

MODEL VERIFIER

모델 동적 검증 도구

TABLE OF CONTENTS

Why MODEL VERIFIER

Needs

Solution

Feature

Specification

Reference

SUPERESORT

Why MODEL VERIFIER

차별점

 Responsibility	 Performance	 Convenience	 Customizing
국내 유일의 모델 동적 검증 도구	타사 대비 빠른 분석 속도/테스트 케이스 생성	엑셀 형식의 편리한 테스트 케이스 편집기	고객사 자체 보고서 구현 (별도 협의)

강점

 도구품질	▪ 도구와 모델의 편리한 연동 ▪ 핵심 커버리지 측정	 도구연동	▪ 모델 데이터 자동 추출 ▪ 실차 데이터 가져오기 지원
 국제인증	▪ Cert. from SGS TÜV ▪ Tool Qualification 제공	 유연성	▪ 자체 개발 및 공급 ▪ 고객 요구사항에 대한 빠른 대처

기대효과

MODEL VERIFIER

- 01 모델과 오토 코드의 일치성 확보
- 02 사용자 중심의 인터페이스 제공
- 03 테스트 케이스 편집 및 관리 기능
- 04 요구사항에 따른 테스트 케이스 관리
- 05 테스트 자동화 환경 구축



Mission Critical 도메인은 안전성 및 신뢰성 인증 필요

- 테스트를 통해 구조적인 커버리지 목표를 달성해야 함
- 모델과 코드의 일치성 테스트 필요



Table 12 — Methods for software unit testing

Methods		ASIL			
		A	B	C	D
1a	Requirement-based test	++	++	++	++
1b	Interface test	++	++	++	++
1c	Fault injection test ^a	+	+	+	++
1d	Resource usage test ^b	+	+	+	++

1e Back-to-back test between model and code, if applicable^c

^b Some aspects of the resource usage test can only be evaluated properly when the software unit tests are executed on the target hardware or if the emulator for the target processor supports resource usage tests.

^c This method requires a model that can simulate the functionality of the software units. Here, the model and code are stimulated in the same way and results compared with each other.

Table 14 — Structural coverage metrics at the software unit level

Methods		ASIL			
		A	B	C	D
1a	Statement coverage	++	++	+	+
1b	Branch coverage	+	++	++	++
1c	MC/DC (Modified Condition/Decision Coverage)				

1b Branch coverage
1c MC/DC (Modified Condition/Decision Coverage)

NOTE 2 In the case of model-based development, software unit testing can be executed in different environments, for example:

NOTE 3 Software unit testing can be executed in different environments, for example:

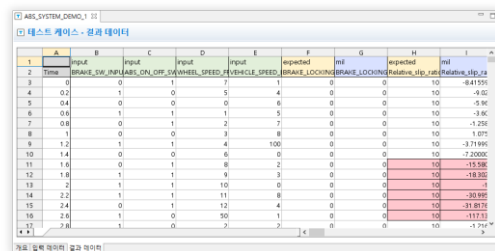
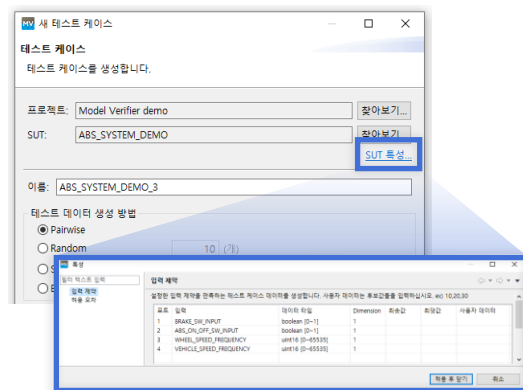
- Model-in-the-loop tests;
- Software-in-the-loop tests;
- Processor-in-the-loop tests; and
- Hardware-in-the-loop tests.

NOTE 4 For model-based development, software unit testing can be executed in different environments, for example:

- Model-in-the-loop tests;
- Software-in-the-loop tests;
- Processor-in-the-loop tests; and
- Hardware-in-the-loop tests.

