

역공학을 이용한 ECU의 Hardware-In-the-Loop Simulation

박지명 · 함원경 · 고민석 · 박상철*

Hardware-In-the-Loop Simulation of ECU using Reverse Engineering

Ji-Myoung Park · Won-Kyung Ham · Min-Suk Ko · Sang-Chul Park*

ABSTRACT

Increasing the proportion of an embedded system in automotive industry, test methods for evaluation and fault detection of the embedded system have been researched. HILS is a test method that is used in the development and test of complex real-time embedded systems. In this study, we defined the HILS method of the ECU, one of the embedded systems used in automobiles. Our method is to create a test model that can provide a virtual vehicle environment to the ECU on the basis of the actual vehicle data. The test model has reference information that can transmit the sensor signal and CAN Message into the ECU from HILS tester. In this study, the HILS can detect faults of the target ECU.

Key words : Hardware In the Loop Simulation (HILS), Reverse engineering, Embedded System, Electronic Control Unit, Engine Management System ECU (EMS ECU)

요약

자동차 산업에서 임베디드 시스템의 비중이 증가하면서, 임베디드 시스템의 불량 검출과 평가를 위한 시험 방법들이 연구되고 있다. 하드웨어 인 더 루프 시뮬레이션은 실시간 임베디드 시스템의 개발 및 시험을 위해서 사용되는 시험 방법이다. 본 연구는 자동차에서 사용되는 임베디드 시스템인 ECU의 하드웨어 인 더 루프 시뮬레이션 방법에 대해서 정의한다. 이 방법은 실제 차량 데이터를 기반으로 ECU에게 가상차량환경을 제공할 수 있는 시험 모델을 작성할 수 있다. 시험 모델은 시험기로부터 ECU에게 센서 신호와 CAN 메시지를 모사할 수 있는 참조 정보를 가진다. 본 연구의 하드웨어 인 더 루프 시뮬레이션은 목표한 ECU의 불량을 검출할 수 있다.

주요어 : 하드웨어 인 더 루프 시뮬레이션, 역공학, 임베디드 시스템, 전자제어장치, 엔진제어장치

Nomenclature

VSD (Vehicle sensor data) : 실제 차량에서 추출한 센서 데이터

1. 서론

전자기술의 급속한 발전에 힘입어 자동차 분야에서 임베디드 시스템(Embedded system)의 비중은 지속적으로 증가하고 있다¹⁻³⁾. 임베디드 시스템은 기계 또는 전자 장치에 두뇌 역할을 하는 마이크로 프로세서(Microprocessor)를 장착하여 제어를 하는 시스템으로, 기기를 동작하는 소프트웨어를 칩에 담아 기기에 내장시킨 형태의 장치를 의미한다. 일반적으로 임베디드 시스템의 검증은 하드웨어의 불량, 소프트웨어의 코드 오류, 임베디드 시스템의 동작을 시험하는 3가지 분야로 나눌 수 있다. 하드웨어의 불량을 검증하는 시험은 단선 시험(Open test), 합선 시험(Short circuit test), 지락 시험(Grounding test) 등이 있고,

* 이 연구는 한국연구재단 (2015R1A2A2A01005871)의 중견연구자 과제를 통한 지원으로 수행되었습니다.

Received: 26 January 2016, **Revised:** 26 February 2016, **Accepted:** 16 March 2016

***Corresponding Author:** Sang-Chul Park
E-mail: scpark@ajou.ac.kr
Ajou University, Industrial Engineering

소프트웨어의 코드 오류를 검증하는 시험은 정확성 테스트(Correctness test), 강건성 테스트(Robustness test), 완벽성 테스트(Completeness test) 등이 있다^[4]. 마지막으로 임베디드 시스템의 동작을 검증하는 시험은 타겟 시스템에 장착하여 수행하는 필드 테스트(Field test)와 타겟 시스템에 장착된 것처럼 모사하는 HILS(Hardware-in-the-loop simulation)가 있다.

HILS는 순수 시뮬레이션이 갖는 부정확성과 필드 테스트의 고비용 및 위험성을 보완하기 위한 테스트 방법으로^[5], 자동차·항공우주·국방 분야의 임베디드 시스템 검증에 사용되고 있다.^[6, 7] 이렇게 자동차·비행기·전차 등과 같은 이동수단의 상태 및 동작을 제어하기 위해 장착되는 임베디드 시스템을 ECU(Electronic Control Unit)라고 부른다. ECU에게 타겟 시스템에 장착된 것과 같은 환경을 모사해주려면, 가상 차량 환경(Virtual vehicle environment)이 필요한데^[8], 대부분의 연구는 이를 위한 환경의 구축 방법으로 모델 기반 설계 기법(Model based design)을 사용하고 있다^[8-12].

