

MATLAB/Simulink를 이용한 DEVS 모델의 시물레이션 환경 구축

서경민^{1†} · 성장호¹ · 김탁곤¹

Simulation Environment of DEVS Models using MATLAB/Simulink

Kyung Min Seo · Chang Ho Sung · Tag Gon Kim

ABSTRACT

The DEVS (Discrete Event Systems Specification) formalism supports specification of discrete event models in a hierarchical modular manner. MATLAB/Simulink is widely used for modeling, simulating and analyzing continuous and discrete time systems. This paper proposes a realization of the DEVS formalism in MATLAB/Simulink. The proposed design enables to use a great amount of mathematical packages and functions included in MATLAB/Simulink. The design is also employed as real time simulation and hybrid system simulation which is a mixture of continuous systems and discrete event systems. The paper introduces Simulink-DEVS model, in which a simulation algorithm is embedded. The model consists of a Simulink-atomic model and a Simulink-coupled model. In addition, the time advance algorithm to simulate the model is suggested. The algorithm handles the time synchronization and the accommodation of different concepts specific to continuous and discrete event models. Two experimental results are presented for a pure discrete event model and a hybrid model.

Key words : DEVS formalism, MATLAB, Simulink, Discrete event simulation

요 약

DEVS 형식론은 이산 사건 시스템을 계층적이고 모듈러하게 표현할 수 있다. MATLAB/Simulink는 연속 시스템과 이산 시간 시스템을 모델링하고 시물레이션을 수행하는 데 널리 쓰인다. 본 논문은 MATLAB/Simulink 환경에서 DEVS 형식론을 구현하는 방법론을 제안한다. 본 논문에서 제안하는 방법론을 이용하여 MATLAB/Simulink에서 제공하는 다양한 공학 방정식과 알고리즘을 사용할 수 있다. 또한 동일한 시물레이션 환경에서 연속 시스템과 이산 사건 시스템이 혼합된 하이브리드 시스템에 대한 시물레이션을 수행할 수 있다. 이를 위해 본 논문은 Simulink-DEVS 모델과 이 모델을 시물레이션을 수행하는 데 필요한 시간 진행 알고리즘을 제안한다. 특히 시간 진행 알고리즘은 시스템의 유형에 상관없이 적절한 시간 진행을 수행한다. 두 가지 실험 결과를 통해 본 논문에서 제안하는 방법론의 효용성을 입증한다.

주요어 : DEVS 형식론, MATLAB/Simulink, 이산 사건 시물레이션

1. 서 론

DEVS 형식론은 Discrete Event Systems Specification으로 집합론에 근거한 형식론을 바탕으로 시스템을 계층적이고, 모듈러하게 표현할 수 있다. DEVS 형식론은 이

산 사건 시스템의 M&S(Modeling and Simulation)에 널리 사용되는데, 특히 시물레이션을 수행함에 있어 DEVS 모델의 성능 평가와 동작 검증을 하는데 주로 사용된다. 최근에는 통신 산업이나 제조업, 국방 산업 등에서 복잡한 내부 알고리즘을 사용하는 시스템에 DEVS 형식론을 적용하고 있다¹⁾. 이러한 이유로 DEVS 모델을 설계하고 설계된 모델을 분석하는 데에는 좀 더 복잡한 계산이나 알고리즘을 수행할 수 있는 시물레이션 도구가 필요하다. MATLAB/Simulink는 고성능 언어로써 수치 해석, 행렬 계산뿐만 아니라 미분 방정식, 알고리즘 개발, 과학 및 공학의 모델링, 신호 처리들을 다양한 그래픽 기능을 이용

2008년 11월 13일 접수, 2008년 12월 6일 채택

¹⁾ 한국과학기술원 전자전산학과 시스템 모델링 시물레이션 연구실

주 저 자 : 서경민

교신저자 : 서경민

E-mail: kmseo@smslab.kaist.ac.kr

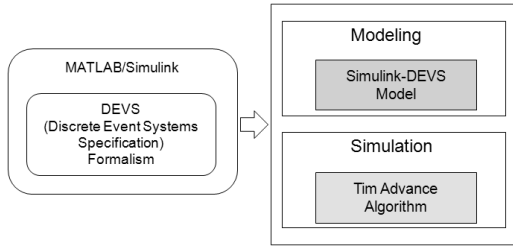


그림 1. 제안하는 프레임워크 개요

하여 제공하는 시뮬레이션 도구이다²⁾. 특히 MATLAB/Simulink는 연속 시스템이나 이산 시간 시스템을 모델링하고 시뮬레이션을 수행하는 M&S 기술자에게는 친숙한 시뮬레이션 도구이다.

