Testing Framework with Simulink Test

김종현 부장
Test Harness (Test Scenario)

Hybrid Electric Vehicle Demo: Multi-Mode Powertrain*

Target algorithm for verification

Test Harness (Plant model)
How to Test Your Model…?

Production model

Do you need to modify it for testing?
Building Test Harness Model using Model Reference

- Separated model not for code generation but only for testing
Simulink Test
Why Simulink Test?

Saves you time:

- Creating / managing test infrastructure
- Generating & (re)-running multiple tests
- Reporting results
- Easy integration with other tools
  (Requirements, Coverage, Test Generation, MATLAB Unit Test, Continuous Integration)
- A common test environment
  – everyone doing things in a consistent manner
## Simulink Test Overview

1. **Test Harnesses**
   - Synchronized, simulatable test environment

2. **Test Stimulus Integration**
   - Inputs and assessments based on logical, temporal conditions

3. **Test Manager**
   - Author, execute, manage test cases
   - Review, export, report

---

### Main Model

**Component under test**

- **Test Harness**
- **Main Model**

**Simulink**

- **Excel**

**Report**

- Generated by Test Manager
Agenda

- Creating Test Harnesses
- Creating Test Cases & Test Stimuli
- Testing against Requirements
- Reporting
- Coverage analysis
Creating Test Harness

Specify the properties of the test harness. The component under test is the system for which the harness is being created. After creation, use the block badge to find and open harnesses.

Component under test: MultiModeCntrli_KO_R2016a_r1_Harness1

- **Basic Properties**
  - Name: MultiModeCntrli_KO_R2016a_r1_Harness1
  - Save test harnesses externally

- **Sources and Sinks**
  - [Import] ➔ [Component under Test] ➔ [Output]
  - Create scalar inputs
  - Add separate assessment block
  - Open harness after creation

[Image of the Simulink interface showing the process of creating a test harness]
What if you already have a harness model....
Agenda

- Creating Test Harnesses
- **Creating Test Cases & Test Stimuli**
- Testing against Requirements
- Reporting
- Coverage analysis
Example 1: Create a test case using the original signal builder
Create test cases with Signal Builder
What have we done so far....

- Created and imported test harnesses
- Created a test case for running multiple simulations (iterations) with different scenarios
When should I use iterations vs multiple test cases?
Comparison

- Use iterations if:
  - Only changing parameters, inputs, or configuration settings
  - Same model/harness & test type
  - Same set-up (callbacks)
  - Usually run together
  - Relate to same requirements(s)
  - Can use fast-restart

- Use separate test cases if:
  - Need independent configuration control
  - Different model/harness/test type or callbacks
  - Relate to distinct requirements
  - Distinct control of coverage
Example 2: Create a test case using real-world recorded data
Site A Wind on 23 May 2011

Wind Direction

Wind Speed / m/s

Time of day
Importing time-stamped data from Excel or text files

% pre-process .xlsx file
% get import options
importOptions = detectImportOptions('SiteWindDataRecorded.xlsx')
% set sheet
importOptions.Sheet = '2011_05_23';
% tell it that Time is in a date-time format
importOptions = setvartype(importOptions,'Time','datetime');
importOptions = setvaropts(importOptions,'Time', 'DatetimeFormat', 'HH:mm:ss.SSS ');
% read data in
T = readtable('SiteWindDataRecorded.xlsx',importOptions);
% convert to timetable
TT = table2timetable(T);
% re-sample to 1sec intervals
TTT = retime(TT,'secondly','nearest');
What have we done so far....

- Created and imported test harnesses
- Created a test case for multiple simulations (iterations)
- Created a test case importing real-world data from Excel using root import mapping
Agenda

- Creating Test Harnesses
- Creating Test Cases & Test Stimuli
- Testing against Requirements
- Reporting
- Coverage analysis
Requirement based testing

Requirements

Input Scenarios

Implementation

Dynamic Testing

Baseline

MATLAB Unit Test

Assertions

Test Sequence

and more!
Requirements Editor in Simulink Requirements
Manage and Organize Requirements

Organize with Requirement Sets

Import from External Sources

View and Author

References to crs_req.docx

Description | Rationale
--- | ---
If the Cancel switch is pressed, the value of reqDrv should be set to reqMode.Cancel.
Requirements Perspective in Simulink Requirements
Track Implementation and Verification
Example 1: Baseline test
Test types in Test Manager

- **Baseline Test**
  
  ![Baseline Test Diagram]
  
  Ex) Regression test

- **Equivalence Test**
  
  ![Equivalence Test Diagram]
  
