

통합시험환경 모델 검증 자동화 설계

Design of Automatic Model Verification for System Integration Laboratory

양승구* · 조연제 · 조경용 · 류창명

LIGNex원 항공연구소

Seung-Gu Yang* · Yeon-Je Cho · Kyoung-Yong Jo · Chang-Myung Ryu

Avionic Research Institute, LIGNex1, Daejeon 34127, Korea

[요 약]

항공전자시스템을 개발하는 과정에서 개별 구성품의 기능 및 연동 검증을 위한 통합시험환경(SIL; system integration laboratory)을 구성한다. 이러한 SIL의 구성품 개별 검증 및 시스템 통합 시 각 탑재장비의 기능 및 연동을 모의하는 SIL 모델을 개발하여 이용한다. SIL 모델은 실 장비와 연동되기 전 연동통제문서(ICD; interface control document)에 정의된 모든 데이터에 대해 선 검증되어야 하며 ICD 변경 혹은 기능 변경 발생 시에도 재 검증되어야 한다. 하지만 SIL 모델의 검증의 수동 수행 시 개별 SIL 모델의 검증에도 상당한 시간이 소요된다. 이러한 시간상의 문제로 ICD 변경이나 일부 기능 변경 시에는 SIL 모델의 영향성을 판단하여 선택적 회귀시험이 이루어지곤 한다. 본 논문에서는 이러한 SIL 모델의 검증에 소요되는 시간을 최소화하여 회귀시험 수행 시 모든 시험항목의 검증이 가능하도록 SIL 모델 검증 자동화 방안을 설계하고 설계에 따른 SIL 모델 검증 자동화도구를 개발하여 SIL 모델 검증자동화 설계의 유용성을 검증하였다.

[Abstract]

In developing the avionics system, a system integration laboratory (SIL) is established to verify the function and interworking of individual components. In case of individual verification of SIL's components and system integration, a SIL model that simulates the function and interworking of each equipment is developed and used. A SIL model shall be pre-verified against all data defined in the interface control document (ICD) before interworking with the actual equipment and reverified even when the ICD changes or functions change. However, if the verification of the SIL model is performed manually, the verification of the individual SIL model takes considerable time. For this reason, selective regression tests are often performed to determine a impact of SIL models on ICD changes and some functional changes. In this paper, we designed SIL model verification automation method to perform regression test by reducing verification time of SIL model and verify the usefulness of verification automation design by developing SIL model verification automation tool.

Key word : Avionics system, System integration laboratory, Automatic verification, Model verification.

<https://doi.org/10.12673/jant.2019.23.5.361>



This is an Open Access article distributed under the terms of the Creative Commons Attribution Non-Commercial License (<http://creativecommons.org/licenses/by-nc/3.0/>) which permits unrestricted non-commercial use, distribution, and reproduction in any medium, provided the original work is properly cited.

Received 30 September 2019; Revised 21 October 2019
Accepted (Publication) 7 October 2019 (30 October 2019)

*Corresponding Author; Seung-Gu Yang

Tel: +82-42-718-3576

E-mail: seunggu.yang@lignex1.com

I. 서론

항공전자시스템은 항공기에 탑재되는 전자장비 및 각종 센서 등이 통합된 것으로 각 센서로부터 수신한 데이터를 처리 및 시현하는 기능을 제공하는 시스템을 의미하며 임무컴퓨터를 비롯한 통신장비, 식별장비, 항법장비 등을 포함한다[1]. 항공전자시스템 개발 시 각 탑재장비의 개별 검증 혹은 시스템 통합을 위해 통합시험환경(SIL; system integration laboratory)을 개발한다[2],[3]. SIL 개발 시 구성품의 개별 검증 및 시스템 통합 시 실 장비를 연동하지 못하는 경우를 위해 각 탑재장비의 기능 및 연동을 모의하는 시뮬레이션 모델을 개발하여 이용한다. 이러한 시뮬레이션 모델은 실 장비와 연동되기 전 연동통제문서에 정의된 모든 데이터에 대해 선 검증되어야하며 연동통제문서 혹은 기능 변경 발생 시에도 재 검증되어야한다.