모델 동적 검증 프로세스

1 테스트 케이스 자동 생성

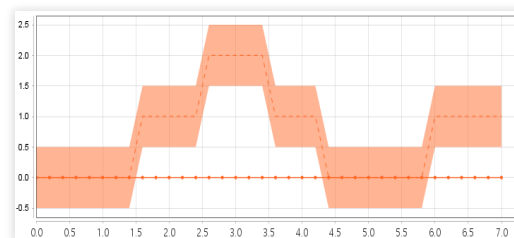


01 다양한 데이터 생성 방법 지원

02 엑셀 형태의 편리한 편집기

2 테스트 케이스 실행 결과 확인

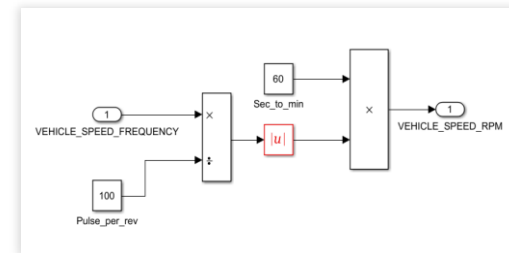
F	G	H	I
expected	mil	expected	mil
BRAKE_LOCKING	BRAKE_LOCKING	Relative_slip_ratio	Relative_slip_ratio
0	0	10	-8.415598
0	0	10	-9.024
0	0	10	-5.964
0	0	10	-3.606
0	0	10	-1.2588
0	0	10	1.0752
0	0	10	-3.719998
0	0	10	-7.200001
0	0	10	-15.5808
0	0	10	-18.3028
0	0	10	-19
0	0	10	-30.9952
0	0	10	-31.81761
0	0	10	-117.132



01 기대 값과 결과 값 자동 비교

02 입/출력 시그널 그래프 제공

3 모델 커버리지 측정



설정 | 테스트 케이스 | Top

테스트 케이스: Total

ABS_SYSTEM_DEMO/ABS_Controller/Shaft_speed_acquisition/
Vehicle_speed_acquisition/Abs (Abs)

Decision: 50%

조건	True	False
input < 0	X	O

Relational Boundary: 50%

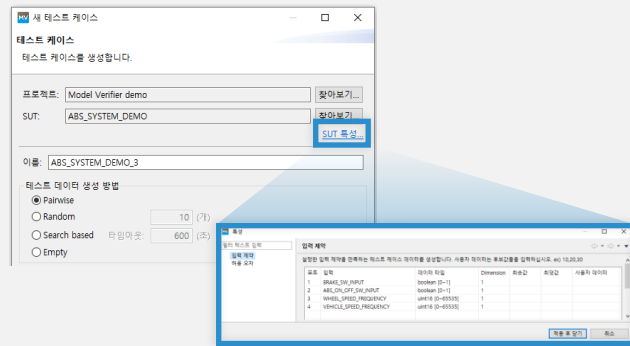
설명	결과
-tol <= input-0 < 0	X
0 <= input-0 <= tol	O

01 모델 커버리지 정보 제공

02 커버리지 상세 정보 확인 가능

테스트 케이스 자동 생성

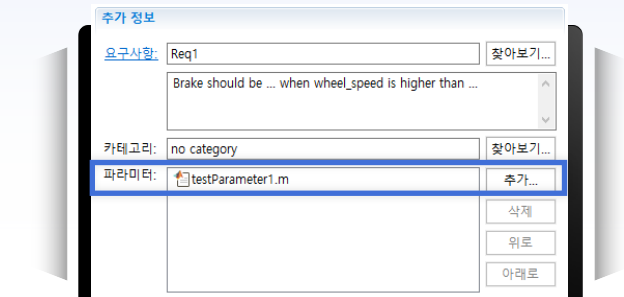
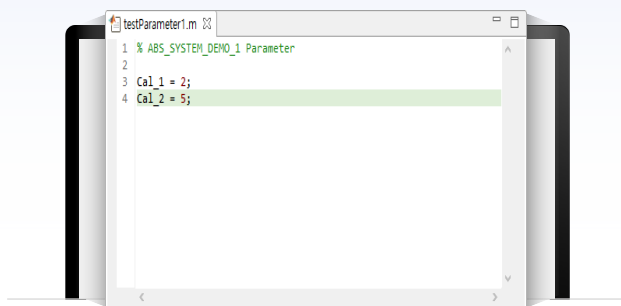
- 시스템 또는 서브시스템 단위의 테스트 케이스 자동 생성
- 테스트 데이터 생성 방법 및 입력 제약에 따라 테스트 데이터 생성



	A	B	C	D
1		input	input	input
2	Time	BRAKE_SW_INPU	ABS_ON_OFF_SW	WHEEL_SPEED_FF
3	0	0	1	7
4	0.2	1	0	5
5	0.4	0	0	0
6	0.6	1	1	1
7	0.8	0	1	2
8	1	0	0	3

