

# DEVS 형식론에 기반한 하이브리드 시스템 모델링 시뮬레이션 방법론

임성용, 김탁곤

한국과학기술원, 전자전산학과, 전기 및 전자 공학 전공

시스템 모델링 시뮬레이션 연구실

Tel: 042-869-5454, E-mail: sylim@smslab.kaist.ac.kr, tkim@ee.kaist.ac.kr

하이브리드 시스템의 행동 표현을 각각의 시스템에 적합한 모델링 형식론으로 표현하면, 각각의 모델이 지니고 있는 정확도를 유지할 수 있고, 수행속도를 증가시키는 이점을 가지게 된다. 하이브리드 시스템을 모델링 하는 과정에서 시간 진행에 따른 수행 의미의 고찰은 모델링 방법에 중요한 영향을 끼칠 수 있다. 본 논문에서는 이산 사건 시스템과 연속 시간 시스템간의 입출력 교환을 위해서, 사건을 연속 입력으로 변환하고, 연속 출력을 사건으로 변환하는 인터페이스를 표현하는 형식론을 정의하고, 시뮬레이션의 시간 진행에 따른 종속 관계 방법과 독립 관계 방법을 모두 수행할 수 있는 알고리즘을 제안한다. 이 때, 이산 사건 시스템은 DEVS 형식론에 기반한 알고리즘으로 수행하고, 연속 시스템을 미분방정식의 산술적 누적 알고리즘에 따라 수행하는 통합 환경을 구축하며, 이를 바탕으로 두 미사일의 격추 예제를 수행하여 전쟁 시뮬레이션 분야에 적용가능성을 확인한다.

## 1. 서론

일반적으로 두 가지 이상의 시스템 의미가 혼재되어 있는 시스템을 하이브리드 시스템이라고 정의하고, 시간과 상태 변수의 연속성에 바탕을 둔 시스템의 분류에 의하여, 연속시스템 모델과 이산 사건 시스템 모델이 동시에 수행되는 형태의 시스템 또한 하이브리드 시스템이라고 볼 수 있다. 물리적인 연속 시스템을 이산적인 결정에 바탕을 둔 제어기를 통해 목적된 행동을 수행하기 위해서, 혹은 연속 시스템의 불연속적인 행동을 표현하기 위해서 하이브리드 시스템의 필요성은 존재한다.

계층적 제어분야에서 도로와 차량을 연속시스템으로 하고, 지능적 제어기를 Petri Net으로, 두 시스템 사이를 흐름 인터페이스를 통하여 모델링

하는 연구가 제안되었고[2], 하이브리드 시스템 모델의 검증을 위해서 오토마타의 상태변수 변화 조건에 연속 변수 조건을 포함한 하이브리드 오토마타의 연구도 진행되었다[3]. DEVS형식론을 바탕으로 하는 연구로는 기존의 DEVS형식론에 연속 상태 변수와 이에 따르는 상태 변화 함수, 그리고 연속 상태 변수의 변화에 의해 발생하는 상태 사건을 포함하는 멀티형식론이 제안되었고[1], 연속 시스템과 이산 사건 시스템과의 인터페이스를 상수 사건으로 정형화시킨 연구도 이루어 졌다[5]. 하지만 기존의 연구내용은 하이브리드 시스템의 표현 능력에 초점이 맞추어져 있기 때문에 제안되는 형식론을 바탕으로 하이브리드 시스템의 행동을 표현할 수는 있지만, 실제 수행 알고리즘에 따른 분류에 따라서 형식론이 연구되어지지 못했다.

본 논문의 2장에서는 하이브리드 시스템을 모델링 형식론과 수행 의미(Execution Semantic)를 중심으로 분류하며, 3장과 4장에서는 일반적인 하이브리드 시스템을 모델링하고 시뮬레이션하기 위해서 인터페이스 중심의 형식론과 그에 따르는 수행 알고리즘을 제안한다. 마지막으로 5장에서는 제안된 형식론과 수행 알고리즘의 필요성을 평가하고, 전쟁 시뮬레이션 분야에의 적용을 평가한다.