기존 사업을 진행 시 일정 및 비용의 문제로 인해 SIL 모델의 요구사항 검증을 위한 모든 시험절차를 시험운영자가 직접 입력하여 진행하였다[4]. 그러나 시험절차를 직접 입력하여 진행하게 되면 모든 시험절차를 검증하는데 상당한 시간이 소요되어 일부 기능 변경이나 연동통제문서(ICD; interface control document) 변경 시 해당 부분의 영향성에 따른 선택적 회귀시험(regression test)이 이루어졌다.

본 논문에서 국산 상용품인 통합개발환경을 이용해 개발된 SIL 모델의 검증 시간을 줄여 선택적 회귀시험이 아닌 모든 시험절차에 대한 회귀시험이 가능하도록 SIL 모델 검증 자동화 방안을 제시하고 자동화도구를 개발하여 SIL 모델 검증 자동화 방안에 대한 유용성을 검토하였다.

II. SIL 모델 검증 자동화 설계

2-1 SIL 모델 기본 구조

SIL 모델의 구성은 그림 1과 같다. 여기서 소프트웨어 모델은 실시간 운영을 보장하기 위하여 실시간 운영체제(RTOS; real-time operating system)를 기반으로 동작한다. 소프트웨어

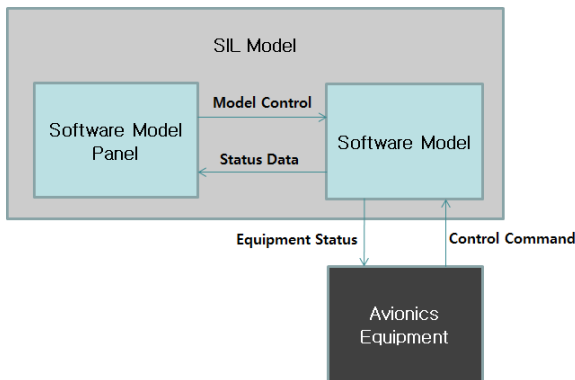


그림 1. SIL 모델 개념
Fig. 1. concept of SIL model.

모델 패널은 사용자 인터페이스(GUI; graphical user interface)를 제공하기 위해 Windows 운영체제를 기반으로 동작하며 운영자의 모델 모의 데이터 설정, 소프트웨어 모델의 입력데이터 및 출력 데이터의 시현을 담당한다[4].

SIL 모델은 항공전자시스템의 구성품과 연동되거나 항공전자시스템 구성품을 대체해 검증할 수 있는 환경을 제공해야하므로 연동통제문서(ICD; interface control document)에 의존적으로 개발되어지며 소프트웨어 모델 패널의 GUI를 통해 ICD 및 시스템 규격에 정의된 동작을 모의할 수 있도록 개발된다.

2-2 기존 SIL 모델의 검증 절차

기존의 SIL 모델 검증 절차는 4단계로 나누어 이루어 졌으며 그 흐름은 그림 2와 같다. SIL 모델 개발이 완료된 후 테스트 모델 개발, 시험절차서 작성, 시험, 시험결과서 작성의 순서로 진행된다.

첫 번째로 검증 대상 SIL 모델의 연동 및 기능에 대응하는 테스트 모델의 개발이다. 예를 들어 임무컴퓨터 검증을 위한 SIL을 개발한다고 가정했을 때 통신장비, 식별장비 등의 임무컴퓨터와 연동되는 탑재장비를 모의하는 SIL 모델의 개발이 필요하다. 이러한 SIL 모델의 검증을 위해서는 SIL 모델과 임무컴퓨터 사이의 ICD에 정의된 데이터 연동 기능을 수행할 테스트 모델이 필요하게 된다. 테스트용 모델은 ICD에 의존적으로 개발되어야 하므로 사업별로 재사용이 불가능하여 각 사업별 혹은 검증을 위한 SIL 모델별로 개발이 이루어진다.

두 번째로는 시험절차서 작성이다. 시험절차서의 작성을 위해서는 SIL 모델의 기능 및 연동데이터를 식별하여 식별된 사항에 대한 검증절차를 수립하여야하며 기 개발된 SIL 모델의 시험절차서에는 시험식별자, 데이터 입력 위치 및 입력 값, 출력위치 및 기대 값 등으로 이루어진다.