Parameter 설정

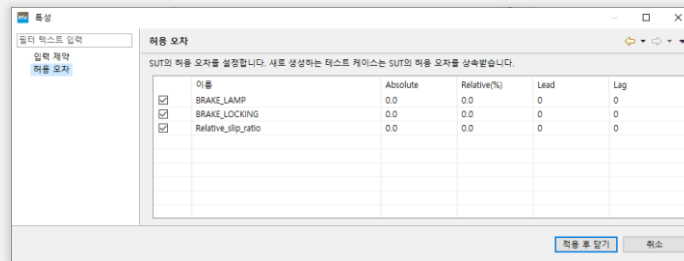
모델에서 참조하는 변수 값 등을 스크립트 파일로 관리 및 테스트 케이스에 적용하여 테스트 가능



테스트 수행 전 실행할 파라미터 파일 설정

쉬운 테스트 케이스 편집

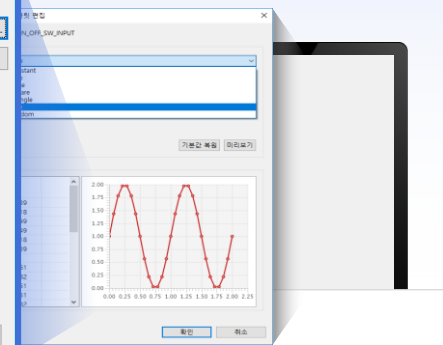
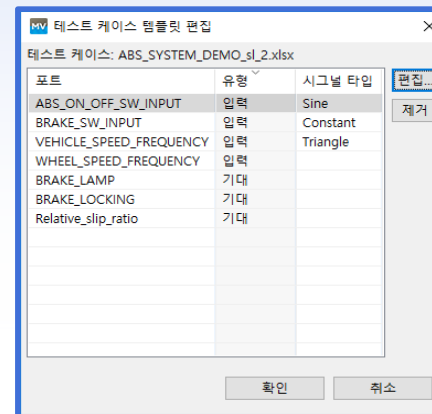
- 테스트 케이스 관련 요구사항, 카테고리, 파라미터, 허용오차 설정 지원
- 엑셀 형태의 테스트 데이터 편집기 지원



Time	INPUT01	INPUT02	INPUT03	INPUT04	OUTPUT01	OUTPUT02	OUTPUT03
0	1	2	9	7	1	2	9
10	2	4	5	31	2	4	5
20	0	6	1	0	6	1	0
30	1	8	2	1	8	2	1
40	5	10	6	1	2	9	5
50	4	12	11	2	4	5	1
60	3	14	19	0	6	1	0
70	9	16	2	1	8	2	0
80	1	18	13	5	10	6	0
90							
100							
110							
120							
130							
140							
150							
160							
170							
180							
190							
200							
210							
220							

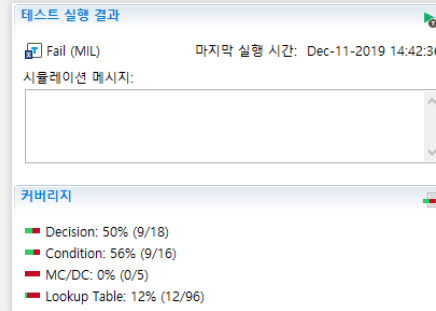
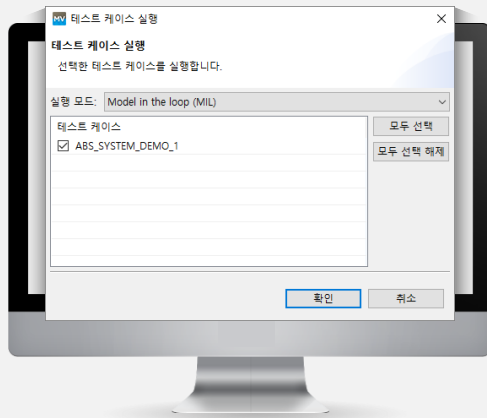
시그널 템플릿 지원