## 2. 하이브리드 시스템

### 2.1 하이브리드 시스템

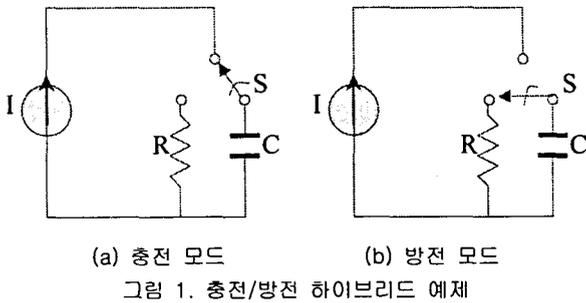


그림 1에서와 같이 전류전원 I와 저항R 그리고 축전기 C는 연속시스템으로 볼 수 있고, 각각 미분방정식의 형태로 표현할 수 있다. (a)에서는 I와 C의 연결로 구성되는 축전기의 충전상황을, (b)에서는 R과 C의 연결로 인하여 축전기에서 방전되는 상황을 미분방정식으로 표현할 수 있다. 하지만 스위치 S의 변화에 의해 발생하는 충전, 방전의 이산적인 변화는 미분방정식으로 표현할 수 없다. 예제에서와 같이 하이브리드 시스템은 이산 상태 변수와 연속 상태 변수를 동시에 포함하고 이산 입출력과 연속 입출력을 포함하고 있는 시스템이다.

### 2.2 하이브리드 시스템의 분류

하이브리드 시스템을 모델링 하는 과정에서

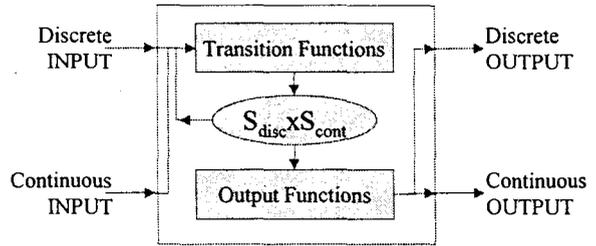


그림 2. (a) 확장형 모델

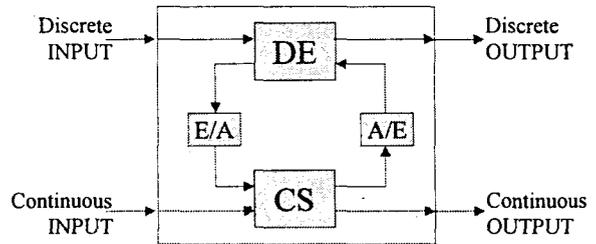
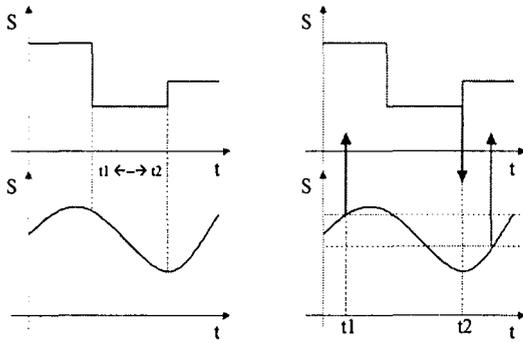


그림 2. (b) 인터페이스 적용 모델

고려해야 하는 점은 연속 상태 변수와 이산 상태 변수를 어떻게 배치할 것인가에 있다. 그림 2의 (a)에서와 같이 두 변수를 함께 사용하는 확장형 모델인 경우에는 연속 입력에 따르는 상태 변화 함수와 함께 이산 입력에 의한 상태 변화 함수를 함께 기술하여 주어야 한다[1]. 이 경우에는 포괄적인 하이브리드 시스템을 모델링 할 수 있다는 이점을 가지고 있지만, 이산 모델, 연속 모델 모두 기존의 모델을 사용할 수 없고 제안된 형식론에 따르는 새로운 모델을 기술하여야 한다는 단점을 가지고 있다. 반대로 (b)에서와 같이 기존의 모델을 그대로 사용하면서 하이브리드 시스템을 모델링 하기 위해서 이산 사건 시스템과 연속 시스템 사이의 입출력 교환을 위한 인터페이스의 서술이 필요하다[4].