세 번째로는 작성된 시험절차서를 이용한 시험을 수행한다. 두 번째 단계에서 생성된 각 시험항목 별로 테스트 모델에 시험 값을 입력하고 테스트 모델에 수신된 출력 값과 기대 값을 비교하여 같은 결과를 나타내면 해당 시험항목에 대한 검증이 성공한 것으로 판단할 수 있다. 시험 진행 중 시험절차서 작성 오류와 테스트 모델 개발 오류, SIL 모델 오류 등이 발생할 수 있다. 시험절차서 작성 오류 시 시험절차서를 수정하고 다시 시험 진행 단계로 넘어가게 되고 테스트 모델의 오류가

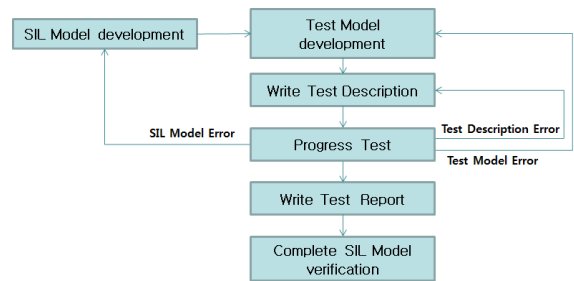


그림 2. 기존 SIL 모델 검증 절차 흐름도
Fig. 2. flow chart of SIL model verification.

발견되면 테스트 모델을 수정한 후 다시 시험 진행을 하게 된다. 만약 SIL모델의 오류를 발견하게 된다면 SIL모델의 오류를 수정하여 다시 재검증 절차에 들어가 모든 시험 절차를 진행할 수 있을 때까지 반복한다.

마지막으로 시험결과서를 작성하는 단계이다. 시험결과서는 시험절차서의 모든 항목의 시험을 진행한 후 각 시험항목별 성공/실패 여부를 기록한다.

기존 진행사업의 경우 테스트 모델이 개발 및 검증이 완료된 상황에서 시험절차를 직접 입력하는 방식으로 11종의 SIL 모델의 400여개 시험항목을 시험하는데 약 50시간 정도의 시간이 소요되었다. 이러한 시간상의 문제 ICD의 수정이나 소프트웨어 모델의 요구사항 변경 발생 시 해당 SIL 모델의 모든 시험항목을 시험하기 어려워 수정에 따른 영향성이 있다고 판단되는 부분만 회귀시험을 진행하게 되었다.

2-3 SIL 모델 자동화 검증 방안

SIL 모델 검증 자동화를 위해 먼저 자동화 가능한 부분을 식별하였다. 본 논문에서는 소프트웨어 모델 패널 검증에 대한 자동화는 추후 연구과제로 남겨두고 우선 소프트웨어 모델의 자동화를 목표로 하였다. 소프트웨어 모델 검증 자동화가 가능한 부분으로 테스트 모델의 개발 및 시험진행, 시험결과서 작성을 식별하였다. SIL 모델 자동화 검증의 자동화 부분 및 그 흐름은 그림 3과 같다.

모델 검증 자동화를 위해 먼저 테스트 모델 개발의 자동화방안을 설계하였다. 기존의 테스트 모델은 검증 대상 모델에 데이터를 입력하고 출력된 데이터를 통해 시험항목의 성공 여부를 판단한다. 이러한 기능을 자동으로 수행하기 위해서는 시험항목에 따라 입력 데이터를 삽입하고 입력에 따른 출력 데이터와 시험항목의 기댓값을 비교할 수 있도록 테스트 모델에 대한 코드를 자동 생성할 수 있도록 하였다. 이러한 방식을 사용하면 테스트 모델을 각 사업 별 ICD에 따라 개발하지 않고 작성된 시험절차서에 따라 입력데이터 및 출력데이터, 기댓값에 따라 각 시험항목을 시험하고 성공 여부를 판단하는 기능을 자동으로 수행할 수 있다. 다만 이를 위해 기존의 시험절차서에 일부 항목을 추가할 필요가 있다.

자동코드생성을 위한 시험절차서의 추가항목으로 코드생성

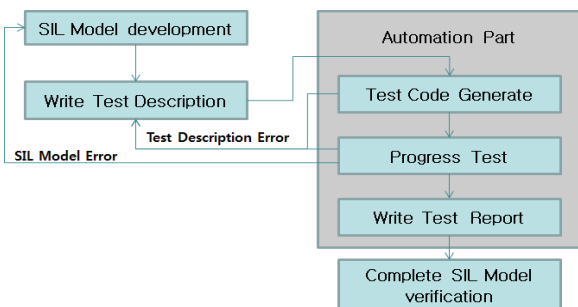


그림 3. SIL 모델 검증 자동화 절차 흐름도
Fig. 3. flow chart of SIL model autonomous verification.