모델 기반 설계 기법은 구축하고자 하는 시스템의 각 모듈 별로 수학적인 알고리즘을 사용하여 모델링함으로써 가상 차량 환경을 구축하는 방법이다. 이 기법은 전자 기계적인 컴포넌트, 제어 알고리즘, 통신 네트워크 등을 포함하는 전체 시스템을 모델링 할 수 있는 방안을 제공해 줌으로써 디자인 검증은 물론 각 디자인의 요소들과

요구사항 및 검증 테스트와의 연계가 가능하도록 지원해 준다. 따라서 ECU와 같은 제어시스템 개발을 비롯하여 엔진 및 트랜스미션과 같은 물리 시스템, 센서, 액추에이터 그리고 운전자의 조작과 같은 외부 시스템 등을 모델링 하여 HILS를 가능하게 한다^[13, 14]. 모델 기반 설계 기법은 모델 구축 기간 단축, 높은 시험 자유도, 재사용성 등의 장점을 가지고 있지만 모델이 실제 차량 동역학을 따르는지에 대한 검토(Validation)와 실시간 계산이 어려운 문제점을 가지고 있다^[15].

따라서 본 연구에서는 VSD를 역공학(Reverse Engineering)으로 분석할 수 있는 분석방법을 제안하였다. 이를 바탕으로 입력 시나리오를 작성함으로써, 모델 기반 설계 기법을 사용하지 않고 EMS ECU(Engine Management System Electronic Control Unit)의 시험에 필요한 가상 차량 환경을 구축하였고 이를 바탕으로 HILS를 수행하였다.

2. HILS 환경 구축

2.1 EMS ECU의 HILS를 위한 가상 차량 환경

EMS ECU는 엔진 및 운전과 관련된 센서들로부터 전자 신호를 받고 내부 알고리즘에 의하여 액추에이터들에게 전자 신호를 보냄으로써 엔진을 동작하여 자동차 운전 에 관여한다. EMS ECU가 실제 차량에 장착되었을 때의 프로세스는 Fig. 1(a)와 같고, HILS를 수행하기 위해서 가

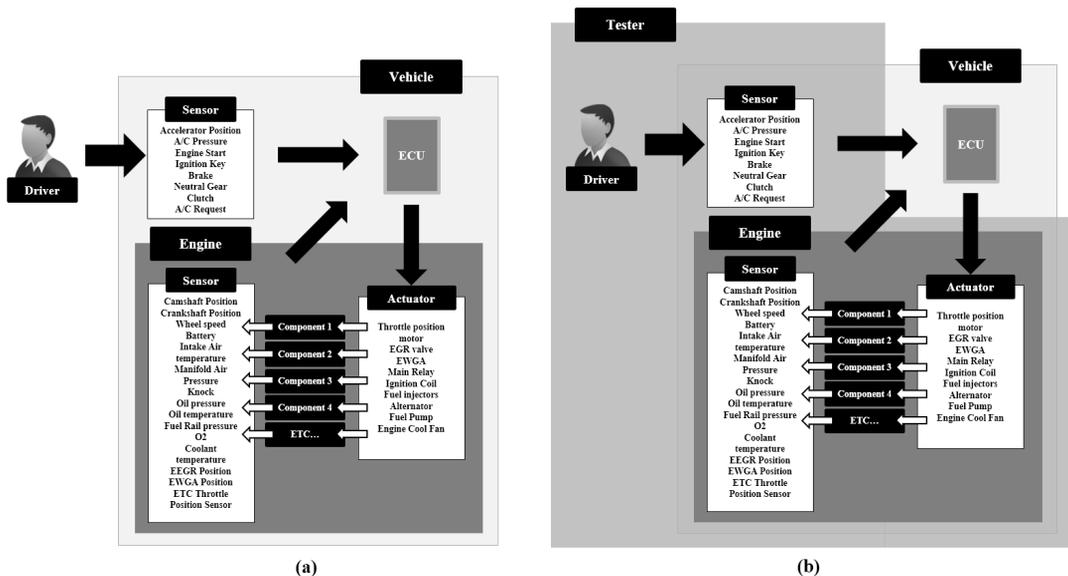


Fig. 1. Operating Process (a) Vehicle, (b) Tester

상 차량 환경을 구축했을 때의 프로세스는 Fig. 1(b)와 같다.

