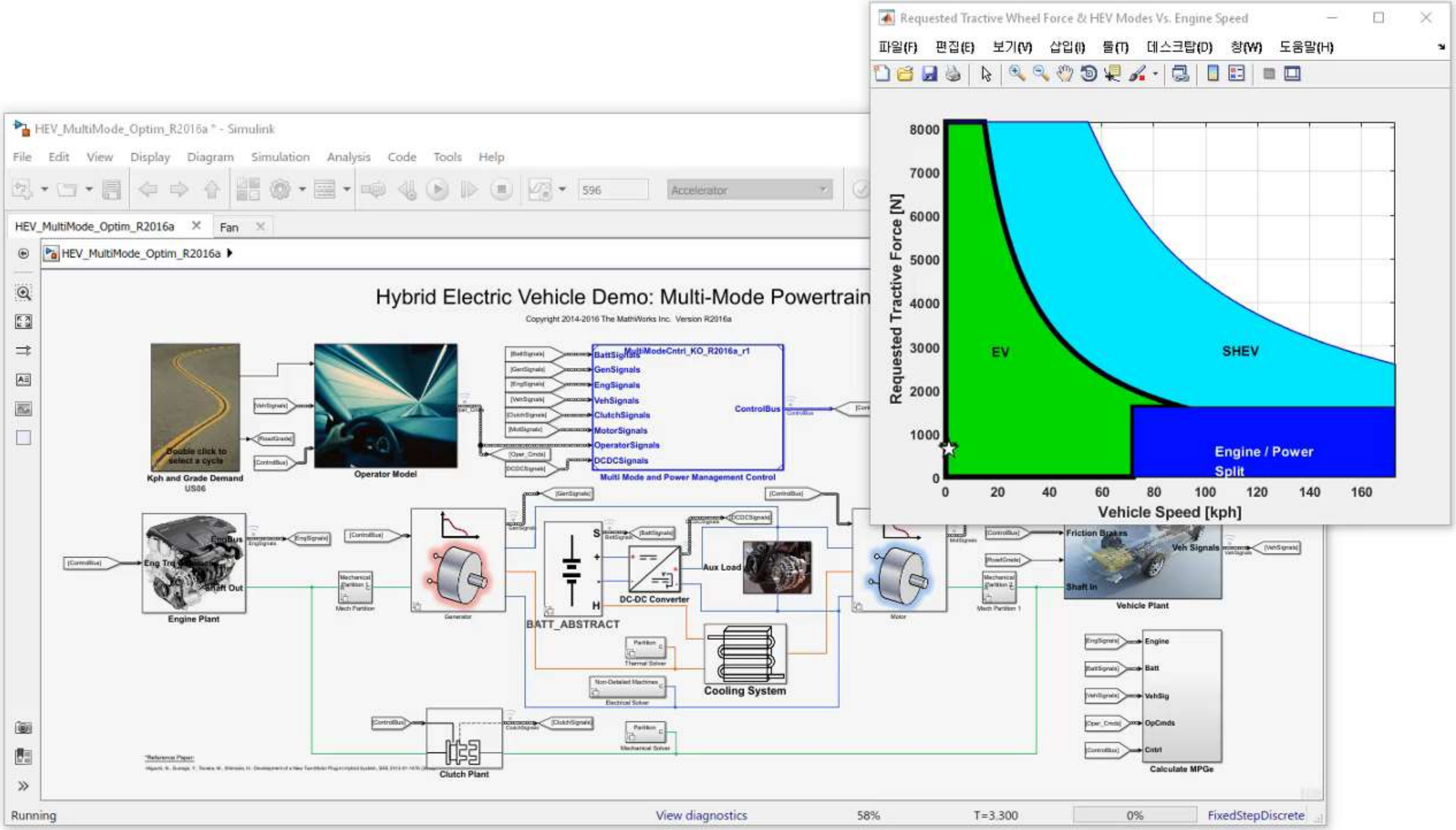
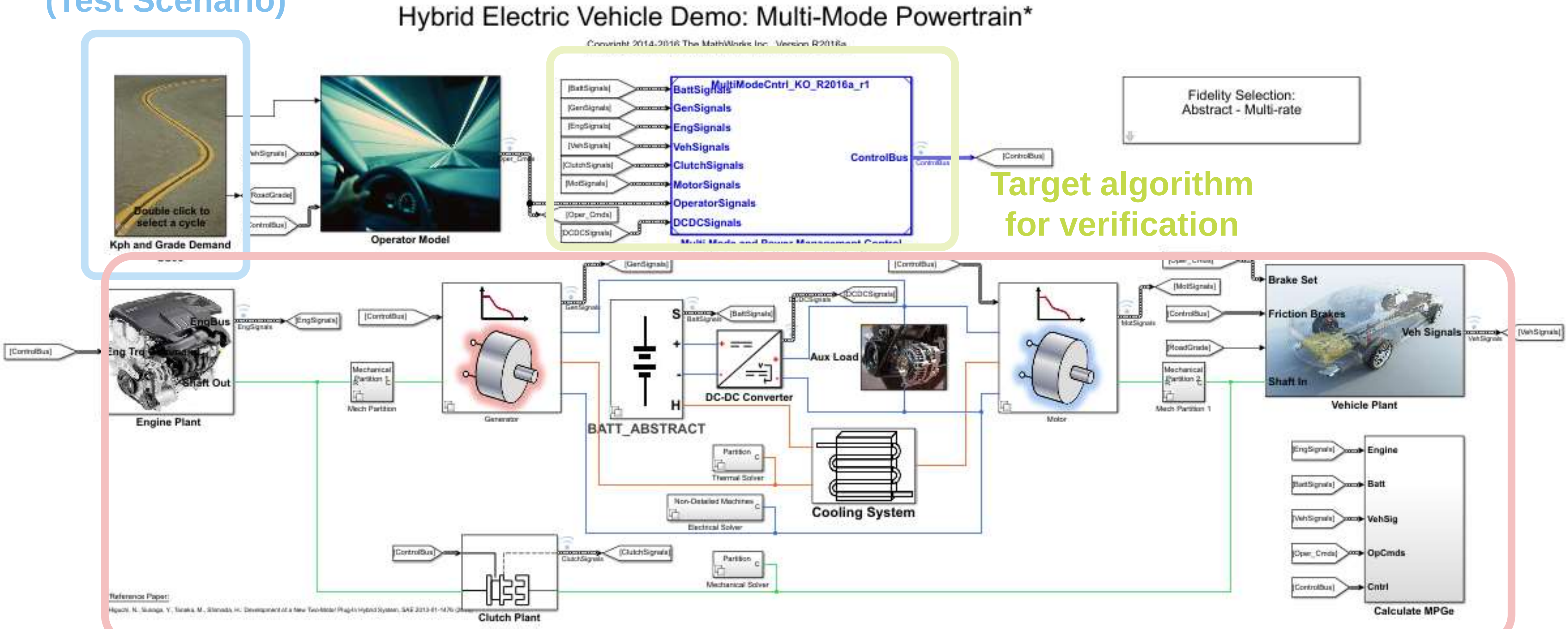
MathWorks
AUTOMOTIVE
CONFERENCE
2018