쉬운 테스트 케이스 편집을
위하여 **7가지의 시그널 템플릿**
(Sine, Pulse, Triangle, Square 등)을 지원



다양한 테스트 모드

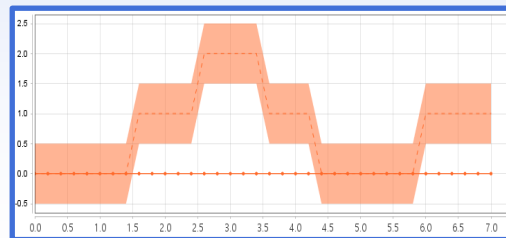
- MIL, SIL, Back-to-back 테스트 자동 수행
: 기댓값 비교, MIL 및 SIL 시뮬레이션 뿐만 아니라, 모델과 코드의 일치성 검증을 위한 Back-to-back 테스트 지원



G	H	I
expected	mil	
E_LOCKING	Relative_slip_ratio	Relative_slip_ratio
0	10	-8.415598
0	10	-9.024
0	10	-5.964
0	10	-3.606
0	10	-1.2588
0	10	1.0752
0	10	-3.719998
0	10	-7.200001
0	10	-15.5808
0	10	-18.3028
0	10	-19
0	10	-30.9952
0	10	-31.81761
0	10	-117.132

테스트 결과 시각화

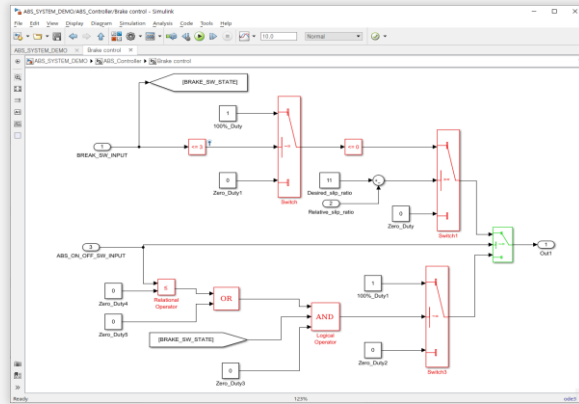
시그널 그래프와 Time Step별 비교 결과를 제공하여 쉽게 결과 확인



F	G
expected	mil
BRAKE_LOCKING	BRAKE_LOCKING
0	0
0	0
0	0
1	0
1	0
1	0
1	0

모델 커버리지 측정

- Decision, Condition, MC/DC, Lookup Table 등 총 12종 커버리지 측정
- 원본 모델에 측정 커버리지를 시각화 및 상호 연동되는 커버리지 다이어로그를 통한 상세 커버리지 제공



상호 연동

Model Verifier - ABS_SYSTEM_DEMO

테스트 케이스: Total

ABS_SYSTEM_DEMO/ABS_Controller/Brake control/Logical Operator (Logic)

Condition: 67%

조건	True	False
input point 1	O	X
input point 2	O	O
input point 3	X	O

MC/DC: 0%

조건	True	False	결과
input point 1	TTT(O)	FFx(O)	X
input point 2	TTT(O)	TFx(O)	X
input point 3	TTT(O)	TTT(O)	X

MCC: 50%

결과	결과
FFx	X
TFx	O
TTT	O
TTT	X

Model Verifier - ABS_SYSTEM_DEMO

테스트 케이스: Total

ABS_SYSTEM_DEMO/ABS_Controller/Brake control/Switch2 (Switch)

Decision: 100%

결과	결과
input point 1	O
input point 3	O

커버리지 제공

총 12종의 상세 커버리지 측정 및 제공

커버리지 시그널 그래프

'TEST' 프로젝트의 'CV_TEST_MODEL_1.csv' 테스트 케이스 커버리지를 보여줍니다.

이름	Decision	Condition	MC/DC	Lookup Table
CV_TEST_MODEL.mdl	43% (22/51)	62% (16/26)	14% (1/7)	5% (16/336)
CV_TEST_MODEL	N/A	N/A	N/A	N/A
CV_Subsystem	N/A	N/A	N/A	N/A
A_CV_Subsystem	40% (8/20)	72% (13/18)	20% (1/5)	N/A
H_CV_Subsystem	67% (2/3)	N/A	N/A	N/A
B_CV_Subsystem	N/A	N/A	N/A	N/A
E_CV_Subsystem	50% (1/2)	N/A	N/A	8% (7/84)
F_CV_Subsystem	N/A	75% (3/4)	N/A	N/A
C_CV_Subsystem	100% (2/2)	N/A	N/A	5% (5/96)
CV_Chart	25% (2/8)	0% (0/4)	0% (0/2)	N/A
D_CV_Subsystem	38% (3/8)	N/A	N/A	3% (4/156)

시그널 로깅

- 테스트 케이스 실행 중 특정 모델 시그널 모니터링

로깅 블록 선택

- 실제 테스트 모델에서 로깅 블록 선택

The screenshot shows the Simulink environment for the 'ABS_SYSTEM_DEMO' model. In the Model Browser, the 'Wheel_speed_acquisition' block is selected. A blue arrow points from this block to the 'Signal Logging' configuration window. The 'Signal Logging' window shows the selected block path: 'ABS_SYSTEM_DEMO/ABS_Controller_mod/Shaft_speed_acquisition/Wheel_speed_acquisition'. The 'Output' column is set to '1'.