이산 모델과 연속 모델을 분리한 인터페이스 적용 모델의 경우에 수행 의미에 관계된 두 상태 변수의 진행 방법에 따라서 그림 3에서와 같이 종속 관계와 독립 관계로 나누어 생각할 수 있다. (a)에서와 같이 이산 모델의 시간 진행에 따라서 연속 모델의 시간 진행이 종속되어 있는 경우에는



(a) 종속 관계 (b) 독립 관계

그림 3. 수행 의미에 따른 분류

연속 모델의 시간 진행은 항상 이산 모델에서 출력되는 시간 진행 사건에 의하여 시물레이션 시간을 지정하게 된다. 이산 모델은 시간 간격이 넓은 영역에서 정성적인 시물레이션을 진행하고, 이산 사건과 사건사이에는 연속 모델의 정량적인 시물레이션을 통하여 상태 변수가 진화하는 수행 의미를 지닌다. (b)에서는 이산 모델과 연속 모델은 시간 진행을 병렬적으로 수행하면서 연속 모델에서는 상태 변수의 영역에 따라서 사건을 발생시키고 이산 모델에서는 사건에 할당되어 있는 연속 입력을 발생시키게 된다. 수행 의미에 따라서 분류한 하이브리드 시스템을 모두 시물레이션할 수 있는 알고리즘을 제시하기 위해서 이론적 배경이 되는 형식론이 필요하게 된다.

### 3. 하이브리드 DEVS 형식론

#### 3.1 하이브리드 DEVS coupled 모델

기존의 DEVS coupled 모델과 비교하여 추가되어야 하는 요소로는 먼저 연속 입력과 연속 출력을 연속 모델과 연결시켜야 하고, 동시에 연속 모델들 사이의 연결을 추가하여야 한다. 또한 Atomic 모델들과 함께 연속 모델과 이산 모델 사이에는 입출력 교환을 위한 인터페이스 모델을 기

술해 주어야 한다. 그림 4와 같이 위의 요소들을 추가한 하이브리드 DEVS의 coupled 형식론은 아래와 같다.

$$HDEVS = \langle X, Y, M, EIC, EOC, IC, IM, Select \rangle$$

where

$$X = X_{disc} \cup X_{cont}$$

$$Y = Y_{disc} \cup Y_{cont}$$

$$M = \{DAM_d \mid DAM_d \in DAM\} \cup \{CAM_d \mid CAM_d \in CAM\}$$

$$EIC \subseteq \{(HDCM, x_{disc}, DAM_d, ip) \mid x_{disc} \in X_{disc}, DAM_d \in DAM\}$$

$$\cup \{(HDCM, x_{cont}, CAM_d, ip) \mid x_{cont} \in X_{cont}, CAM_d \in CAM\}$$

$$EOC \subseteq \{(DAM_d, op, HDCM, y_{disc}) \mid y_{disc} \in Y_{disc}, DAM_d \in DAM\}$$

$$\cup \{(CAM_d, op, HDCM, y_{cont}) \mid y_{cont} \in Y_{cont}, CAM_d \in CAM\}$$

$$IC \subseteq \{(DAM_d, op, DAM_d, ip) \mid DAM_d, DAM_d \in DAM\}$$

$$\cup \{(DAM_d, op, CAM_d, ip) \mid DAM_d \in DAM, CAM_d \in CAM\}$$

$$\cup \{(CAM_d, op, DAM_d, ip) \mid DAM_d \in DAM, CAM_d \in CAM\}$$

$$\cup \{(CAM_d, op, CAM_d, ip) \mid CAM_d, CAM_d \in CAM\}$$

$$IM \subseteq \{IM_{DAM_d, CAM_d} \mid DAM_d \in DAM, CAM_d \in CAM\}$$

$$\cup \{IM_{CAM_d, DAM_d} \mid DAM_d \in DAM, CAM_d \in CAM\}$$

$$Select: 2^M - \emptyset \rightarrow M$$

#### 3.2 이산 모델(DAM: Discrete Atomic Model)

DEVS 형식론은 이산 사건 모델을 계층적으로 모듈라하게 표현하기 위하여, 행동 표현의 가장 작은 단위인 Atomic 모델과 이들을 component로 가지면서 그들의 연결 명세를 기술하는 Coupled