에 사용할 시험항목인지 여부, 데이터 입력 후 대기 시간, 시험 절차 수행 후 대기 시간으로 3개의 항목을 선정하였다. 자동코드 생성여부는 간혹 시험에 필요하지 않거나 시험을 진행하지 않을 시험항목이 있을 경우를 위해 추가하였고 입력 후 대기 시간은 소프트웨어 모델이 입력데이터 수신 후 출력 데이터를 산출하는데 필요한 시간이며 시험항목 시험 후 대기 시간은 이전 시험의 데이터 출력에 따라 검증 대상 모델의 내부 로직의 변경이 필요한 부분에 대한 대기 시간이다. 이를 이용해 자동코드 생성기를 설계하였다.

자동코드생성기는 시험절차서 파싱 기능, 코드 자동생성기능의 두 가지 기능을 가지고 있다. 본 논문에서는 시험절차서 작성 및 시험결과서 작성의 편의성을 위해 ‘,’와 ‘carriage return’으로 구분되는 CSV(comma-separated values) 파일의 형식으로 시험절차서를 작성하였다.

시험절차서 파싱 기능의 흐름은 그림 4와 같다. 시험절차서 파싱 기능은 먼저 자동화 시험을 진행하기 위한 시험절차서를 선택한 후 해당 시험절차서 파일을 파싱한다. 시험절차서를 파싱하는 과정에서 시험절차서의 데이터를 모두 ‘carriage return’을 기준으로 토큰화하고 이것을 다시 ‘,’를 기준으로 토큰화한다. 토큰화된 데이터는 모든 데이터가 저장되는 순간까지 반복하여 저장하며 추후 자동코드생성 시 필요한 데이터 구조체의 형태로 메모리에 저장한다.

코드 자동생성기능은 파싱된 시험절차서의 시험항목 데이터를 이용하여 테스트 모델 코드를 생성하는 기능이다. 파싱된 시험항목의 내용으로는 순번, 코드생성여부, 시험항목 식별자, 입력 위치, 입력 값, 입력 후 대기 시간, 출력 위치, 기대 값, 결과 값, 시험항목 성공여부, 완료 후 대기시간이다.

코드 자동생성기능의 흐름은 그림 5와 같다. 코드 자동생성기능은 먼저 코드생성여부를 확인한다. 현재 순번의 코드생성여부가 참이면 이전의 시험 식별자와 동일한지 확인 후 동일하지 않다면 생성할 시험항목의 시험 수행 프레임 확인코드를 생성 후 데이터 입력 코드를 생성한다. 그 후 입력 후 대기할 수 있도록 대기코드를 생성한 후 출력 데이터를 저장하고 기댓값과의 비교를 통해 시험항목의 성공/실패여부를 저장한다. 마지막

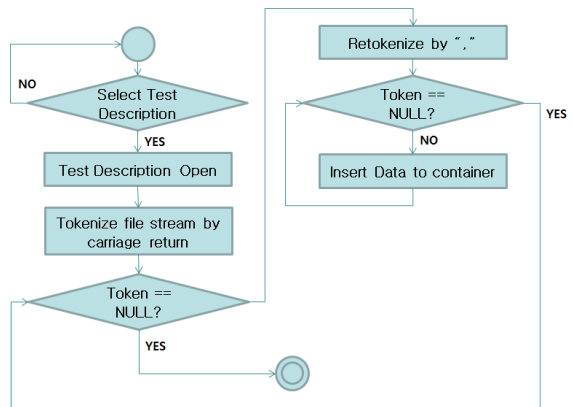


그림 4. 시험절차서 파싱 기능 흐름도
Fig. 4. flow chart of test description parsing.