가상 차량 환경을 구축하는 것은 운전 에 따른 센서 신호들을 EMS ECU에게 송출해줄 수 있고, EMS ECU가 내부 알고리즘에 의한 명령을 액추에이터에게 전달할 수 있는 가상 환경을 제공하는 것이다. 앞서 언급한 기존 연구들은 이러한 가상 차량 환경을 구축하기 위하여 모델 기반 설계 기법을 사용하였다. 엔진 등의 물리 시스템, 운전 전자 조작 등의 외부 시스템을 부품 또는 기능 별로 모델링함으로써 엔진의 동작과 센서 신호를 모사하는 것이다. 모델 기반 설계 기법의 단점은 각 부품과 부품들의 상호작용이 동역학적으로 표현되기 위해서 실시간 시뮬레이션이 가능한 정확한 엔진 모델을 구축하는 것에 비용과 시간이 많이 소요되는 것이다.

A. Palladino 외 2명은 이러한 단점을 극복하고자 실제 차량데이터를 사용하여 HILS를 수행하였다^[15]. VSD를 EMS ECU에게 그대로 송출해줌으로써 복잡한 모델이나 시험 장비 없이 시험을 수행할 수 있었지만, 시나리오를 수정할 수 없는 것이 한계점이었다.

2.2 역공학적 HILS 수행 절차

모델 기반 설계 기법을 이용하지 않고 HILS를 수행하기 위한 방법으로 역공학을 이용하였다. VSD를 분석하여 함수로 표현하고, 이를 포함한 시험 모델을 작성하는 절차를 Fig. 2와 같이 구성하였다. 절차는 아래와 같이 총 12단계로 진행된다.

- 1) 목표 제품 및 주행 시나리오 선정
- 2) 차량 데이터 수집 및 정상제품의 DTC(Master Sample) 확인
- 3) 시험모델 메타 정보 입력
- 4) 핀 정보 작성
- 5) 차량 데이터 불러오기
- 6) 차량 데이터를 핀 정보에 할당
- 7) 신호 분석
- 8) 명령(함수) 입력 및 수정
- 9) 모델 핀 정보와 시험기 H/W 연결
- 10) HILS 테스트 수행
- 11) DTC 확인 및 비교 판정
- 12) 모델 이상원인 분석

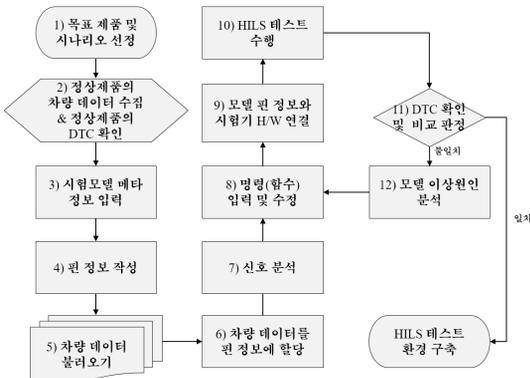


Fig. 2. HILS flow chart

- 4) 핀 정보 작성
- 5) 차량 데이터 불러오기
- 6) 차량 데이터를 핀 정보에 할당
- 7) 신호 분석
- 8) 명령(함수) 입력 및 수정
- 9) 모델 핀 정보와 시험기 H/W 연결
- 10) HILS 테스트 수행
- 11) DTC 확인 및 Master Sample과 비교 판정
- 12) 모델 이상원인 분석

단계 2)에서 확인하는 DTC(Diagnostic Trouble Codes)는 OBD-II(On-Board Diagnostics version II)에서 제공하는 고장 정보이다. OBD-II는 자동차에 부착된 센서들로부터 ECU로 전달된 자동차의 주요 계통에 대한 정보나 고장 정보를 관리하고 CAN을 이용하여 자동차의 콘솔이나 외부장치에서 확인할 수 있게 하는 진단시스템이다^[16]. 단계 11)에서 정상제품의 DTC는 HILS의 시험 결과로 나온 DTC와 비교하기 위한 Master Sample 역할을 한다. 이 Master Sample을 기준으로 같은 DTC 결과가 나올 경우 합격 판정을, 추가로 다른 DTC가 발생할 경우 불합격 판정을 한다. 단계 8)부터 12)까지는 시험모델의 적합성을 판단하기 위해 다수의 sample을 대상으로 정상 시나리오와 오류 시나리오 두 가지 방법으로 시험을 반복 수행한다. 정상 시나리오는 EMS ECU가 사용하는 모든 핀을 시험기와 연결해주는 시나리오이고 오류 시나리오는 Fig. 3(a)와 같이 EMS ECU와 시험기의 일부 핀을 단

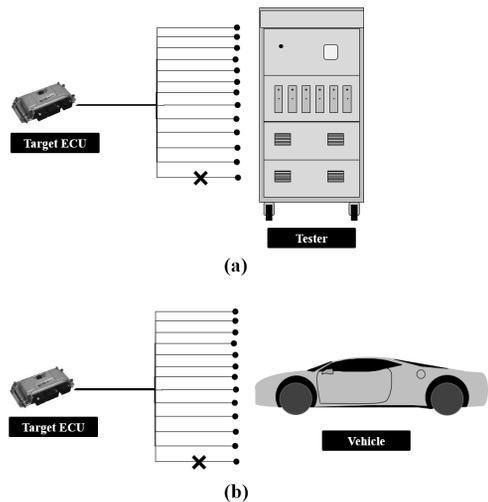


Fig. 3. Error case scenario (a) fulfill(Tester's situation), (b) Purpose(Vehicle's situation)