Testing Framework with Simulink Test

김종헌 부장


Requested Tactive Wheel Force & HEV Modes Vs. Engine Speed
파일(F) 편집(E) 보기(V) 삽입(I) 툴(T) 데스크탑(D) 창(W) 도움말(H)
Requested Tractive Force [N]
8000
7000
6000
5000
4000
3000
2000
1000
0
EV
SHEV
Engine / Power Split
0
20
40
60
80
100
120
140
160
Vehicle Speed [kph]
HEV_MultiMode_Optim_R2016a * - Simulink
File Edit View Display Diagram Simulation Analysis Code Tools Help
596
Accelerator
HEV_MultiMode_Optim_R2016a
Fan
Hybrid Electric Vehicle Demo: Multi-Mode Powertrain
Copyright 2014-2016 The MathWorks Inc. Version R2016a
Double click to select a cycle
Kph and Grade Demand
US06
Operator Model
MultiModeCntrl_KO_R2016a_r1
BattSignals
GenSignals
EngSignals
VehSignals
ClutchSignals
MotorSignals
OperatorSignals
DCDCSignals
ControlBus
Multi Mode and Power Management Control
EngBus
Eng Trq
Shaft Out
Engine Plant
Mech Partition
Generator
BATT_ABSTRACT
DC-DC Converter
Aux Load
Motor
Friction Brakes
Veh Signals
Shaft In
Vehicle Plant
Cooling System
Thermal Solver
Electrical Solver
Clutch Plant
Mechanical Solver
Engine
Batt
VehSig
OpCmds
Cntrl
Calculate MPGe
Running
View diagnostics
58%
T=3.300
0%
FixedStepDiscrete