클릭으로 선택

로깅 데이터 확인

- 테스트 케이스 수행 후 테스트 케이스 결과 데이터 탭에서 로깅 데이터 확인

테스트 케이스 - 결과 데이터

	A	B	C	D	E	F	G	H	I
1	time	input	input	input	input	output	output	output	logging
2		BRAKE_SW	ABS_ON	WHEEL_VELOCITY	VEHICLE_VELOCITY	BRAKE_LO	Relative_slip	BRAKE_LAMP	Product_out1
3	0	1	1	10	12	0	-18.352	1	0.1
4	0.036	1	1	10	12	0	-18.352	1	0.1
5	0.072	1	1	10	12	0	-18.352	1	0.1
6	0.108	1	1	10	12	0	-18.352	1	0.1
7	0.144	1	1	10	12	0	-18.352	1	0.1
8	0.18	1	1	10	12	0	-18.352	1	0.1
9	0.216	0	9	9	5	0	-18.238	1	0.09
10	0.252	0	9	9	5	0	-18.238	1	0.09
11	0.288	0	9	9	5	0	-18.238	1	0.09
12	0.324	0	9	9	5	0	-18.238	1	0.09
13	0.36	0	9	9	5	0	-18.238	1	0.09
14	0.396	0	9	9	5	0	-18.238	1	0.09
15	0.432	1	5	10	4	0	-18.784	1	0.1
16	0.468	1	5	10	4	0	-18.784	1	0.1
17	0.504	1	5	10	4	0	-18.784	1	0.1
18	0.54	1	5	10	4	0	-18.784	1	0.1
19	0.576	1	5	10	4	0	-18.784	1	0.1
20	0.612	0	3	9	9	0	-18.1084	1	0.09

테스트 케이스 재활용 (1/2)

무결성 검사

- 테스트 케이스의 입출력 변수와 실제 모델 입출력과 비교를 통한 무결성 검사 지원

ABS SYSTEM DEMO 3

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
time	note	input	input	input	input	input	input	input	input	input	input	input	input	input	input	input	input	input	input
		brake_on	abs_on	cn	cn	cn	cn	cn	cn	cn	cn	cn	cn	cn	cn	cn	cn	cn	cn
0.0		1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
0.2		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0.4		1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
0.6		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0.8		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1.0		1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
1.2		1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
1.4		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1.6		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1.8		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

테스트 케이스 무결성 검사

테스트 케이스에 SUT와 다른 입/출력 변수가 있습니다. SUT와 일치하도록 변경하거나 삭제하십시오.

입력되지 않는 입/출력 변수

이름	SUT	입/출력	테스트 케이스 변수 이름	변경할 변수 이름
Runtime	ABS_SYSTEM_DEMO	input	ABS_ON_OFF_SW_INPUT	ABS_ON_OFF_SW_INPUT_2

☐ 테스트 케이스에 누락된 SUT 입/출력 추가

확인 취소

테스트 케이스 일괄 변경

- 여러 테스트 케이스 내용 일괄 변경 지원

프로젝트

- Model Verifier demo
 - model
 - ABS_SYSTEM_DEMO.slx
 - ABS_SYSTEM_DEMO
 - ABS_SYSTEM_DEMO_1
 - ABS_SYSTEM_DEMO_2
 - ABS_Controller
 - DEMO_hysteresis_cont
 - no SUT
 - parameter
 - testParameter1.m

새로 만들기(N)

삭제(D) Delete

커버리지 보기 F7

커버리지 모델 열기 F8

가져오기(I)...

내보내기(O)...

테스트 케이스 복사 Ctrl+C

테스트 케이스 일괄 변경

새로 고치기(F) F5

테스트 케이스 실행... Ctrl+R

테스트 케이스 무결성 검사

테스트 케이스 일괄 변경

테스트 케이스 일괄 변경

선택한 값으로 테스트 케이스를 일괄 변경합니다.