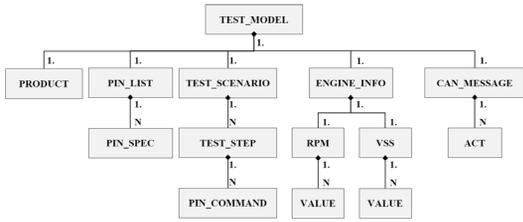


Fig. 4. Test model overall structure

선 시켜서 Fig. 3(b)와 같이 차량 입장에서 신호 단선 상황을 연출하는 오류 시나리오이다. 정상 시나리오에서 모두 합격 판정을 받고 오류 시나리오에서 모두 불합격 판정을 받을 경우, EMS ECU가 타겟 시스템에 장착되었을 때와 같은 결과를 제공하기 때문에 목표 제품에 대한 HILS 환경이 구축되었다고 판단한다.

2.3 가상 차량 환경 모델 정의

실제 차량속도, RPM, 센서 데이터, CAN(Controller Area Network) 메시지 등을 표현할 수 있는 시험 모델 구조를 Fig. 4와 같이 정의하였다. PRODUCT구조는 EMS ECU 제품정보(제품 번호 등)와 시험 모델 정보(작성 일자, 버전 등)를 정의하고, PIN_LIST구조는 EMS ECU 제품의 핀 구성 정보와 핀 별 세부 정보(번호, 종류, 설명 등)를 정의한다. TEST_SCENARIO구조는 시나리오 이름, 스텝 구성 정보(스텝 이름, 종료 시간 등)를 정의하고 각 스텝마다 PIN_LIST구조에서 정의한 핀들이 신호를 송출할 수 있도록 핀들의 명령을 정의한다. ENGINE_INFO 구조는 엔진 정보(엔진 종류, 기통 수, Crank Tooth 수, Cam 위상 정보 등)와 차량데이터에서 추출한 RPM, 차량속도 데이터를 정의함으로써 시험에서 CKP(Crank Position) Sensor 신호, CMP(Cam Position) Sensor 신호, VSS (Vehicle Speed Sensor) 신호를 만들어서 송출할 수 있도록 구성한다. CKP, CMP 신호는 각각 Crank와 Cam의 표면을 관찰하는 센서 신호이기 때문에 파형의 패턴이 Crank와 Cam의 표면 모양에 영향을 받고 파형의 Period가 엔진의 분당 회전 속도인 RPM의 영향을 받는다. 따라서 ENGINE_INFO구조의 RPM, Crank Tooth 수, Crank Tooth 모양, Cam 위상 정보를 이용하여 CKP, CMP 신호를 생성해야 한다. 또한 VSS 신호는 센서에서 측정되는 파형의 Period의 길이를 통해서 차량속도를 도출하기 위한 신호로서 ENGINE_INFO구조의 차량속도 데이터를 바탕으로 VSS 신호를 역으로 생성해줄 수 있다. 마치

Table 1. Pros and cons of engine model

| Engine Model | Pros and cons | |
|---------------------|----------------------------------|-------------|
| | Flexibility of testing scenarios | Validation |
| Model based | High | Hard |
| Vehicle data based | Low | Unnecessary |
| Vehicle model based | Middle | Easy |

막으로 CAN_MESSAGE구조는 시험에서 사용될 캔 메시지를 정의한다.

본 연구의 모델을 사용한 시험 방법은 VSD를 가공하여 함수(명령 및 입력 수식)로 표현함으로써 시나리오의 수정이 가능하고 검토가 쉬운 장점을 가지고 있다. 이에 대한 설명은 Table 1과 같다. 명령 및 입력 수식에 대해서는 3.3절에서 설명하겠다.

3. 차량 데이터 분석을 통한 시험 시나리오 작성

3.1 차량 데이터의 종류

분석할 차량의 데이터는 EMS ECU에게 입력해줄 센서들의 데이터로서 그 핀의 종류에 따라서 디지털 스위치 핀, 아날로그 핀, 파형 핀으로 구분할 수 있고, 각 종류별로 구성되어있는 인풋 핀의 예시는 Table 2와 같다.