따라서 본 논문은 MATLAB/Simulink를 이용하여 DEVS 형식론을 실현하는 방법론을 제안한다. 그림 1은 본 논문에서 제안하는 프레임워크 개요이다. 본 논문은 크게 두 가지를 제안한다. MATLAB/Simulink 환경에 적합한 DEVS 모델인 Simulink-DEVS 모델과 이 모델을 시뮬레이션을 하기 위해 필요한 시뮬레이션 시간 진행 알고리즘을 제안한다. 본 논문에서 제안하는 프레임워크를 이용하면 MATLAB과 Simulink에서 제공하는 다양한 공학 방정식과 알고리즘, 그리고 시각적인 도구를 이용할 수 있고, 동일한 시뮬레이션 환경에서 연속 시스템과 이산 사건 시스템이 혼합된 하이브리드 시스템⁴⁾을 모델링하고 시뮬레이션을 수행할 수 있는 장점이 있다.

[3]에서는 본 논문에서 제안하는 방법론과 유사하게 Simulink에서 제공하는 SimEvents⁷⁾라는 툴박스(toolbox)를 이용하여 이산 사건 시스템 및 하이브리드 시스템의 모델링 및 시뮬레이션을 제안하였다. 그러나 SimEvents는 큐(Queue)와 서버(Server)를 이용한 이산 사건 시스템에 국한하여 사용할 수밖에 없고, DEVS 형식론과 같은 이산 사건 시스템에 관한 수학적 형식론을 적용할 수 없기 때문에, 수학적으로 시스템을 모델링하는 것이 불가능하다.

본 논문의 구성은 다음과 같다. 2장과 3장에서 DEVS 형식론과 MATALB/Simulink에 대한 간단한 소개를 한 다음, 본 논문의 목적인 Simulink-DEVS 모델과 시뮬레이션 시간 진행 알고리즘을 4장과 5장에서 다룬다. 6장에서 두 가지 사례 연구를 다루고 7장에서 결론을 맺는다.

2. DEVS 형식론

DEVS 형식론은 이산 사건 시스템을 객체 단위로 모들

화 하여 계층적으로 결합하여 표현하는 집합론에 근거한 수학적 틀이다. DEVS 형식론에는 시스템 기본적인 구성 요소를 나타내는 원자 모델과 여러 모델을 합쳐서 새로운 모델을 구성할 수 있는 결합 모델이 있다⁵⁾.

2.1 원자 모델 표현

원자 모델(Atomic Model)은 DEVS 형식론을 구성하는 가장 기본적인 모듈로서 시스템의 행동을 기술하는 모델이다. 원자 모델 M의 수학적 표현은 다음과 같다.

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

X : 이산사건 입력 집합
 Y : 이산사건 출력 집합
 S : 일련의 이산사건 상태의 집합
 $\delta_{ext} : Q \times X \rightarrow S$: 외부 상태 천이 함수
 $Q = \{(s,e) | s \in S, 0 \leq e \leq ta(s)\}$: total state of M
 $\delta_{int} : Q \rightarrow Q$: 내부 상태 천이 함수
 $\lambda : Q \rightarrow Y$: 외부 상태 천이 함수
 $ta : S \rightarrow R_{0,\infty}^+$: 시간 진행 함수

2.2 결합 모델

결합 모델(Coupled Model)은 여러 모델을 내부적으로 연결하여 만든 모델이다. 내부 구성요소가 되는 모델은 원자 모델과 결합 모델이 모두 가능한데, 이러한 내부 모델들을 계속 합쳐서 더욱 큰 시스템을 표현할 수 있다. 다음은 결합 모델의 수학적 명세이다.

$$CM = \langle X, Y, \{M_i\}, EIC, EOC, IC, SELECT \rangle$$

X : 이산사건 입력 집합
 Y : 이산사건 출력 집합
 $\{M_i\}$: 모든 이산사건 컴퍼넌트 모델들의 집합
 EIC : 외부 입력 연결 관계
 EOC : 외부 출력 연결 관계
 IC : 내부 연결 관계
 $SELECT : 2^{\{M_i\}} - \emptyset \rightarrow M_i$: 같은 시각에 존재하는 사건을 발생하는 모델들에 대한 선택 함수

2.3 DEVS 추상화 시뮬레이터

DEVS 형식론에서 모델 자체는 수동적인 개체이다. 모델은 미리 정해진 집합과 함수들로 이루어져 있을 뿐, 실제로 정해져 있는 함수를 호출하는 것은 모델에 연결되어 있는 시뮬레이션 프로세스이다. 원자 모델에는 시뮬레이터(Simulator)라고 하는 시뮬레이션 프로세스가, 결합 모

표 1. DEVS 시뮬레이션 메시지

Message	Meaning
$(* , t)$	Internally generated event at time t that notifies the scheduled time is completely elapsed
(x, t)	External input event at time t
(y, t)	Internal output event at time t
$(done, t_N)$	Synchronization event generated at time t_N that notifies the next scheduled time is t_N

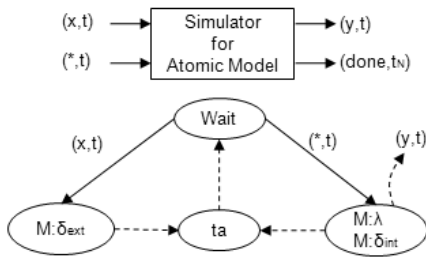


그림 2. 시뮬레이터 알고리즘

텔에는 코디네이터(Coordinator)라는 시뮬레이션 프로세스가 위치한다^[6].