SUT: ABS_SYSTEM_DEMO 찾아보기...

요구사항: 찾아보기...

카테고리: 찾아보기...

☐ 파라미터 설정 활성화

파라미터

| 이름 | Absolute | Relative(%) | Lead | Lag |
|---------------------|----------|-------------|------|-----|
| BRAKE_LAMP | 0.0 | 0.0 | 0 | 0 |
| BRAKE_LOCKING | 0.0 | 0.0 | 0 | 0 |
| Relative_slip_ratio | 0.0 | 0.0 | 0 | 0 |

확인 취소

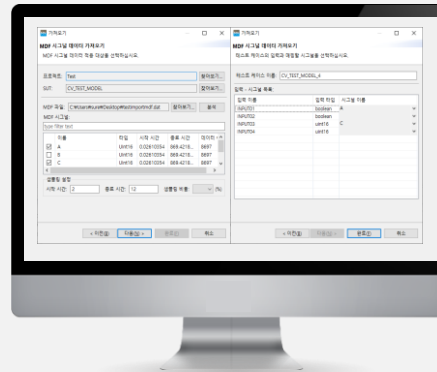
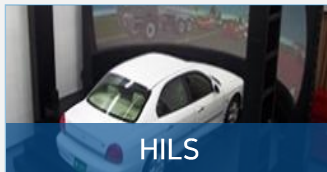
테스트 케이스 공유

- 다양한 파일 형식의 테스트 케이스 가져오기/내보내기 지원

테스트 케이스 재활용 (2/2)

실차 데이터 가져오기

- mdf 형식의 실차데이터를 분석하여 쉽게 테스트 케이스 생성

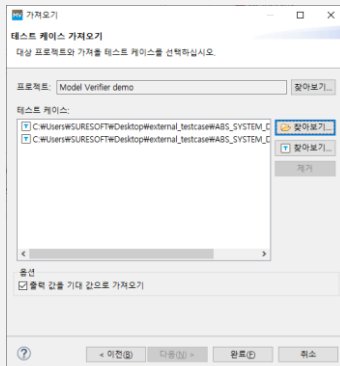


| | A | B | C | D | E |
|---|------|---------------|---------------|---------------|----------------|
| 1 | | input | input | input | input |
| 2 | Time | BRAKE_SW_INPU | ABS_ON_OFF_SW | WHEEL_SPEED_F | VEHICLE_SPEED_ |
| 3 | 0 | 0 | 1 | 7 | 1 |
| 4 | 0.2 | 1 | 0 | 5 | 4 |
| 5 | 0.4 | 0 | 0 | 0 | 6 |
| 6 | 0.6 | 1 | 1 | 1 | 5 |
| 7 | 0.8 | 0 | 1 | 2 | 7 |
| 8 | 1 | 0 | 0 | 3 | 8 |