Table 2. Signal type of input pins

| Pin name | Type | Description |
|----------|----------------|-----------------------------------|
| ENGSS | Digital Switch | Engine Start Switch |
| IGN | Digital Switch | Ignition Key on |
| HL | Digital Switch | Head Lamp |
| WIPER | Digital Switch | Wiper Switch |
| OILT | Analog | Oil Temperature Sensor |
| KS | Analog | Knock Sensor |
| CRSPOS | PWM | Crankshaft Position Sensor Signal |
| CMPOS1 | PWM | Cam Position Sensor 1 |
| CMPOS2 | PWM | Cam Position Sensor 2 |
| VSS | PWM | Vehicle Speed Signal |
| ALT | PWM | Alternator Load State |

각 디지털 스위치 핀은 전압이 High인 상태와 Low인 상태에서 특정 전압을 가지고, High일 때 동작하는 핀을 Active High핀, Low일 때 동작하는 핀을 Active Low핀이라고 부른다. 디지털 스위치 핀은 자동차의 동작에 따라 High 또는 Low상태를 유지한다. 아날로그 핀은 일정한 전압을 유지하는 핀과 자동차의 동작에 따라 변동하는 핀이 존재한다. 파형 핀은 자동차의 동작에 따라 파형의 Min, Max, Duty, Period가 바뀔 수 있다.

3.2 역공학적 데이터 분석 방법

노이즈를 제외하고 전압이 특정한 모양을 갖는 최소 단위를 한 패턴(Pattern)으로 정의하면 가능한 패턴은 Static 패턴, Slope 패턴, Pulse 패턴으로 존재하고 각 패턴이 가지고 있는 정보는 Fig. 5와 같다. 각 테스트 스텝 안에서 디지털 스위치 핀의 신호는 Static 패턴의 합으로, 아날로그 핀의 신호는 Static 패턴과 Slope 패턴의 합으로, 마지막으로 파형 핀의 신호는 Pulse 패턴의 합으로 구성된다. Fig. 6은 아날로그 핀의 신호를 패턴으로 구분한 그림이다.

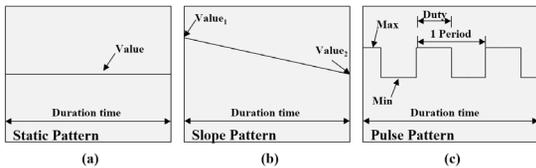


Fig. 5. Information for each pattern

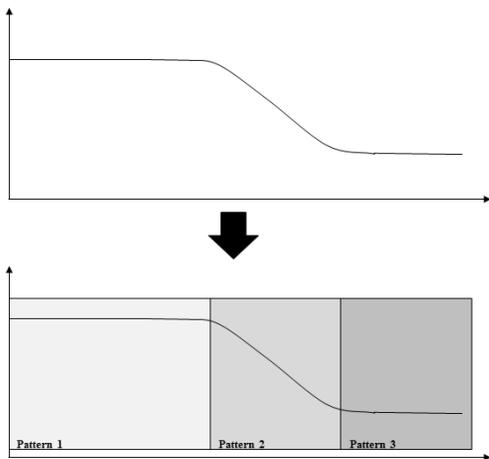
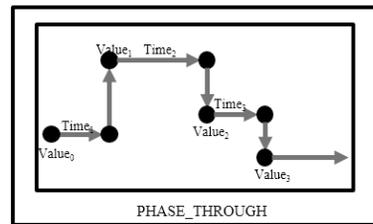


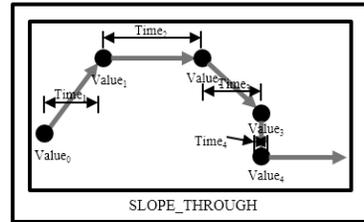
Fig. 6. Pattern division of analog signal

3.3 명령 (Command) 정의

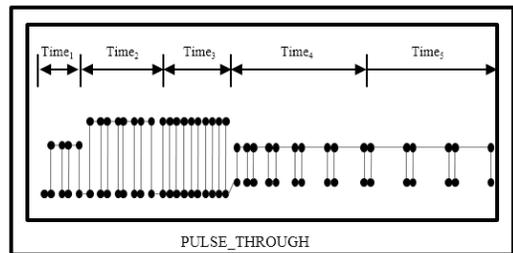
명령은 각 테스트 스텝에서 핀들의 행동을 모사하기 위한 함수이다. 각 명령에 따라 특정한 입력 수식을 가지고, 시험기에서는 명령 종류와 입력 수식에 따라 신호를 생성해주거나 상태를 지정해준다. 명령은 총 다섯 개로 VSD를 직접 정의할 수 있는 명령 3개(PHASE_THROUGH, SLOPE_THROUGH, PULSE_THROUGH), 모델에 있는 정보를 이용하여 신호를 생성하는 명령 1개(READ), 출력 핀과 접지 핀에 대한 연결을 표현해주는 명령 1개(LOAD)로 구성되고 각 명령에 따른 입력 수식과 설명은 Table 3과 같다. VSD를 직접 정의할 수 있는 명령들에 대한 설명 그림은 Fig. 7과 같다.