표 1은 DEVS 시뮬레이션 메시지를 나타내고, 그림 2와 그림 3은 각각 시뮬레이터와 코디네이터의 알고리즘을 나타낸다.

3. MATLAB/SIMULINK

3.1 MATLAB

MATLAB은 행렬을 기본적인 자료의 단위로 처리하는 프로그램으로 초기에는 행렬 연산, 선형 대수 및 수치 해석에서 쉽게 사용할 수 있도록 만들었다. 현재의 MATLAB은 고성능 언어로서 수치 해석, 선형 대수, 행렬 계산뿐만 아니라 미적분학, 미분 방정식, 알고리즘 개발, 과학 및 공학의 모델링, 신호 처리, 제어 계측, 인공 지능의 활용 및 다양한 그래픽 기능 등을 제공하는 패키지이다^[2].

3.2 SIMULINK

Simulink는 동적인 시스템을 모델링 및 시뮬레이션을 수행하기 위한 도구로 강력한 GUI(Graphic User Interface) 기능을 제공한다^[7,8]. MATLAB과는 달리 Simulink는 일정한 시간 간격을 두고 모델을 구성하는 모든 블록들을 실행하여 하나의 출력을 얻어낸다. 여기서 언급하는 시간

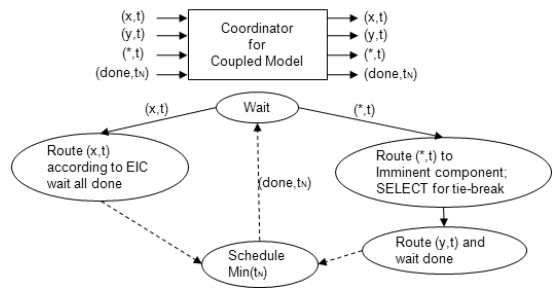


그림 3. 코디네이터 알고리즘

간격은 Simulink 매뉴얼에서 step time이라고 언급하고 있다. 본 논문에서는 step time을 이산 시간 시뮬레이션에서 알기 쉬운 개념인 Simulink 클럭으로 대체하여 설명한다.

3.2.1 S-함수

S-함수(System-function)는 Simulink의 기능을 확장하기 위한 기능이다. S-함수는 임의의 Simulink 블록을 MATLAB, C/C++, Ada 또는 Fortran과 같은 컴퓨터 언어로 표현할 수 있다. 본 논문에서는 MATLAB 언어를 이용하여 M 파일로 구현한 S-함수를 사용한다. M 파일로 구현한 S-함수를 사용하는 이유는 별도의 컴파일 과정이 필요 없고, MATABL 언어를 아는 사용자라면 누구나 손쉽게 간편하게 구현이 가능하기 때문이다.

4. SIMULINK-DEVS 모델

Simulink-DEVS 모델은 Simulink-원자 모델과 Simulink-결합 모델로 구성된다. Simulink-원자 모델에는 DEVS 원자 모델 시뮬레이션 프로세스인 시뮬레이터에 대응하는 시뮬레이션 알고리즘이 포함되어 있다.

4.1 Simulink-원자 모델

그림 4는 DEVS 원자 모델과 Simulink-원자 모델을 나타낸다. Simulink-원자 모델은 3개의 집합과 4개의 함수 블록, 그리고 초기화(Initialization), 메시지 분류(Message Classifier), 이벤트 시간 스케줄러(Event Time Scheduler) 블록에 대한 템플릿을 제공한다. DEVS 원자 모델과 비교하여 볼 때, Simulink-원자 모델에는 메시지 분류 블록, 이벤트 시간 스케줄러 블록 그리고 초기화 블록이 추가적으로 존재한다. 세 블록에 대한 설명은 4.3장에서 하도록 한다. Simulink-원자 모델의 4개의 함수 블록과 시뮬레이션 알고리즘 블록, 초기화 블록은 모두 S-함수를 이용하

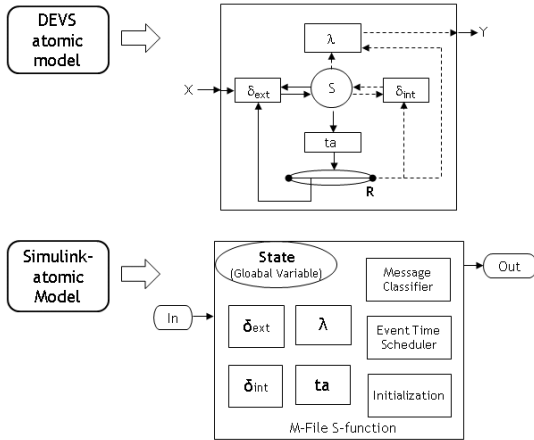


그림 4. DEVS 원자 모델과 Simulink-원자 모델

여 구현을 하였는데, 특히 S-함수의 여러 가지 콜백(call back) 함수 중에 출력 함수를 이용하였다. S-함수에 사용하는 시뮬레이션 언어로는 MATLAB 언어를 사용한 M-파일 S-함수를 사용하였다.