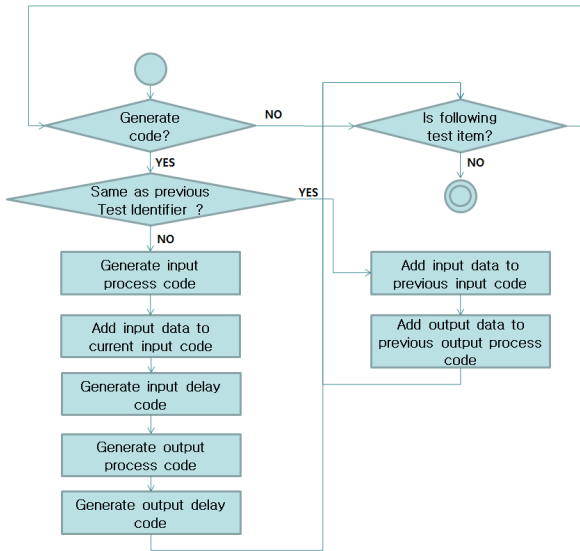


그림 5. 자동코드생성 기능 흐름도
 Fig. 5. flow chart of auto code generator.

으로 출력 대기 후 다음 시험항목을 수행할 수 있도록 코드를 생성한 후 이후의 시험항목이 있는지 확인해 다음의 시험항목이 있다면 다시 처음부터 반복한다. 만약 이전과 동일한 시험식별자를 갖는 시험항목이 있다면 기존의 데이터 입력 코드와 데이터 출력 코드에 현재 시험항목의 데이터 입력 및 데이터 출력 비교 코드를 이전의 데이터 입력 및 데이터 출력 비교 코드에 추가 삽입하여 다음 시험 항목이 있는지 확인한다.

III. SIL 모델 검증 자동화 도구 개발 및 검증

그림 6은 본 연구의 유용성 검증을 위해 개발된 자동코드생성기의 실행화면이다. 자동코드생성기는 Visual Studio 2010환경에서 C++언어를 사용하여 개발되었다.

자동코드생성기는 먼저 소프트웨어 모델의 IDE를 선택하고 테스트 모델의 실행 주기를 설정한다. 그 후 ‘Select Test Case’ 버튼을 눌러 시험절차서를 선택한 후 ‘Generate Code’ 버튼을

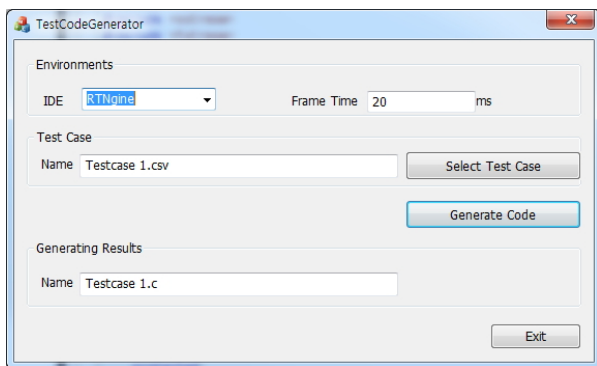


그림 6. 자동코드생성기 실행화면
 Fig. 6. GUI of auto code generator.

순번	코드 제어식별자	입력 GST 명	입력 값	입력 후 대용량 GST명	기대 값	결과 값	Pass/Fail (완료 후 대기 시간(ms))
1 O	T-STE-SFR-0010	INPUTGST1	0x01	100	OUTPUTGST1	0x03	100
2 O	T-STE-SFR-0010	INPUTGST2	0x10	100	-	-	100
3 O	T-STE-SFR-0011	INPUTGST3	0x02	100	OUTPUTGST2	0x06	100
4 O	T-STE-SFR-0012	INPUTGST4	0x00	100	OUTPUTGST3	0x00	100
5 O	T-STE-SFR-0013	INPUTGST5	0x01	200	OUTPUTGST4	0x07	200
6 O	T-STE-SFR-0013	-	-	200	OUTPUTGST5	0x10	200
7 X	T-STE-SFR-0014	INPUTGST7	0x10	200	OUTPUTGST6	4096	200

그림 7. 입력으로 사용가능한 시험절차서 예시
 Fig. 7. Example of test description.

누르면 선택한 시험절차서와 동일한 이름의 테스트 모델 코드가 생성된다. 자동코드생성기에서 IDE는 현재 국산 상용품을 선택할 수 있고 추후 외산 상용품에 대한 테스트 모델에 대한 자동코드 생성기를 수정 개발할 예정이다. 현재 기 개발된 소프트웨어 모델은 50Hz의 주기를 갖는 hard realtime system으로 테스트모델의 실행 주기는 20ms로 설정하였다. 아래 그림 7은 시험절차서의 예시이며 그림 7은 그림6의 시험절차서 예시를 입력으로하여 자동코드생성기에 의해 생성 테스트 모델의 코드이다.