MODEL VERIFIER

리그레션 테스트 지원

- 테스트 케이스 가져오기 시 결과 값을 기대 값으로 갖는 테스트 케이스를 생성
- 기존 데이터를 활용하여 리그레션 테스트 지원



ABS_SYSTEM_DEMO_1.1

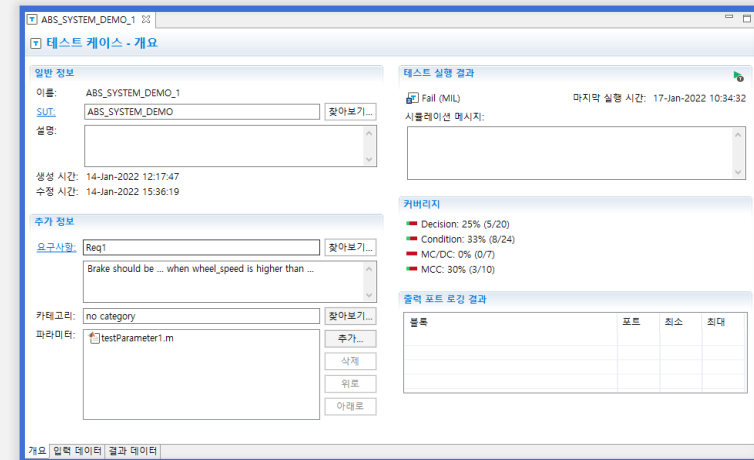
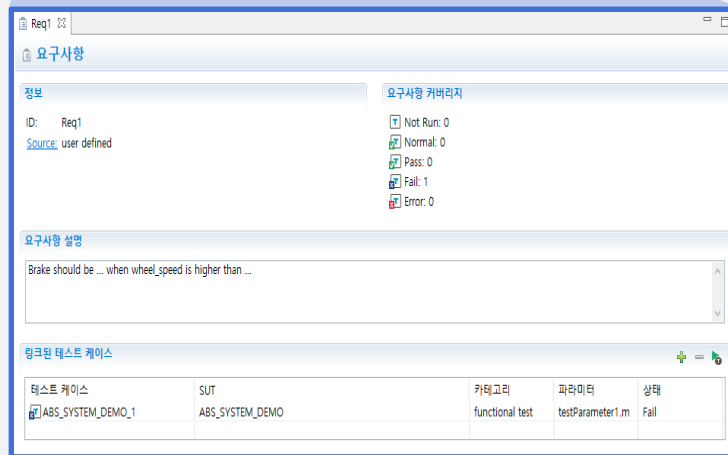
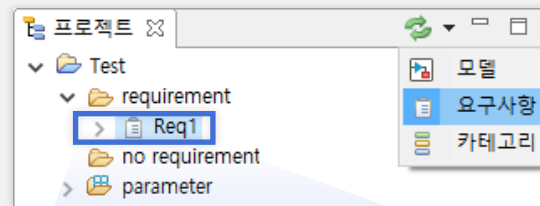
테스트 케이스 - 입력 데이터

| | A | B | C | D | E | F | G | H | I |
|----|------|------|---------------|---------------|---------------|----------------|---------------|-------------------|------------|
| 1 | | | input | input | input | input | expected | expected | expected |
| 2 | time | note | BRAKE_SW_INPU | ABS_ON_OFF_SW | WHEEL_SPEED_F | VEHICLE_SPEED_ | BRAKE_LOCKING | Relative_slip_rat | BRAKE_LAMP |
| 3 | 0 | | 0 | 1 | 7 | 9 | 0 | -8.5404 | |
| 4 | 0.78 | | 1 | 1 | 1 | 4 | 0 | -3.6048 | |
| 5 | 1.56 | | 0 | 1 | 6 | 6 | 0 | -7.3008 | |
| 6 | 2.34 | | 1 | 0 | 11 | 65535 | 0 | -387.21 | |
| 7 | 3.12 | | 0 | 0 | 12 | 4 | 0 | -31.8176 | |
| 8 | 3.9 | | 1 | 1 | 3 | 1 | 0 | 1.1844 | |
| 9 | 4.68 | | 0 | 1 | 8 | 0 | 0 | -15.6 | |
| 10 | 5.46 | | 1 | 0 | 65535 | 6 | 0 | -200487 | |
| 11 | 6.24 | | 0 | 0 | 3 | 65535 | 0 | -401.61 | |
| 12 | 7.02 | | 1 | 0 | 7 | 65535 | 0 | -1639.88 | |
| 13 | 7.8 | | 0 | 0 | 11 | 12 | 0 | -30.6928 | |

개요 | 입력 데이터 | 결과 데이터

요구사항 기반 테스트

- 요구사항 단위로 프로젝트 및 테스트 케이스 관리 가능
- 테스트 케이스와 요구사항 간의 양방향 추적성 지원



상세 환경

| 구분 | | 세부사항 |
|----------|--------|--|
| 모델 지원 | | <ul style="list-style-type: none">- Simulink 2011b 버전 이상- TargetLink 3.3 버전 이상- Embedded Coder |
| 도구 운영 환경 | OS | <ul style="list-style-type: none">- Windows 7 이상 |
| | 다국어 지원 | <ul style="list-style-type: none">- 한국어, 영어 지원 |

권장 사양

| 구분 | CPU | RAM | HDD |
|------------------|-------------------|--------|-------|
| 도구 운영 장비
(PC) | Dual Core 2GHz 이상 | 2GB 이상 | 1.5GB |



