

Simulink 기반의 Test 환경

김재열
 아주대학교 정보및컴퓨터공학과
 e-mail : picaroom@gmail.com

Automatic Testing Based on a Simulink

Kim Jae Youl
 Ajou University, Information & Computer Engineering

요 약

시스템은 자발성, 자율성, 사회성, 반응성을 갖는 독립된 프로그램인 에이전트를 조합하여 구성되는 시스템으로, 일반 사용자에게 편리하고 자연스러운 메타포를 제공한다. 그러나, 개발자 측면에서는 에이전트 시스템에서 요구하는 각종 기능 및 제약규칙...

1. 서론

Cirtification 과 Varification 은 임베디드 시스템의 개발 기간과 개발 단가의 효율적 절감이라는 측면에서 중요성이 대두되고 있다. 특히 임베디드 소프트웨어 개발을 용이하게 하기 위해 Simulink 와 Labview 와 같은 다양한 Tool 이 개발되었으며, 이와 같은 Tools 은 기존의 소프트웨어 개발 과정을 획기적으로 개선시킬 수 있는 다양한 방법을 제시하고 있다. 본 논문에서는 Simulink 를 이용한 임베디드 소프트웨어 테스트를 수행시 발생할 수 있는 문제와 테스트의 편의성을 확보하기 위한 방법을 제시하고자 한다.

2. Simuilnk

Simulink 는 modeling, code generation, simulation, debugging 등 다양한 기능을 지원하는 소프트웨어 패키지이며 임베디드 시스템 개발에 매우 널리 사용되고 있다. 시스템 전체를 하나의 모델로 모델링하는 경우도 있다. 그러나 독립된 기능을 수행하는 여러개의 module 로 분할한 후, 각 module 을 개발하고 이들을 통합하여 목적 시스템을 완성하는 것이 일반적인 접근 방법이다. 이 경우, 개별 module 이 관련된 요구사항을 충족하는 지를 검증하는 작업은 신뢰성 있는 목적 시스템 개발을 위한 작업 중 하나이다.

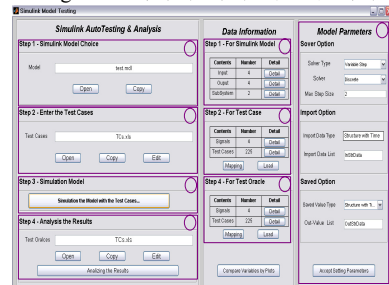
특히 테스트는 가장 현실적인 검증 방법 방안의 하나이며, module 테스트를 위해 simulation 의 기능을 사용할 수 있다. 그러나, 다음과 같은 점에서 볼 때, 단순한 simulation 으로는 module 이 관련 요구사항을 완벽하게 충족하고 있는지 검증하는 데에 한계가 있다. 첫째, Simulink model 에 복수 개의 테스트 케이스를 적용시키기 위해서는 Matlab 의 Matrix 를 이용하여야 한다. 엔지니어는 반복된 테스트 수행을 위하여 지루하고 시간 소모되는 simulation 입력 작업을 해야 한다. 둘째, 엔지니어는 테스트 결과가 요구사항에 일치하는 것인지를 확인하기 위해서 simulation 결과를 분석

해야 한다. 이러한 불편함의 극복이 가능하더라도 요구사항에 관련된 테스트 케이스를 Simulink model 이나 simulation 을 통하여 도출하는 것은 매우 어렵다. 이것이 Simulink module 의 검증에서 발생하는 가장 큰 단점이다. 결과적으로 Simulink 의 simulation 기능만으로는 model 을 테스트 할 수는 있으나 요구사항과 관련된 검증 작업에 사용하기에는 한계가 있다.

3. Plugged-in Programs

Simulink Verification & Cirtification 기능을 이용시 발생하는 문제를 해결하기 위해서는 요구사항과 Model 간의 동기화를 유지할 필요성이 있다. 이를 위해서는 첫째는 module 과 관련되는 요구사항들로부터 module 검증에 충분한 테스트 케이스들을 제공하는 것이다. 뿐만 아니라, 제공된 테스트 케이스를 Simulink model 에 적용하여 얻은 simulation 결과가 요구사항과 일치하는 지를 확인하는 데에 사용할 테스트 오라클도 함께 제공하는 것이다. 둘째로는 생성된 테스트 케이스나 테스트 오라클을 Simulink model simulation 의 입력으로 사용 가능하게 하는 것이다.

다음 Plugged-in Programs 은 Matlab 을 이용하여 구현한 프로그램으로 Module 과 요구사항과의 동기화와 Simulink Testing 을 관리하기 하기 위한 것이다.



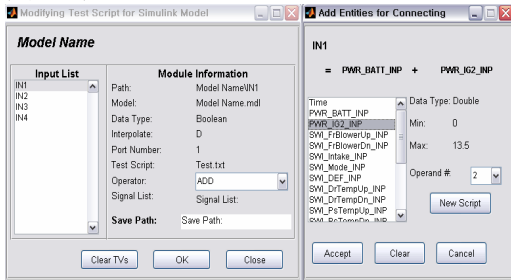
(그림 1) Plugged-in Program - Main

3-1. Step I – Input Target Model

첫째 단계인 ①은 테스트 할 대상인 Simulink model 파일을 선택하는 단계이다. ①에서 model 이 정상적으로 선택되면, Simulink model 을 분석하여 주요 정보를 (1-1)에 출력한다. 또한 (1-2)와 같은 model parameter 윈도우를 통해 현재 활성화된 model 의 Simulation Option 을 확인할 수 있다. (1-1)에 출력된 정보를 이용하면 model 의 Input, Output, Subsystem 에 관한 상세 정보를 확인할 수 있다. 특히 Input 과 Output 의 경우, 그들의 Name, Data Type, Init Value 를 확인할 수 있다. Subsystem 의 경우에는 Name, I/O 의 개수 등을 파악할 수 있다. 또한 (1-1)의 Detail Window 에서 각 Model 의 Input 과 Output Module 을 R-Bench 에서 정의된 I/O Entity 와 Mapping 할 수 있다.