사용자가 Simulink-원자 모델을 사용하기 위해서는 Simulink-원자 모델에서 제공하는 템플릿에 따라 3개의 집합 블록과 4개의 함수 블록을 완성하면 된다. 입력 이벤트 집합과 출력 이벤트 집합은 Simulink의 입력 포트 블록과 출력 포트 블록을 사용하여 표현하였다. 상태 집합은 4개의 함수 블록 내부에 전역 변수로 선언되어 있기 때문에 Simulink-원자 모델의 함수 블록들은 블록들 간의 입출력 관계가 없이도 상태 변수에 대한 접근이 언제든지 가능하다. Simulink는 기본적으로 이산 시간 시뮬레이션 기반이기 때문에 DEVS 형식론을 그대로 Simulink 상에서 표현하는 데는 제약 사항이 따른다. 따라서 본 논문에서는 모델의 어떤 상태에 머무를 수 있는 time advance 값이 반드시 양수라고 가정한다.

4.2 Simulink-결합 모델

좀 더 복잡한 Simulink 모델을 만들다 보면, 하나의 Simulink 모델 윈도우에 모든 블록들을 배치하기에는 공간이 부족한 경우가 발생한다. Simulink 모델에서 일정한 기능을 수행하는 부분을 따로 하나의 서브시스템 블록으로 만들어 놓고, 상위 Simulink 모델과 계층적인 링크 관계를 유지하는 형태로 할 수 있다면 보다 논리적이고, 간결한 모델링이 될 수 있다. 이를 위해 Simulink에서는 서브시스템(SubSystem)이라는 기능을 추가적으로 제공하고 있는데, 이는 DEVS 형식론의 결합 모델과 의미가 동

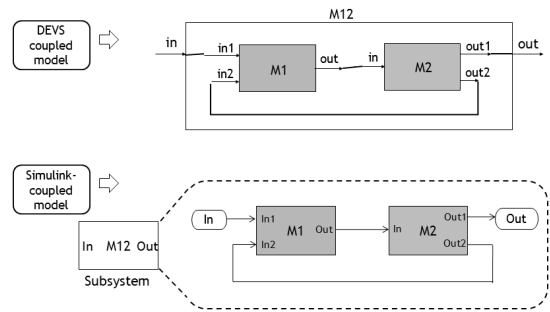


그림 5. DEVS 결합 모델과 Simulink-결합 모델

일하다. 따라서 Simulink-결합 모델의 구현은 결합 모델 내부의 컴퍼넌트들을 선으로 연결하고 서브시스템으로 묶음으로써 표현할 수 있다.

그림 5는 Simulink-결합 모델과 DEVS 결합 모델 사이의 상관관계를 나타낸다. 서브시스템으로 구현되어 있는 Simulink-결합 모델을 더블 클릭하게 윈도우에 Simulink-결합 모델 내부가 나타나게 되는데 그 내부 구조가 DEVS 형식론의 결합 모델과 형태와 일치한다. Simulink-원자 모델과 마찬가지로 Simulink-결합 모델에서도 Simulink의 특성으로 인한 제약 사항이 따른다. 같은 시간에 두 블록이 실행이 되어야 하는 경우 두 블록 간의 연결 관계에 의해 실행 순서가 결정이 된다. 따라서 본 논문에서는 DEVS 결합 모델의 선택 함수(Select Function)는 사용하지 않도록 한다.

4.3 Simulink-원자 모델 시뮬레이션 알고리즘

Simulink-원자 모델은 시뮬레이션 알고리즘이 포함되어 있다. 본 장에서는 Simulink-원자 모델의 시뮬레이션 알고리즘에 대해 설명한다.

그림 6은 Simulink-원자 모델의 구성을 나타낸다. 메시지 분류(Message Classifier) 블록과 이벤트 시간 스케줄러(Event Time Scheduler) 블록은 DEVS 추상화 시뮬레이터에서 시뮬레이터 알고리즘의 역할을 수행하고, 초기화(Initialization) 블록은 Simulink-원자 모델의 초기 상태와 첫 번째 이벤트 시간을 결정하는 역할을 수행한다. 앞서 설명하였듯이 사용자가 Simulink-원자 모델을 사용하기 위해서는 메시지 분류 블록과 이벤트 시간 스케줄러 블록에 대해서는 신경 쓸 필요 없이 Simulink-원자 모델에서 제공하는 템플릿에 따라 3개의 집합과 4개의 함수 블록을 완성하면 된다. Simulink-원자 모델에서 초기화 블록과 출력 함수 블록 사이에 MUX 블록을 이용함으로써 메시지 흐름이 두 가지 경우로 분기한다. 분기된 신호