(a)



(b)



(c)

Fig. 7. Information for input signal commands: (a) PHASE_THROUGH, (b) SLOPE_THROUGH, (c)PULSE_THROUGH

Table 3. Commands for test

| Command | formula | Description |
|---------------|---|---|
| PHASE_THROUGH | “Value ₀ ;Time ₁ ; Value ₁ ;…;Time _(n-1) ;Value _(n-1) ; Time _(n) ;Value _(n) ” | Command for the digital switch signal. Change the voltage from Value ₀ to Value ₁ after Time ₁ past. Perform (n) times. (n ≥ 0) |
| SLOPE_THROUGH | “Value ₀ ;Time ₁ ; Value ₁ ;…;Time _(n-1) ; Value _(n-1) ;Time _(n) ;Value _(n) ” | Command for the analog signal. Change the voltage from Value ₀ to Value ₁ gradually for Time ₁ . Perform (n) times. (n ≥ 0) |
| PULSE_THROUGH | ““Min ₀ ;Max ₀ ;Duty ₀ ;Frequency ₀ ; … ;Time _(n-1) ;Min _(n-1) ;Max _(n-1) ;Duty _(n-1) ; ;Frequency _(n-1) ;Time _(n) ;Min _(n) ;Max _(n) ; ;Duty _(n) ;Frequency _(n) ” | Command for the Pulse signal. Change the voltage from Pulse ₀ (Min ₀ ;Max ₀ ;Duty ₀ ;Frequency ₀) to Pulse ₁ (Min ₁ ;Max ₁ ;Duty ₁ ;Frequency ₁) after Time ₁ past. Perform (n) times. (n ≥ 0) |
| READ | "Data Name; Purpose; Min; Max" | Command for CKP, CMP, VSS and CAN. Generate signal using RPM and vehicle speed information in the test model. (Data Name: ENGINE_INFO, TEACHING) (Purpose: CKP, CMP, CMP2, VSS) |
| LOAD | "OPEN" or "CLOSE" | Representation for connection of output pins and actual load pins. |

3.4 시나리오 생성 방법

EMS ECU의 시험에 필요한 시험모델을 작성하려면 시나리오를 정하고 EMS ECU와 연결되는 모든 핀들의 신호를 시나리오에 맞게 표현해주어야 한다. SCENARIO 구조는 Fig. 8과 같이 다수의 TEST_STEP구조로 구성되고 각 TEST_STEP구조는 PIN_LIST구조에서 정의된 핀 개수만큼 명령을 정의한다. 입력 핀들 중 CKP, CMP, VSS신호는 ENGINE_INFO구조의 정보를 이용하여 READ 명령으로 정의하고 Throttle Position Sensor는 실제 부하로 수행하기 때문에 LOAD명령을 사용한다. 그 외에 다른 입력 핀들은 앞서 설명한 PHASE_THROUGH, SLOPE_THROUGH, PULSE_THROUGH로 정의한다. Fig. 9, 10은 FTP-72(Federal Test Procedure-72) 시나리오의 VSD와 모델 정보로 생성한 디지털 스위치 신호 및 아날로그 신호를 보여준다. FTP-72는 자동차의 성능을 평가하기 위한 미국의 표준 주행 시나리오이다¹⁷⁾. CAN 신호는

| Name | Value |
|---------------|----------|
| TEST_STEPS_ID | SCENARIO |
| DESC | 자랑 시나리오 |

6 Subtags:

| Tag name/Text | STEP_ID | END_TIME | DESC | REF_PIN_NO | PIN |
|---------------|---------|----------|----------------|------------|-----------------------|
| TEST_STEP 10 | 2 | | 셋업 | | PIN (100 occurrences) |
| TEST_STEP 20 | 4 | | 시동 | | PIN (100 occurrences) |
| TEST_STEP 30 | 6 | | Immobilizer 통과 | | PIN (100 occurrences) |
| TEST_STEP 40 | 30 | | 주행 | | PIN (100 occurrences) |
| TEST_STEP 50 | 32 | | DTC 확인 | | PIN (100 occurrences) |
| TEST_STEP 60 | 34 | | 시험 종료 | | PIN (100 occurrences) |

Fig. 8. Configuration information of scenario

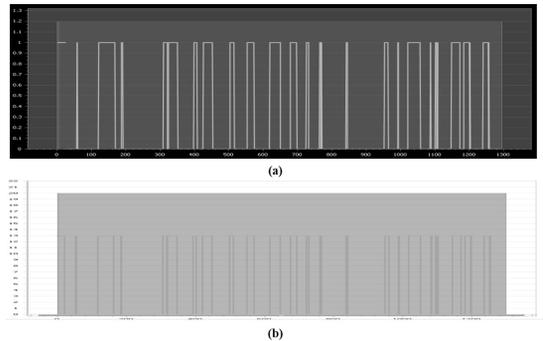


Fig. 9. Signal of digital switch pin: (a) by actual vehicle data, (b) by modeling