3-2. Step II – Input Test Cases

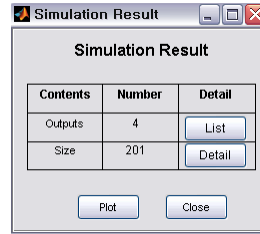
둘째 단계인 ②에서는 R-Bench 에서 생성한 테스트 케이스가 기록된 파일을 선택한다. 테스트 케이스 및 스크립트 파일의 형식은 Excel 파일 형식이다. 테스트 케이스 파일이 정상적으로 선택되면, 상세한 정보가 (2-1)에 출력된다. ‘Signals’은 테스트 케이스의 Input 변수의 개수를 의미하며, ‘Test Cases’는 테스트 케이스의 길이를 의미한다. (2-1)에서 제공하는 매핑을 이용해서 R-Bench 에서 생성된 테스트 케이스를 Simulink model simulation 에 적합한 구조로 변경한다. Simulink simulation 을 수행하기 위해서는 외부 데이터를 ‘Load’를 사용해서 Simulink model 의 입력으로 적용시켜야 한다.



(그림 2) Plugged-In Program - Mapping I/O

3-3. Step III - Simulation

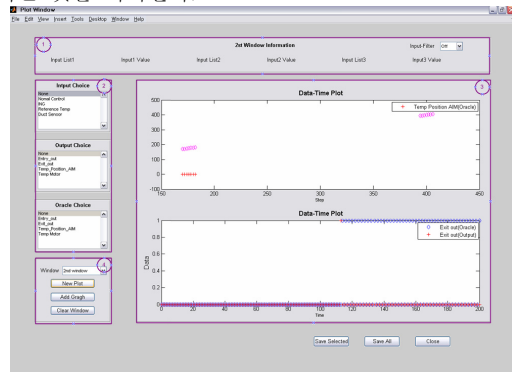
셋째 단계인 ③은 simulation 을 실행시키는 단계이다. 이 단계에서는 virtual Time 으로 model 테스트를 수행한다. Simulation 이 완료되면 수행 결과에 대한 요약 정보가 다음 (그림 3)과 같이 출력된다.



(그림 3) Plugged-In Program - Simulation Result

3-4. Step IV – Analysis

넷째 단계인 ④에서는 R-Bench 가 생성한 테스트 오라클과 simulation 결과를 하나의 그래프에 출력한다. 오라클이 수정된 경우에는 ‘Load’를 이용해 오라클을 다시 로드 할 수 있다. 다음 (그림 4)은 단계 ④를 실행하였을 때의 모습이다. Input 은 단계 ②에서 Simulation 을 위하여 설정된 테스트 케이스이다. Output 은 Simulation 결과 얻어진 값이며, Oracle 은 단계 ②에서 테스트 케이스와 함께 입력된 테스트 오라클을 가리킨다. 하나의 그래프에 테스트 결과와 오라클을 동시에 출력함으로써 테스트가 input 과 output, 그리고 오라클 사이의 관계를 직관적으로 이해할 수 있다. 또한 (그림 4)의 ①이 제공하는 Input Filter 를 이용하면 특정 입력 대비 출력을 비교할 수 있다. (그림 4)의 ①은 (그림 4)의 ③의 2 개의 윈도우 중에서 그래프를 출력한 윈도우의 Input Filter 정보를 보여준다. 그림 4-3 의 ② 는 선택된 윈도우에 그럴 그래프의 Source 를 선택하는 데 사용된다. (그림 4)의 ④는 (그림 4)의 ②에서 선택된 Source 를 표현하는 방법을 선택하는 윈도우이다. ‘New’는 기존의 그래프를 삭제하고 다시 그리는 것을 의미한다. ‘Add’는 Graph 에 추가하는 것을 의미한다.



(그림 4) Plugged-In Program - Simulation Result

3-5. Overview

이상과 같이 Matlab 의 Workspace 를 이용한 Plugged-in Program 을 이용하여 Simulink 를 이용한 Testing 환경을 구축하였다. 기본적인 테스트 Tool 은 Simulink 의 Verification & Certification Tool 을 이용함

로써 Tool 의 신뢰성을 확보하였으며, Matlab Program 을 이용함으로써 각 Test 단계의 추적 및 관리가 용이하도록 하였다.

4. To do

Real-Time Workshop 을 이용한 실시간 테스트 환경의 구축을 위해 Simulink 기반의 범용 Communicator 의 설계가 필요하며 또한 Real Time Workshop 을 이용한 Plugged-in Program 이 추가적으로 필요하다.

또한 Simulink 의 설계 기법을 이용하여 요구사항을 입력, 관리 및 Test case 의 생성이 가능하게 하는 프로그램의 작성이 추가적으로 수행되어야 한다.

참고문헌

- [1] Mathworkstm
<http://www.mathworks.com/products/simulink/>