Testing Framework with Simulink Test
Test Harness (Test Scenario)

Test Harness (Plant model)

Target algorithm for verification
How to Test Your 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

![Test harness model](image1)

![Production model](image2)

- Signal Builder
- Model Reference (Model block)
- Scope
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

<table>
<thead>
<tr>
<th>1. Test Harnesses</th>
<th>2. Test Stimulus Integration</th>
<th>3. Test Manager</th>
</tr>
</thead>
<tbody>
<tr>
<td>• Synchronized, simulatable test environment</td>
<td>• Inputs and assessments based on logical,</td>
<td>• Author, execute, manage test cases</td>
</tr>
<tr>
<td></td>
<td>temporal conditions</td>
<td>• Review, export, report</td>
</tr>
</tbody>
</table>

### 1. Test Harnesses

- Main Model

![Main Model](Image)

- Component under test

![Component under test](Image)

### 2. Test Stimulus Integration

- Test Harness

![Test Harness](Image)

- High Wind Speed

![High Wind Speed](Image)

- Constant Wind No Yaw

![Constant Wind No Yaw](Image)

- Wind Input

![Wind Input](Image)

- Excel

![Excel](Image)

### 3. Test Manager

- Test Manager

![Test Manager](Image)
Agenda

- Creating Test Harnesses
- Creating Test Cases & Test Stimuli
- Testing against Requirements
- Reporting
- Coverage analysis
Creating 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
Common questions...

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
My 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');

% Time | WindSpeed | WindDirection
0 14.59 | 214.9
1 15.25 | 218.2
2 16.46 | 212.2
3 16.08 | 207.3
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
Requirements Perspective in Simulink Requirements
Track Implementation and Verification
Example 1: Baseline test
Test types in Test Manager

- **Baseline Test**
  - Ex) Regression test

- **Equivalence Test**
  - Ex) Back-to-Back test like SIL, PIL

- **Simulation Test**
  - 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

```
Start
speed = ramp(10); throttle = 25; verify(gear == 1);
```

Define Inputs
```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

> 14:30~150:00 Simulink를 통한 효율적인 레거시 코드 검증 방안 소개
Agenda

- Creating Test Harnesses
- Creating Test Cases & Test Stimuli
- Testing against Requirements
- Reporting
- Coverage analysis
Test Manager
Simulink Coverage
Simulink Design Verifier
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