그림 8에서 보는바와 같이 생성된 테스트 모델은 수행 시간을 확인하기 위한 정수형 변수와 결과 값을 저장할 배열, 시험절차서공여부를 저장할 배열인 3개의 전역변수를 선언하고 각 시험항목 식별자 별로 생성된다. 각 시험 항목은 시험식별자를 시작으로 시험을 시작할 프레임과 현재 진행시간이 같은지 확인하여 시작할 프레임이 되면 입력 위치인 공유메모리에 입력 값을 설정한다. 그 후 입력 후 대기시간이 지나면 각 출력 값과 기댓값을 비교하여 시험절차의 실패/성공 여부를 판단하고 이를 전역변수에 저장한다. 마지막으로 실제 출력된 값을 전역변수에 저장하여 추후 시험결과서를 작성하는데 쓰일 수 있도록 한다. 이러한 시험코드가 시험절차서에 정의된 모든 시험항목에 대해 생성되며 이 테스트모델 코드를 통합개발환경에 삽입하여 빌드하면 자동화 시험을 위한 모든 준비가 완료된다.

현재 개발된 자동코드생성기에서는 시험결과에 대한 검증

```

1 int g_iCounter = 0;
2 int g_iResultValue[7];
3 int g_iResult[7];
4
5 //T-STE-SFR-0010
6 if(g_iCounter == 0)
7 {
8     INPUTGST1 = 0x01;
9     INPUTGST2 = 0x10;
10
11     if(g_iCounter == 5)
12     {
13         if(OUTPUTGST1 == 0x03)
14             g_iResult[0] = 1;
15         else
16             g_iResult[0] = 0;
17
18         g_iResultValue[0] = OUTPUTGST1;
19     }
20
21 //T-STE-SFR-0011
22 if(g_iCounter == 5)
23 {
24     INPUTGST3 = 0x02;
25 }
26 if(g_iCounter == 10)
27 {
28     if(OUTPUTGST2 == 0x06)
29         g_iResult[2] = 1;
30     else
31         g_iResult[2] = 0;
32
33     g_iResultValue[2] = OUTPUTGST2;
34 }
35
    
```

그림 8. 자동코드생성기에서 생성된 테스트 모델 코드
 Fig. 8. test model code by auto code generator.

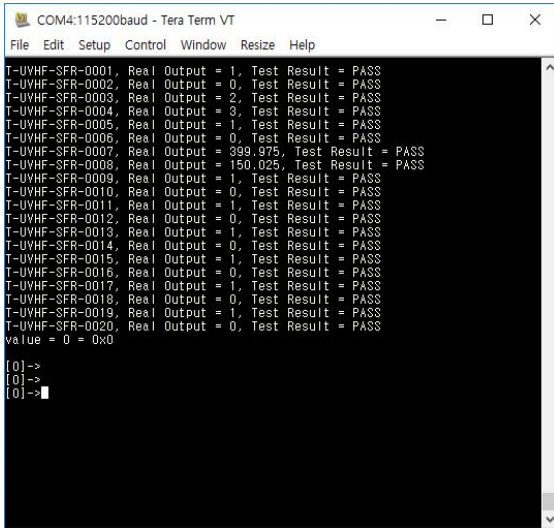


그림 9. 시험 결과 캡처화면
 Fig. 9. capture image of test result.

을 위해 시험결과서 작성을 위한 데이터를 수집하여 터미널 창에 프린트 하는 것으로 개발 되었다. 하지만 현재 시험절차서 작성 기능을 위해 시험절차서에 시험결과를 기록하는 시험결과서 자동작성기능이 추가될 예정이다. 그림 9는 기 개발된 소프트웨어 모델에서 해당 모델의 검증을 위해 사용된 시험절차서의 형식을 본 논문에서 사용될 시험절차서의 형식으로 변경하여 자동화 시험을 수행한 시험결과를 캡처한 화면이다.