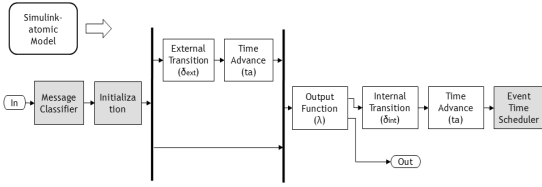


그림 6. Simulink-원자 모델 구성도

```

Simulink-atomic Message Classifier:
Inputs:
  Simulink clock
  message
Output:
  m
Variables:
   $t_N$  → next event time
  t → current simulation time

For each Simulink clock
  when receive x-message (x,t) at time t
    m = 1
  when receive *-message (*,t) at time t
    if t =  $t_N$ , then
      m = 2
    else
      m = 0
  when receive x-message (i,t) at time t
    m = 3
  send m to Initialization block

End Simulink-atomic simulator

m = 0 → No transitions are occurred
m = 1 → External transition occurred
m = 2 → Internal transition occurred
m = 3 → Get initialization
    
```

그림 7. Simulink-원자 모델의 메시지 분류 알고리즘

는 ADD 블록을 이용하여 하나의 신호로 합치게 된다. 먼저, Simulink-원자 모델이 입력 이벤트를 받아서 외부 상태 천이가 발생하였을 경우, MUX를 통한 메시지는 상단으로 전달되어 외부 상태 함수와 시간 진행 블록이 수행이 된다. ADD 블록을 통해 전달된 메시지는 출력 함수, 내부 천이 함수, 시간 진행 함수 블록을 바이패스하고, 이벤트시간 스케줄러 블록을 수행하게 된다. 다음으로 들어오는 입력 이벤트가 없이 Simulink-원자 모델에서 내부 상태 천이가 발생하였을 경우에는 MUX 블록을 지난 메시지가 하단을 통해 전달이 되어 출력 함수, 내부 천이 함수, 시간 진행 함수, 이벤트시간 스케줄러 블록이 수행이 된다.

그림 7과 8을 각각 Simulink-원자 모델의 메시지 분류 알고리즘과 이벤트시간 스케줄러 알고리즘을 나타낸다. 메시지 분류 알고리즘에 표현된 (i,t) 메시지는 Simulink-원자 모델의 초기 상태와 초기 상태에 대한 첫 번째 이벤

```

Simulink-atomic Event Time Scheduler:
Inputs:
  m
Variables:
  t → current simulation time
   $t_N$  → next event time

For each Simulink clock
  if m = 1 or m = 2
     $t_N$  = t + time advance value(ta(s))

End Simulink-atomic Event Time Scheduler
    
```

그림 8. Simulink-원자 모델의 이벤트 시간 스케줄러 알고리즘

트 시간을 얻기 위한 메시지이다. 메시지 분류 알고리즘은 각 Simulink 클럭마다 들어오는 메시지의 유형에 따라 출력 값을 다르게 적용하여 보낸다. 이렇게 결정된 출력 메시지의 값을 이용하여 이벤트 시간 스케줄러 블록에서는 전역 변수로 지정된 현재 상태의 time advance 값을 이용하여 다음 이벤트 시간 t_N 이 결정한다.

5. 시뮬레이션 시간 진행 알고리즘

Simulink-DEVS 모델을 시뮬레이션하기 위해서는 Simulink 클럭과 시뮬레이션 시간 사이에 적절한 매핑이 필요하다. Simulink 클럭은 Simulink의 step time을 의미하는데, Simulink는 이 클럭마다 Simulink내에 존재하는 모든 블록들을 실행하여 하나의 결과를 얻고, 클럭을 업데이트한다. 여기서 Simulink 클럭은 시간의 개념이라기 보다 Simulink 블록들을 모두 수행하는데 필요한 tick이라고 보는 것이 합당하다. 앞서 설명한 일련의 과정은 시뮬레이션이 종료되는 순간까지 반복 수행이 된다. 따라서 Simulink-DEVS 모델을 시뮬레이션을 수행하기 위해서는 Simulink 클럭을 의미 있는 시뮬레이션 시간으로 적절히 변환해 주는 기능이 필요하고, 이러한 역할을 담당하는 블록이 Simulation Time Generator이다. Simulation Time Generator를 설계하는 데 있어 가장 중요한 점은 시스템의 유형이 무엇이든지 상관없이 시간 진행을 수행하여야 한다는 것이다. 따라서 본 논문에서는 시스템의 유형이 연속 시스템이거나 이산 사건 시스템, 혹은 연속 시스템과 이산 사건 시스템이 혼합된 하이브리드 시스템 이거나에 관계없이 Simulation Time Generator가 시스템의 유형에 맞게 시간 진행을 수행할 수 있도록 설계하였다.

