x)\wedge e_a \xrightarrow{t(T)} s_d(\bigcirc x)) \qquad (4)$$

$$\Box(s_d(x)\wedge e_t \xrightarrow[\uparrow e]{t(T)} s_u(\bigcirc x)) \qquad (5)$$

식 (4)에서는 상태천이 중에 다른 사건을 허락하지 않지만 식(5)에서는 차단기가 올라가는 동안 또 다른 기차의 접근이 있으면 바로 내려와야 하므로 다른 사건의 입력을 받아들인다. 두 플랜트의 동기화를 그림 2에 보였다.

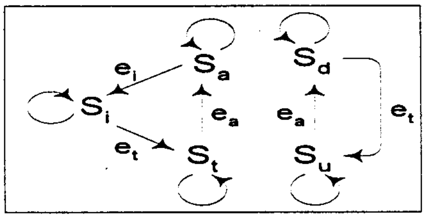

그림 2. 기차와 차단기의 동기화

Fig. 2 synchronization of train and gate

그림 2는 두 플랜트의 동기화이므로 시간을 표현하지 않았다. 이를 시간축 상에 놓고 사건의 발생과 상태의 변화를 그림 3에 보였다. 그림 3에서는 상태 천이시간과 상태천이동안 다른 사건의 허락여부를 쉽게 볼 수 있다.

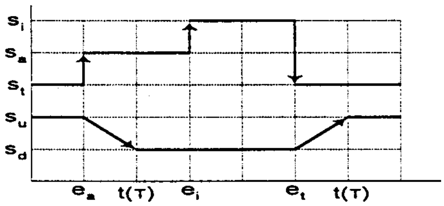

그림 3. 시간축 상의 상태 궤적

Fig. 3 trajectory of state of time axis

다음으로 이 플랜트의 안정성에 대한 조건이 있어야 한다. 예를 들어 상태가 s_i일 때 s_u의 상태를 가져서는 안된다는 것이다. 이러한 안정성에 관한 시간 논리는 식 (6)과 같다.

$$\begin{aligned}&\Box(s_i(x)\wedge\neg e_t \rightarrow \Box s_d(\bigcirc x))\\&\Box(s_t(x)\wedge\neg e_a \rightarrow \Box s_u(\bigcirc x))\\&\Box(\neg(s_i(x)\wedge s_u(x))\\&\Box(\neg(s_t(x)\wedge s_d(x))\end{aligned} \qquad (6)$$

이와 같이 이 시스템을 6 개의 동작조건과 4 개의 안정성을 가지는 실시간 시간논리로 표현하고 도식적으로 나타내었다. 비록 이 시스템은 간단한 모델이지만 보다 실시간 제어가 요구되는 대규모 시스템에도 이 이론을 적용 할 수 있을 것이다.

3. 결 론

이산사건 시스템을 위한 형식화된 해석적 도구들의 개발은 프로세스 모델링과 제어를 위해 중요하게 여기는 연구분야이다. 그러나 아직까지 이런 이론들이 충분히 연구되어지지 않았다.

이런 기법들 중에서 우리는 이산 사건 시스템을 시간논리구조로 표현하고 모델링 하는 방법을 소개하였으며 두 플랜트의 동기화에 대해 설명하였다. 그리고 시간논리구조를 이용하여 실시간으로 플랜트를 모델링 하는 방법을 제안하였고 제안된 이론을 이용하여 간단한 예를 들어 보였다.

앞으로 우리는 제안한 이론을 보다 대규모 시스템에 적용해보고 모델링 된 시스템을 실시간 시간논리 구조로 최적화 하기 위한 제어이론에 대하여 연구하고자 한다.

[참 고 문 헌]

[1] E. C. Yamalidou and J. C. Kantor "Modeling and Optimal Control of Discrete Event Chemical Processes using Petri Nets" Computer Chem. Vol. 15, No. 7, pp. 503-519, 1991

[2] D. Ionescu, J. Y. Lin, and H. S. Hwang, "Specification of Intelligent controllers for discrete event system in a temporal logic framework" IFAC workshop on Algorithms and architecture for real time control, Seoul, Korea Aug. 1992, pp. 237-242.

[3] J. F. Allen, "Toward a general theory of action and time" Artificial Intelligence, Vol. 23, pp. 123-154, 1984.

[4] D. McDermott, "A temporal logic for reasoning about processes and plants" Cognitive Science, Vol. 6, pp. 101-155, 1982.

[5] J. S. Ostroff "real time Computer Control of Discrete Event Systems Modelled by Extended State Machines : A Temporal Logic Approach, PhD. thesis, Dept. of Electrical Eng. University of Toronto, 1987.