  Ex) Back-to-Back test like SIL, PIL

- **Simulation Test**
  
  ![Simulation Test Diagram]
  
  Ex) Verifying algorithm with logical criteria
▪ **Challenges**
  – Not easy to predict expected result
  – Hard to make time-series input data

▪ **Solution**
  – Use data captured from simulation as baseline
  1. Try to run a simulation for each case.
  2. Capture output data from simulation result.
  3. Review captured data to confirm whether it is valid as baseline.
  4. Apply reviewed data to Test Manager as baseline
Baseline test using captured simulation result
Example 2:
Using verify() to test against a requirement
Test Sequence Block
Simulink Test

- A test sequence block can
  - Drive inputs (considering feedback)
  - Assess outputs with verify keyword
```matlab
if EngMode == Start
    verify(GenMode == Run);
end
```
C Caller Block Support

Verify model and hand code together

- C Caller block allows you to call a C function directly from a model
- Test the C function by creating a test harness for the C Caller block
- Author, manage and execute tests of the C function with Simulink Test
Agenda

- Creating Test Harnesses
- Creating Test Cases & Test Stimuli
- Testing against Requirements
- Reporting
- Coverage analysis
## Test Manager

### Simulink Coverage

### Simulink Design Verifier

---

**AGGREGATED COVERAGE RESULTS**

**ANalyzed MODEL**

<table>
<thead>
<tr>
<th>REPORT</th>
<th>COMP.</th>
<th>COC</th>
<th>BIC</th>
<th>MOCO</th>
<th>EXECUTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>HEV MultiMode_Optim_R2016a_R1</td>
<td>389</td>
<td>54%</td>
<td>81%</td>
<td>48%</td>
<td>100%</td>
</tr>
<tr>
<td>MultiModeCtrl_KO_R2016a_R3_ent4</td>
<td>143</td>
<td>76%</td>
<td>62%</td>
<td>41%</td>
<td>97%</td>
</tr>
<tr>
<td>Power_Management_v0</td>
<td>49</td>
<td>91%</td>
<td>97%</td>
<td>100%</td>
<td>100%</td>
</tr>
</tbody>
</table>

**AGGREGATED COVERAGE RESULTS**

<table>
<thead>
<tr>
<th>REPORT</th>
<th>COMP.</th>
<th>COC</th>
<th>BIC</th>
<th>MOCO</th>
<th>EXECUTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>HEV MultiMode_Optim_R2016a_R1</td>
<td>389</td>
<td>54%</td>
<td>81%</td>
<td>48%</td>
<td>100%</td>
</tr>
<tr>
<td>MultiModeCtrl_KO_R2016a_R3_ent4</td>
<td>143</td>
<td>76%</td>
<td>62%</td>
<td>41%</td>
<td>97%</td>
</tr>
<tr>
<td>Power_Management_v0</td>
<td>49</td>
<td>94%</td>
<td>100%</td>
<td>100%</td>
<td>100%</td>
</tr>
</tbody>
</table>

---

**Simulink Design Verifier Results**

**Progress**

- Objectives processed: 131/337
- Satisfied: 127
- Unsatisfiable: 4
- Elapsed time: 0:20

**Creating a new model from the contents of Atomic Subsystem “Multi-Mode_Hybrid_Control”**

- New Model File: C:\work\mslab2016R1\VERM_KO_R2016a_R1\sldx_output\Multi\Multi.slx
- 10-May-2016 13:39:53
- Starting test generation for model ‘Multi’
- Compiling model...done
- Translating model...done

**“MultiModeCtrl_KO_R2016a_R3_Harness1/Multi-Mode_Hybrid_Control” is compatible with Simulink Design Verify.**

**Generating tests...**
Summary

- Benefits of Simulink Test
  - Ease of creation, organisation & control of test harnesses
  - Ease of driving your models with data from various sources
  - Ease of in-harness/model verification of requirements
  - Ease of reporting
  - Ease of integration: requirements, coverage
chirpCustomCriteriaTest

testMotorSystem » System Tests » chirpCustomCriteriaTest

Baseline Test
» TAGS
- myMotorSystem

» DESCRIPTION
- REQUIREMENTS

» SYSTEM UNDER TEST
- Model: myMotorSystem
- TEST HOME
- SIMULATION SETTINGS OVER

» PARAMETER Overrides
- CALLBACKS
- INPUTS
- OUTPUTS

Figure 1: chirpFreq

w/V Gain [dB]

w/V Phase [deg]

FREQ / Hz