그림 8은 Simulation Time Generator의 역할을 나타낸다. Simulation Time Generator는 각 Simulink 클럭마다 모든 Simulink-DEVS 모델들의 다음 이벤트 시간을

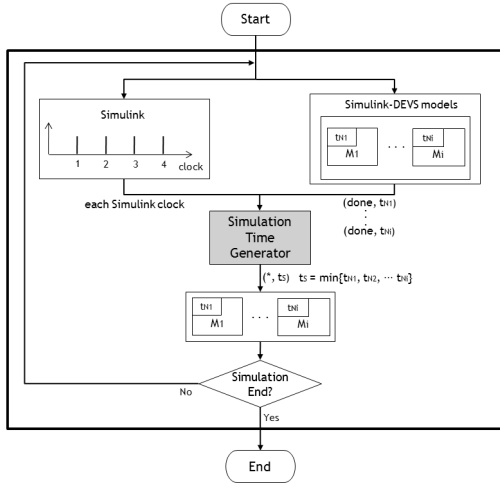


그림 8. Simulation Time Generator

(done, t_N) 메시지를 통해 전달받는다. 이벤트 시간의 경우 Simulink 상에서 전역 변수로 설정되어 있기 때문에 별도의 인터페이스 없이 Simulation Time Generator는 모든 모델들의 다음 이벤트 시간을 알 수 있다. Simulation Time Generator는 모델들로부터 수신한 다음 이벤트 시간을 바탕으로 최소의 이벤트 시간을 시뮬레이션 시간 t_s 로 결정하여 모든 모델에 $(*, t_s)$ 메시지를 전달한다. 여기서 중요한 점은 Simulation Time Generator에서 보내는 $(*, t_s)$ 메시지는 해당 모델에만 보내는 것이 아니라 모든 모델에 $(*, t_s)$ 메시지를 보낸다는 점이다. 따라서 모든 Simulink-원자 모델들은 $(*, t_N)$ 메시지를 수신하고 내부 시뮬레이션 알고리즘을 수행하게 된다.

그림 9는 시간 진행 알고리즘을 나타낸다. 시간 진행 알고리즘은 시스템의 유형에 관계없이 $(*, t_s)$ 메시지를 전달한다. 이산 사건 시뮬레이션의 경우에는 모든 다음 이벤트 시간 중 가장 최소의 이벤트 시간을 t_s 로 설정하여 $(*, t_s)$ 메시지를 전달하고, 이산 시간 시뮬레이션의 경우에는 모델의 다음 이벤트 시간 자체가 다음 Simulink 클럭이 되므로 다음 Simulink 클럭 값을 t_s 로 설정하여 $(*, t_s)$ 메시지를 전달한다.

6. 실험 결과 및 분석

앞서 제안한 Simulink-DEVS 모델에 대한 효율성을 입증하기 위해 두 가지 사례 연구를 수행하였다. 첫 번째 사례 연구는 시뮬레이션 목적 중 하나인 성능 평가를 제대로 수행하는 지 여부를 판단하는 실험이고, 두 번째 사

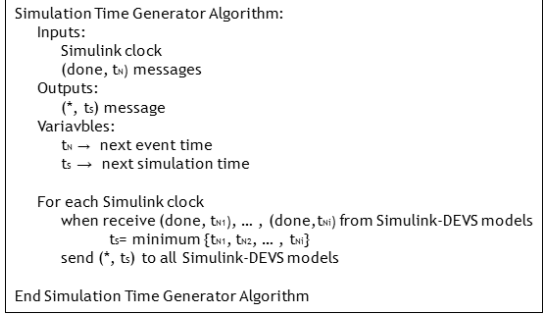


그림 9. Simulation Time Generator의 시간 진행 알고리즘

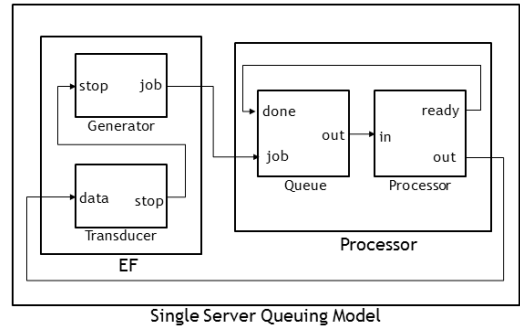


그림 10. 단일 서버 큐잉 DEVS 모델 구성도

례 연구는 Simulink-DEVS 모델의 동작 검증과 더불어 연속 시스템과 이산 사건 시스템이 혼합된 하이브리드 모델에 대한 실험이다.

6.1 단일 서버 큐잉 모델

단일 서버 큐잉 모델(Single Server Queuing Model)은 하나의 신호 발생기(Generator), 큐(Queue), 프로세서(Processor), 트랜스듀서(Transducer) 이렇게 총 4개의 하위 모델로 구성되어 있다. 동일한 단일 서버 큐잉 모델에 대해 그림 10은 DEVS 모델 구성도를 나타내고, 그림 11은 Simulink-DEVS 모델 구성도를 나타낸다.