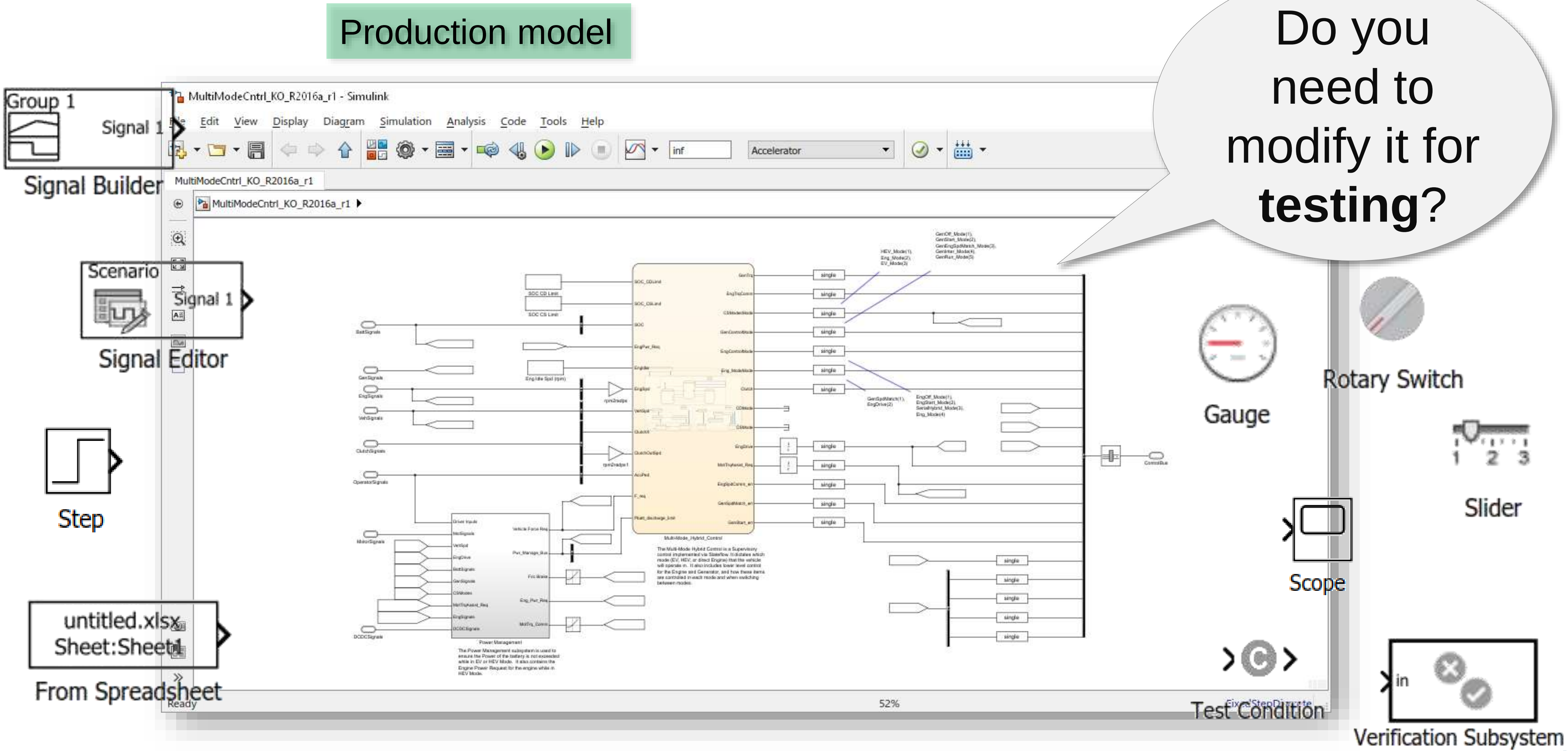
Test Harness
(Test Scenario)



Target algorithm
for verification

Test Harness (Plant model)

How to Test Your Model…?