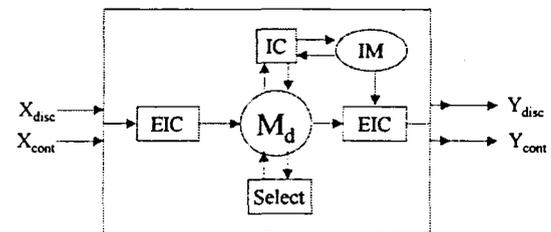


그림 4. 하이브리드 DEVS coupled 모델

모델로 구성된다. 하이브리드 시스템을 표현하기 위해서 하이브리드 DEVS 형식론은 이산 모델(DAM)과 연속 모델(CAM)을 Atomic 모델로써 포함하며, 이들의 연결을 하이브리드 Coupled 모델에서 서술할 수 있다. 개별적인 모델링 형식론은 깨뜨리지 않기 위해 Atomic 모델은 기존의 형식론의 표현력을 그대로 사용하는데, 이산 모델의 경우에도 DEVS 형식론에서 정의한 Atomic 모델의 표현력을 사용한다[1].

$$DAM = \langle X, Y, S, \delta_{ext}, \delta_{int}, \lambda, ta \rangle$$

### 3.3 연속 모델(CAM: Continuous Atomic Model)

물리적인 시스템을 수식적으로 표현하는 연속 시스템은 연속 입력 X, 연속 출력 Y, 연속 상태 변수 집합 S, 출력 함수 G, 그리고 상태 변수의 변화 함수 집합 F로 구성된다.

$$CAM = \langle X, Y, S, F, G \rangle$$

where

$$\dot{S} = F(S, X)$$

$$Y = G(S)$$

### 3.4 인터페이스 모델(IM: Interface Model)

이산 모델의 출력은 사건의 형태로 연속 모델의 입력과 연결되어 있으므로 이산 사건에 대응되는 연속 입력이 대응되어야만 한다. 반대로 연속 모델에 내재되어 있는 상태 변수의 영역 변화에 의하여 대응되는 이산 사건 출력이 발생하는 행동을 하는 인터페이스 모델을 기술해 주어야 한다 [6].

$$E/A: DAM.op \rightarrow W$$

$$A/E: CAM.S \rightarrow E$$

where

W: 입력 w(t)를 위해 정의된 연속 영역.

E: 상태 변수 영역 전이에 따른 사건 집합.

## 4. 시물레이션 알고리즘

기존의 DEVS 형식론을 시물레이션하기 위해서 Atomic 모델과 Coupled 모델에는 추상 시물레이터인 Simulator와 Coordinator의 알고리즘이 제안되었다. 이산 모델의 시물레이션 알고리즘은 기존의 Simulator를 사용하고, 연속 모델의 경우에는 인터페이스 모델의 정보를 포함한 상황에서 상태 변수의 진화에 따른 시간 진행을 결정하는 연속 Simulator를 생각할 수 있다. 입력으로 \*-메시지를 받는 경우에 시간 진행 값이 포함되지 않고, 사건이 발생하지 않는 경우, 시물레이션을 주어진 시간까지 진행하고 상태 변수의 변화를 상위 Coordinator에 돌려주게 되고, \*-메시지에 시간 진행 값이 포함되지 않는 경우에는 최고 시간 진행 값이 되기 전까지, 사건이 발생되면 사건발생 시간을 돌려주게 되고, 사건이 발생되지 않는 경우에는 널 사건을 돌려주게 된다(그림5).