오토론 변속 제어기 모델 검증 사례

모델 동적 검증을 통하여 변속 제어기 SW의 신뢰성 확보

배경

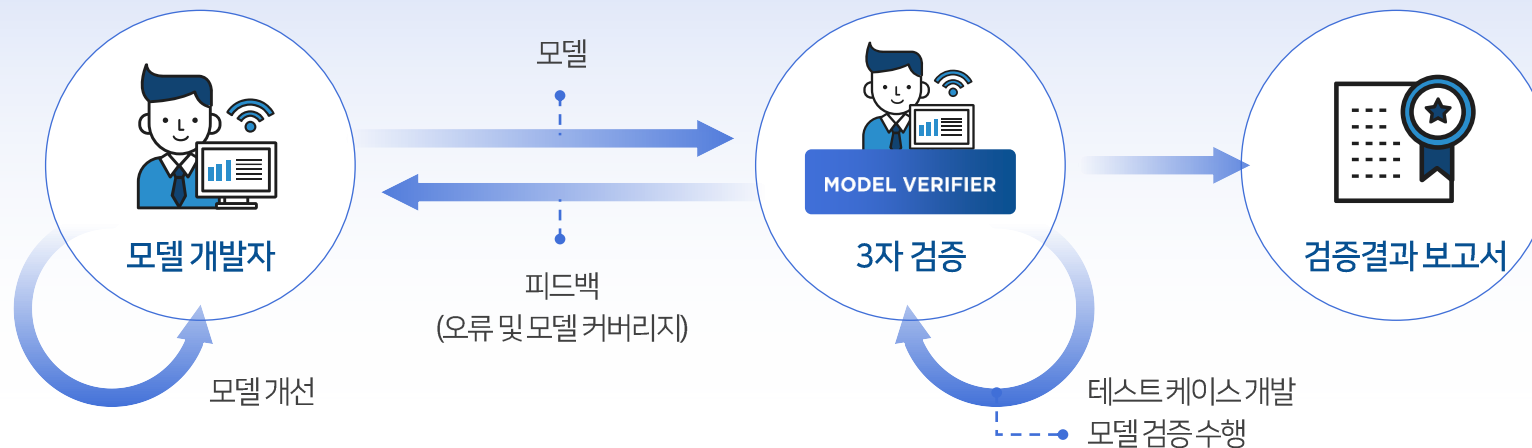
- 양산 차종에 사용될 변속 제어기 모델의 신뢰성 검증

수행 내용

- 모델 설계서 기반으로 테스트 케이스 작성 및 테스트 실행
- 목표 커버리지 달성을 위한 테스트 케이스 개발
- 모델 커버리지 분석하여 설계 오류 또는 커버리지 미달성 사유 보고

결과

- 모델 동적 검증을 통한 SW 신뢰성 향상
- 현대차 심사 기준 만족



친환경차 HIL 테스트 케이스 개선 사례

HIL 테스트 케이스에 대한 테스트 커버리지를 측정하여 HIL 테스트 케이스 개선에 활용

배경

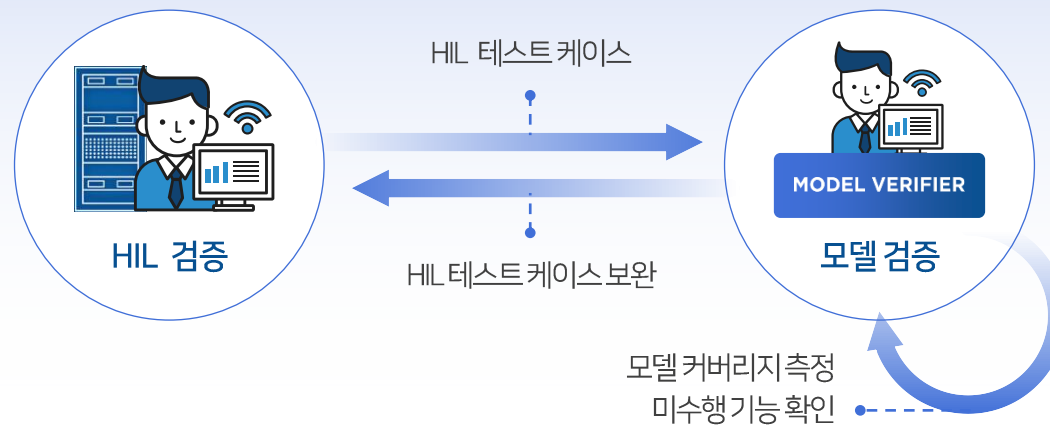
- HIL 검증 시 시험되지 않은 부분 확인 필요

수행 내용

- HIL 테스트 케이스를 모델 검증용 테스트 케이스로 자동 변환
- 변환된 테스트 케이스로 모델 검증 수행하고 커버리지 측정
- 모델 커버리지를 확인하여 HIL 테스트 케이스를 보완

결과

- 시험되지 않은 기능에 대한 HIL 테스트 케이스 보완
- HIL 검증의 정량적 완료 기준 수립



Thank you