그림 12는 동일한 단일 서버 큐잉 모델에 대해 본 논문에서 제안하는 시뮬레이션 도구와 기존에 연구되고, 개발되었던 DEVSim++의 결과를 비교한 것이다. 여기서 DEVSim++는 DEVS 형식론을 시뮬레이션하기 위한 기존의 시뮬레이션 도구로 C++ 언어 기반으로 개발되었다^[10]. 본 논문에서는 나타내지 못했지만 동일한 단일 서버 큐잉 모델에 관해 DEVSim++와 본 논문에서 제안한 시뮬레이션 도구의 이벤트 발생순서가 동일하였다. 그림 12에서는 제안한 방법론이 성능 평가를 제대로 나타내는 지에 대한 그래프이다. 나타내고자 하는 성능 평가 척도는 서버의

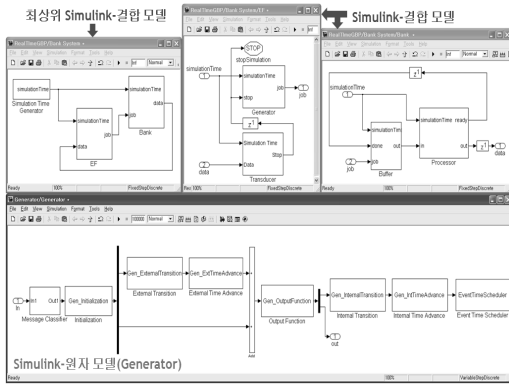


그림 11. 단일 서버 큐잉 Simulink-DEVS 모델 구성도

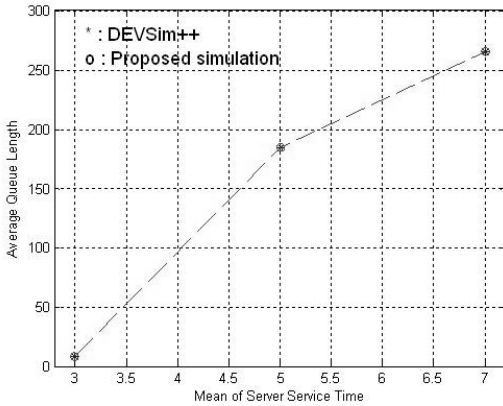


그림 12. 시뮬레이션 결과

서비스 시간에 따른 평균 큐의 길이이다. 서버의 서비스 시간은 entity 하나당 처리되는 시간을 의미한다. 본 실험에서는 서버의 서비스 시간이 분산이 1인 정규 분포를 따른다고 가정한다. 그림 12로부터 두 시뮬레이션의 결과가 완벽히 일치하는 것을 알 수 있다.

6.2 간략화된 위게임 모델

간략화된 위게임 모델(War Game Model)은 하나의 아함에 하나의 표적이 접근하면서 아함이 표적을 격추시키는 모델이다. 그림 13은 위게임 모델 구성을 나타낸다. 전투체계는 크게 기동-탐지-교전-손상평가 이렇게 4단계로 이루어진다. 탐지를 제외한 기동, 교전, 손상평가는 Simulink-DEVS 모델을 이용한 이산 사건 시스템으로 설계하였고, 탐지 모델인 레이더 모델은 연속 시스템으로 표현하였다. 아함 및 표적은 독립적으로 기동을 하게 되며 매 주기마다 레이더 모델에 위치 정보를 전송한다. 레

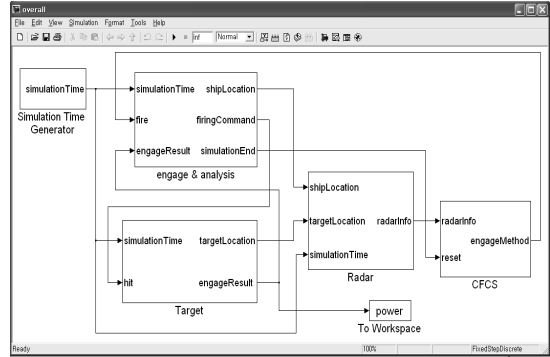


그림 13. 위게임 모델 구성도

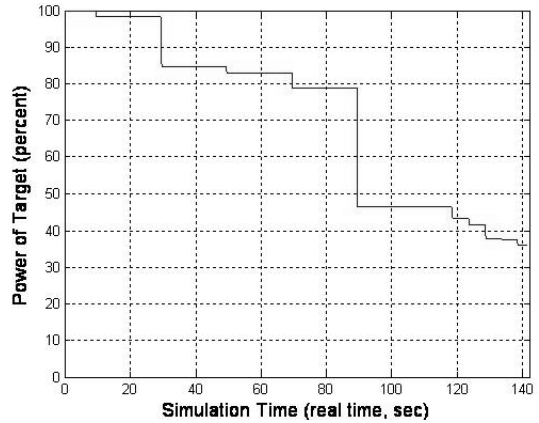


그림 14. 시뮬레이션 결과

이더 모델은 아함 정보와 표적 정보를 통해 탐지 가능 여부를 판단하여 탐지 정보를 CFCS 모델에 전송한다. CFCS 모델은 탐지 정보를 받아 위협 평가 및 무기 할당 과정을 거쳐 할당된 무기를 교전분석 모델에 전달한다. 교전이 시행되고 난 후에는 손상 평가를 하게 된다.

그림 14는 위게임 모델의 시뮬레이션 결과를 나타낸다. 시뮬레이션 결과는 시뮬레이션 시간에 따른 표적의 power를 나타낸다. 위게임 모델은 하이브리드 모델이기 때문에 본 사례 연구에서 시간 진행 알고리즘은 이산 시간 시뮬레이션으로 동작하게 된다. 시뮬레이션 시간이 증가하면서 표적이 아함에 접근함에 따라 아함은 표적을 공격하게 되고, 표적의 power가 줄어든다. 본 사례 연구를 통해 Simulink-DEVS 모델을 이용하여 시뮬레이션 동작 검증과 더불어 하이브리드 모델에 대한 시뮬레이션이 가능하다는 것을 알 수 있다.