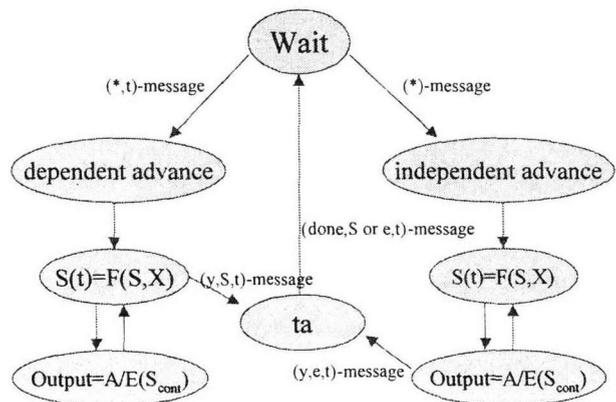


그림 5. CAM Simulator

Coordinator는 하위 모델에게 done-메시지를 받아서 최소의 시간 진행 값을 구해 낸다. 하이브

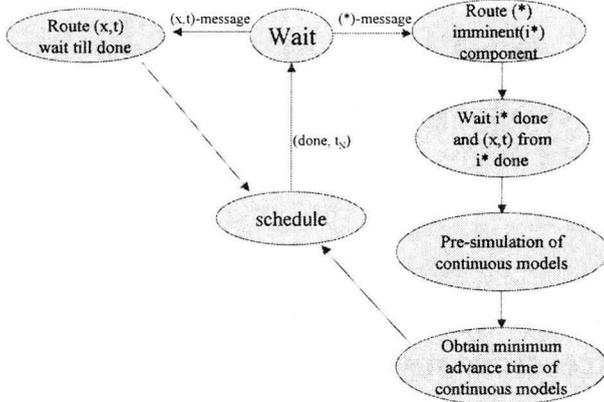


그림 6. HDEVS Coordinator

리드 DEVS coupled의 시물레이션 알고리즘에서 연속 모델은 먼저 시물레이션을 통해서 시간 진행을 시키고 그 결과에 따른 최소 시간 진행 값을 알아내고 이산 모델의 다음 사건 최소 시간 값과 비교하여 전체 모델의 최소 시간 진행 값을 결정하여 상위 모델에 전달하게 된다(그림6).

## 5. 결론 및 추후 과제

하이브리드 시스템을 모델링 하기 위하여 제안된 기존의 형식론들이 표현력의 확장과 검증에 초점이 맞추어져 있는 반면에 본 논문에서는 형식론이 실제로 시물레이션되는 과정에서 모델들의 시간 진행에 따른 수행 의미의 분류를 제안하였고, 수행 의미에 따른 모델링 방법 또한 달라짐을 확인하였다. 종속 관계에서는 시간 진행을 이산 모델에서 결정하고, 독립 관계에서는 이산 모델과 연속 모델에서 출력되는 사건들의 정렬로 시간 진행이 진행되는 알고리즘을 제안하였다.

본 논문에서 제안된 알고리즘을 바탕으로 수행 의미에 따라 분류되어진 예제의 실행결과의 검증과, 모델링 형식론과 시물레이션 알고리즘의 표현력 검증, 마지막으로 하이브리드 시스템과 제안된 형식론의 표현력에 대한 검증은 완결되지 않은

관계로 일단 추후 과제로 남겨 둔다..

## 참고 문헌

- [1] B. P. Zeigler, T. G. Kim and H. Praehofer, *Theory of Modeling and Simulation, 2nd ed.* Academic Press, New York, 2000.
- [2] J. Lygeros, D. Godbole and S. Sastry, "Simulation as a Tool for Hybrid System Design", *AIS '94*, Gainesville, Florida, December, 1994.
- [3] T. A. Henzinger, P. H. Ho, H. Wong-Toi, , "HYTECH: A Model Checker for Hybrid Systems", *Computer-Aided Verification '97*, 1997.
- [4] M. Lemmon, C. Bett, P. Szymanski and P. Antsaklis, "Constructing Hybrid Control Systems from robust Linear Control Agents", *Hybrid Systems II*, Springer, 1995.
- [5] N. Giambiasi, "Generalised Discrete Event Abstractions of Dynamic Systems", *AIS '2000*, Tucson, Arizona, March, 2000.
- [6] B. Lennartson, M. Tittus, B. Egardt and S. Pettersson, "Hybrid Systems in Process Control", *IEEE Control Systems*, October, 1996.