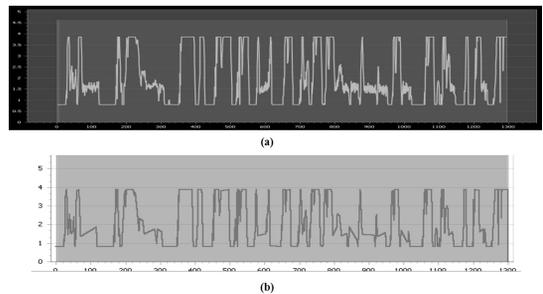


Fig. 10. Signal of analog pin: (a) by actual vehicle data, (b) by modeling

Table 4. DTC code list resulted from error case scenario

| Disconnected pin | DTC code occurrence |
|------------------|---------------------|
| KS | P0325 |
| MAP | P0106 |
| Oxygen | P0134 |
| Injector 1 | P0201 |
| APS 1 | P0123 |

Immobilizer를 사용하는 차량의 경우, CAN_MESSAGE 구조의 정보와 READ 명령을 이용하여 통신으로 시동을 허락하는 CAN 메시지를 주고받는데 사용한다. 모든 출력 핀들은 EMS ECU가 내부 알고리즘에 의하여 신호를 송출해주기 때문에 HILS 시험기에게 신호를 송출할 수 있도록 LOAD 명령으로 연결해준다. 접지 핀들 또한 EMS ECU와 연결해주는 LOAD 명령을 사용한다.

4. HILS 결과

4.1 테스트 수행 결과

다른 성능의 엔진에 사용되는 12개의 제품을 목표 제품으로 선정하고 이에 대해서 시험모델을 작성하였다. 작성된 시험모델과 함께 2.2절에서 언급한 것과 같이 EMS ECU의 모든 핀을 시험기에 연결하는 정상 시나리오와 일부 핀을 단선시키는 오류 시나리오로 시험을 수행하였다. 정상 시나리오는 각 제품마다 20개의 정상제품 sample을 대상으로 수행한 결과 Master Sample의 DTC와 일치하여 240개의 제품 모두 합격 판정을 받았다. 오류 시나리오는 각 제품의 대표 sample을 대상으로 Table 4의 단선 핀들에 대하여 시험을 수행한 결과, Master Sample의 DTC에 추가로, 단선시킨 핀에 대한 DTC가 동일하게 발생하여 모두 불합격 판정을 받았다. 따라서 본 연구의 HILS 테스트 방법이 EMS ECU가 실제 차량에 장착되었을 때와 같은 DTC 결과를 제공해주는 것을 시험을 통해 검증할 수 있었다.

5. 결론

본 연구에서는 실제 차량 데이터를 분석하여 시험모델을 작성하는 역공학적인 방법으로 HILS를 수행하기 위해서, 이를 위한 모델 구조, 차량 데이터 분석 방법, 시나리오 작성 방법을 정의하였다. 비록 제안된 방법이 모델 기반 설계 기법에 비해서 새로운 시나리오의 작성에 오랜

시간을 소요하지만, 아래와 같은 장점을 가진다.

- 1) 시험 모델이 실제 차량의 동역학을 따르는 것에 대한 검토가 쉽다.
- 2) 복잡한 알고리즘을 필요치 않아 실시간 계산이 간단하다.
- 3) VSD를 EMS ECU에게 그대로 송출해주는 시험 방법과 비교하여 시나리오의 확인 및 수정이 가능하다.
- 4) 표준화된 양식이기 때문에 관리가 용이하다.

본 연구에서 정의한 차량 데이터 분석 방법 및 시나리오 작성 방법은 EMS ECU뿐만 아니라 다른 종류의 ECU와 다른 분야의 임베디드 시스템 시험에도 적용될 수 있을 것으로 판단된다.