IV. 결 론

본 논문에서는 SIL 모델의 검증 자동화를 위해 자동코드생성기를 개발하여 SIL 모델 검증 자동화의 유용성을 검증해 보았다. 시험 결과 기존의 수동 시험의 경우 테스트 모델이 구현된 상황에서 약 1시간이 소요되던 20개의 시험절차에 대해 시험절차서 수정을 포함하여 10분내 완료 할 수 있었다. 만약 테스트 모델을 구현하는 시간까지 포함한다면 상당한 시간적 이득을 획득한 것으로 생각할 수 있으며 추후 소프트웨어의 부분

적인 수정이 이루어 졌을 때에도 수정에 따른 영향성이 예상되는 부분뿐만 아니라 모든 시험항목에 대해 회귀시험을 수행하여 소프트웨어 모델의 신뢰성을 확보하는데 도움이 될 것으로 예상된다.

현재 개발된 자동코드생성기는 SIL 모델의 공유메모리를 이용하여 내부 변수의 입력 및 출력으로만 운용될 수 있기에 소프트웨어 모델만 검증할 수 있는 한계가 있다. 기 개발된 SIL 모델 역시 소프트웨어 모델과 소프트웨어 모델 패널로 구분되어 있으므로 소프트웨어 모델에 대한 검증뿐만 아니라 소프트웨어 모델 패널의 검증 또한 추가 검증이 필요하다. 현재 이러한 자동코드생성기의 한계점인 소프트웨어 모델 패널의 검증 자동화 부분에 대 보완하기 위한 추가 연구를 진행하는 중이며 자동코드생성기와 소프트웨어 모델 패널 검증 방안이 통합되어 개발되면 SIL 모델의 모든 시험절차에 대한 검증 자동화가 가능해질 것으로 예상하고 있다.

References

- [1] R. P. G Collinson, *Introduction to Avionics Systems*, 2nd ed. New York, NY: Springer US, 2013.
- [2] A. S. Yildirim, E. Berker, and M. E. Kayakesen, "System level test automation in UAV development," in *2018 IEEE AUTOTEST Conference*, Maryland: MD, pp. 278-283, 2018.
- [3] Y. W. Jo, B. G. Kim, J. S. Park, and J. U. Lee, "Development of system integration laboratory for the verification of UAV avionics system requirements," *Journal of The Korean Society for Aeronautical & Space Sciences*, Vol. 40, No. 5, pp. 446-453. 2012.
- [4] M. G. Seo, W. H. Jang, J. S. Park, D. B. Seong, and S. W. Kim, "Development and verification of models in system integration laboratory for the avionics system", in *Fall Conference on The Korean Society for Aeronautical & Space Sciences*, Jeju: Korea, pp. 2074-2077, 2015.



양 승 구 (Seung-Gu Yang)

2006년 2월 : 중앙대학교 전자전지공학부 (공학사)
 2008년 2월 : 중앙대학교 전기전자공학부 (공학석사)
 2008년 1월 ~ 2011년 10월 : 코원시스템 MD 사업부 대리
 2011년 11월 ~ 현재 : LIG넥스원 항공연구소 선임연구원
 ※ 관심분야 : 항공전자시스템, 내장형 소프트웨어



조연제 (Yeon-Je Cho)

2016년 2월 : 경희대학교 전자전파공학부 (공학사)
2016년 1월 ~ 2018년 3월 : GS ITM 플랜트 ICT 팀 사원
2018년 7월 ~ 현재 : LIG넥스원 항공연구소 연구원
※ 관심분야 : 항공전자시스템, 내장형 소프트웨어



조경용 (Kyoung-Yong Jo)

2012년 8월 : 건국대학교 항공우주정보시스템 (공학사)
2014년 8월 : 건국대학교 항공우주정보시스템 (공학석사)
2014년 9월 ~ 2019년 4월 : 언맨드솔루션 기술연구소 선임연구원
2019년 5월 ~ 현재 : LIG넥스원 항공연구소 선임연구원
※ 관심분야 : 비행체 형상 및 제어, 항공전자시스템



류창명 (Chang-Myung Ryu)

2011년 8월 : 경북대학교 컴퓨터공학부 (공학사)
2014년 2월 : 건국대학교 항공우주정보시스템공학부 (공학석사)
2014년 9월 ~ 현재 : LIG넥스원 항공연구소 선임연구원
※ 관심분야 : 항공전자시스템, 내장형 소프트웨어