7. 결 론

본 논문은 DEVS 형식을 Simulink를 이용하여 구현하는 방안을 제시하였다. DEVS 결합 모델을 표현하기 위해 Simulink의 서브시스템 기능을 사용하여 동일한 형태의 Simulink-결합 모델을 만들고, DEVS 원자 모델의 3개의 집합과 4개의 함수를 표현하기 위하여 M-파일 S-함수를 이용한 사용자 정의 블록을 제안하였다. Simulink 클럭 단위에서 Simulink-DEVS 모델을 시뮬레이션하기 위하여 시뮬레이션 메시지를 정의하고 새로운 시간 진행 알고리즘을 제시하여 기존의 DEVSim++ 환경에서 개발된 모델과 동일한 기능을 제공하는 Simulink-DEVS 모델 개발 방법론을 제시하였다. 이러한 Simulink-DEVS 모델은 재사용적인 측면에서 장점을 가지고 시뮬레이션을 수행시 성능 평가와 동작 검증을 만족한다.

본 논문에서 제시한 방법론은 Simulink의 다양한 수학적 도구와 알고리즘을 이용하여 복잡한 이산 사건 시스템을 쉽게 표현할 뿐만 아니라 시스템을 모델링하고 시뮬레이션을 하기 위한 용도로 사용될 수 있을 것으로 생각된다. 추후 과제로 사용자가 Simulink를 사용하여 DEVS 모델을 보다 쉽게 개발하기 위하여 모델을 라이브러리화 하는 연구가 필요하다.

참 고 문 헌

1. Ki Jung Hong, Tag Gone Kim, "DEVSPEC: DEVS specification language for modeling, simulation and analysis of discrete event system," *Transaction on Information and Software Technology*, Vol. 48, No. 4, pp. 221-234, April, 2006.
2. MathWorks, *Using MATLAB Manual*, The MathWorks Inc., 2007.
3. Michael I. Clune, Pieter J. Mosterman and Christos G. Cassandras, "Discrete Event and Hybrid System Simulation with SimEvents," *Proceedings of the 8th International Workshop on Discrete Event Systems*, pp. 386-387, July, 2006.
4. 임성용, 김탁곤, "DEVS 형식론에 기반한 하이브리드 시스템 모델링 시뮬레이션 방법론," 한국시뮬레이션학회 추계학술대회논문집, 2000.
5. Bernard P. Zeigler, Herbert Praehofer and Tag Gon Kim, *Theory of Modeling and Simulation*. ACADEMIC PRESS, 2001.
6. Tag Gon Kim, *DEVSim++ v3.0 Developer's Manual*, 2006, <http://smslab.kaist.ac.kr/>.
7. MathWorks, *Using SIMULINK Manual*, The MathWorks Inc., 2007.
8. James B. Dabney, Thomas L. Harman, *Mastering Simulink*, PrenticeHall, 2005.
9. Tag Gon Kim, Sung B. Park, "The DEVS Formalism: Hierarchical Modular Systems Specification in C++," *Proceedings of the European Simulation Multiconference*, pp. 152-156, 1992.



서 경 민 (kmseo@smslab.kaist.ac.kr)

2006 부산대학교 전기전자정보컴퓨터공학부 학사
 2008 KAIST 전자전산학과 석사
 2008~현재 KAIST 박사과정

관심분야 : DEVS 형식론, 하이브리드 시스템, MATLAB/Simulink



성 창 호 (chsung@smslab.kaist.ac.kr)

2003 부산대학교 전자공학과 학사
 2005~현재 KAIST 전자전산학과 석박사통합과정

관심분야 : 모델링/시뮬레이션, 시뮬레이터 연동, 하이브리드 시스템



김 탁 곤 (tkim@ee.kaist.ac.kr)

1975 부산대학교 전자공학과 학사
 1980 경북대학교 전자공학과 석사
 1988 Univ. of Arizona, 전기및컴퓨터공학과 박사
 1980~1983 부경대학교, 통신공학과, 전임강사
 1987~1989 (미)아리조나 환경연구소, 연구엔지니어
 1989~1991 Univ. of Kansas, 전기및컴퓨터공학과, 조교수
 1991~현재 KAIST 전자전산학과, 교수

한국시뮬레이션 학회 회장 역임
 SIMULATION(국제시뮬레이션 학회(SCS) 논문지)
 편집위원장 역임
 국제 학회: SCS Fellow, IEEE Senior Member, ACM Member
 자격증 : 모델링 시뮬레이션 기술사(미국)
 국방 M&S 자문위원 : 국방부, 합참, KIDA, ADD 등

관심분야 : 모델링/시뮬레이션 이론, 방법론 및 환경개발, 시뮬레이터 연동