References

1. I. Arsie, G. Betta, D. Capriglione, A. Pietrosanto and P. Sommella (2014), "Functional testing of measurement-based control systems: An application to automotive", Measurement, Vol. 54, pp. 222-233.
2. H. Cho (2014), "Design and Application of Accelerated Run-in Test for ECU Quality Improvement", Transaction of KSAE, Vol. 22, No. 4, pp. 145-151.
3. J. Baek (2011), "A Study on Reliability Evaluation of Embedded Software in Vehicle", Transaction of KSAE, Vol. 19, No. 4, pp. 1-7.
4. S. Lee and S. Shin (2014), "Software Fault Injection Test Methodology for the Software Verification of ISO 26262 Standards-based," Transaction of KSAE, Vol. 22, No. 3, pp. 68-74.
5. D. Jung and K. Min (2009), "Study on the Development of HILS System Using HEV's ECUs," KSAE Annual Conference Proceedings, pp. 2887-2892.
6. H. K. Fathy, Z. S. Filipi, J. Hagen and J. L. Stein (2006), "Review of Hardware-in-the-Loop Simulation and Its Prospects in the Automotive Area", Proceeding of SPIE, Vol. 6228.
7. J. Cho, D. Choi, S. Yoo and W. Lee (2000), "PC-Based Real-Time Driving Simulation", Proceeding of the Korea Society for Simulation, pp. 192-197.
8. D. Ramaswamy, R. McGee, S. Sivashankar, A. Deshpande, J. Allen, K. Rzemien and W. Stuart (2004), "A Case Study in Hardware-In-the-Loop Testing: Development of an ECU for a Hybrid Electric Vehicle", SAE Technical Paper Series, 2004-01-0303.
9. E. Bringmann and A. Kramer (2008), "Model-based Testing of Automotive Systems", International Conference on Software Testing, Verification, and Validation, pp.

- 485-493.
10. W. Lee, S. Park and M. Sunwoo (2004), "Towards a seamless development process for automotive engine-control system," *Control Engineering Practice*, Vol. 12, pp. 977-986.
 11. L. Jun, G. Ying-qing and W. Hai-Quan (2008), "Rapid Prototyping Real-Time Simulation Platform for Digital Electronic Engine Control", *IEEE, and Validation*, pp. 1-5.
 12. R. Isermann, J. Schaffnit and S. Sinsel (1999), "Hardware-in-the-loop simulation for the design and testing of engine-control systems", *Control Engineering Practice*, Vol. 7, pp. 643-653.
 13. Q. R. Butt, A. I. Bhatti, M. R. Mufti, M. A. Rizvi and I. Awan (2013), "Modeling and online parameter estimation of intake manifold in gasoline engines using sliding mode observer", *Simulation Modeling Practice and Theory*, Vol. 32, pp. 138-154.
 14. T. Park, C. Han and S. Lee (2005), "Development of the electronic control unit for the rack-actuating steer-by-wire using the hardware-in-the-loop simulation system," *Mechatronics*, Vol. 15, pp. 899-918.
 15. A. Palladino, G. Fiengo and D. Lanzo (2012), "A portable hardware-in-the-loop (HIL) device for automotive diagnostic control systems," *ISA Transactions*, Vol. 51, pp. 229-236.
 16. C. You and Y. Ko (2015), "Development of the Vehicle Diagnosis Program Using OBD-II", *Transaction of KSAE*, Vol. 23, No. 3, pp. 271-278.
 17. N. Jeong, S. Yang, K. Kim, S. Choi, M. Wang, S. You, H. Kim and M. Suh (2015), "Development of Urban Driving Cycle for Performance Evaluation of Electric Vehicles Part II : Verification of Driving Cycle," *Transaction of KSAE*, Vol. w23, No. 2, pp. 161-168, 2015.



박 지 명 (jimpark89@ajou.ac.kr)

2014 아주대학교 산업공학과 학사
2014~현재 아주대학교 대학원 산업공학과 석사과정

관심분야 : 모델링 및 시뮬레이션, 제조 시스템



함 원 경 (lunacy@ajou.ac.kr)

2011 아주대학교 산업정보시스템공학부 학사
2011~2013 아주대학교 대학원 산업공학과 석사
2013~현재 아주대학교 대학원 산업공학과 박사과정

관심분야 : 모델링 및 시뮬레이션, 분산 시뮬레이션 시스템, 제조 시스템 모델링



고 민 석 (sebastianminsuk@gmail.com)

2007 아주대학교 산업정보시스템공학부 학사
2007~2013 아주대학교 대학원 산업공학과 박사
2013~현재 현대케피코, 매니저

관심분야 : 모델링 및 시뮬레이션, Hardware-In-the-Loop Simulation, PLC



박 상 철 (scpark@ajou.ac.kr)

Ph.D. (2000) in Industrial Engineering, Dept of I.E., KAIST, Korea
B.S. (1994) in Industrial Engineering, Dept of I.E., KAIST, Korea
M.S. (1996) in Industrial Engineering, Dept of I.E., KAIST, Korea
2000년 9월~2001년 12월 큐빅테크, 선임연구원
2002월 1년~2004년 2월 DaimlerChrysler ITM Dept. Research Engineer
2008년 2월~현재 아주대학교 산업정보시스템 공학부, 정교수

관심분야 : 시뮬레이션, 제조 시스템, 이산사건 모델링, CAD/CAM, PLC