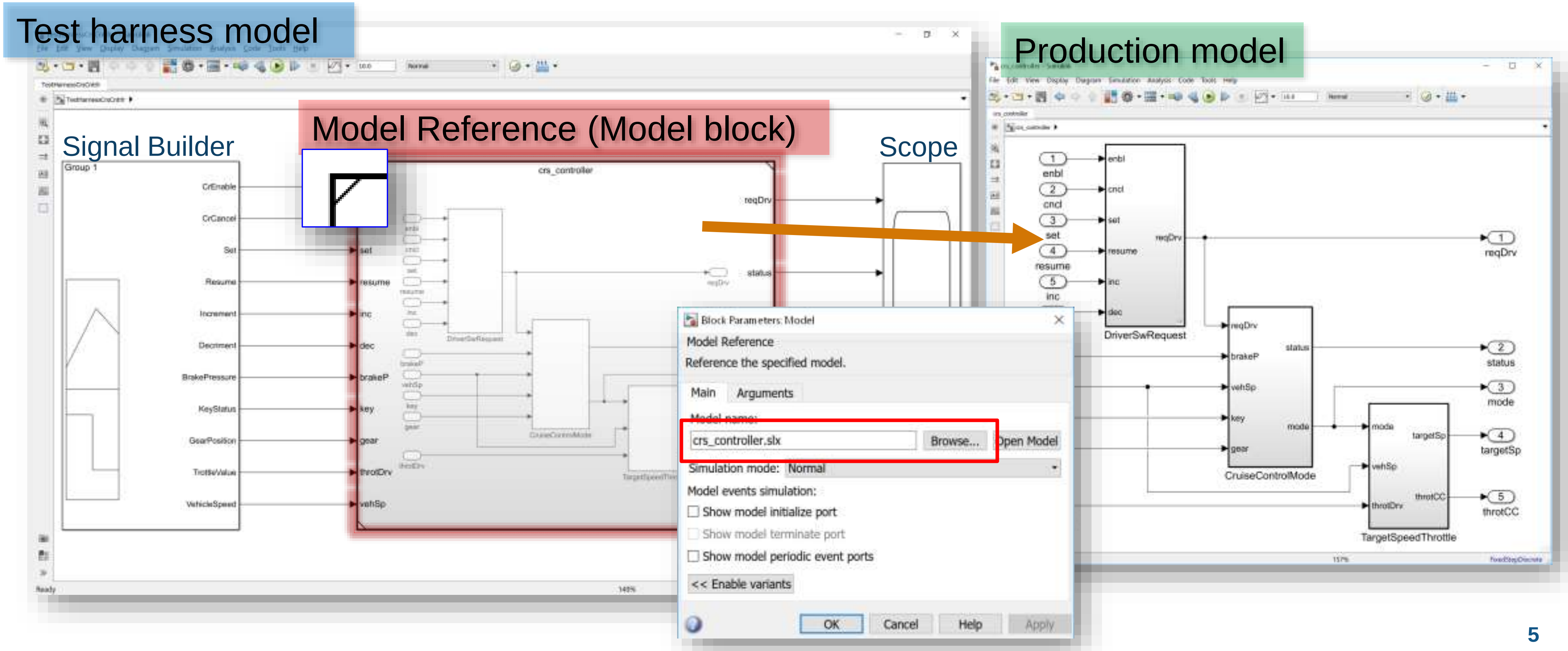


Production model

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



Test harness model

Model Reference (Model block)

Signal Builder

Scope

Production model

Block Parameters: Model

Model Reference

Reference the specified model.

Main | Arguments

Model name: crs_controller.slx

Simulation mode: Normal

Model events simulation:

□ Show model initialize port

□ Show model terminate port

□ Show model periodic event ports

Simulink Test

Why Simulink Test?

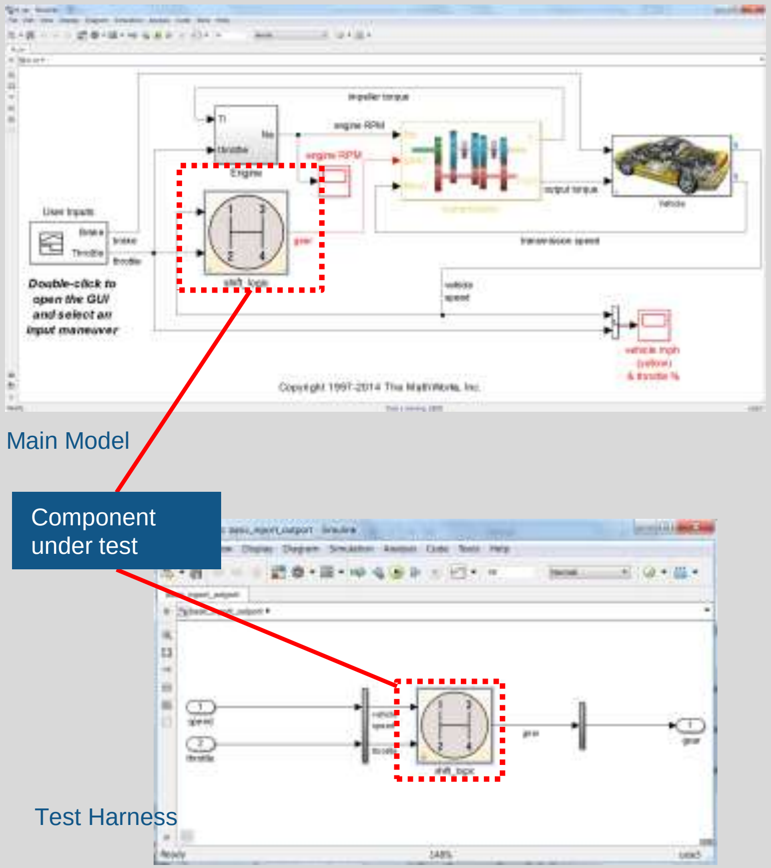
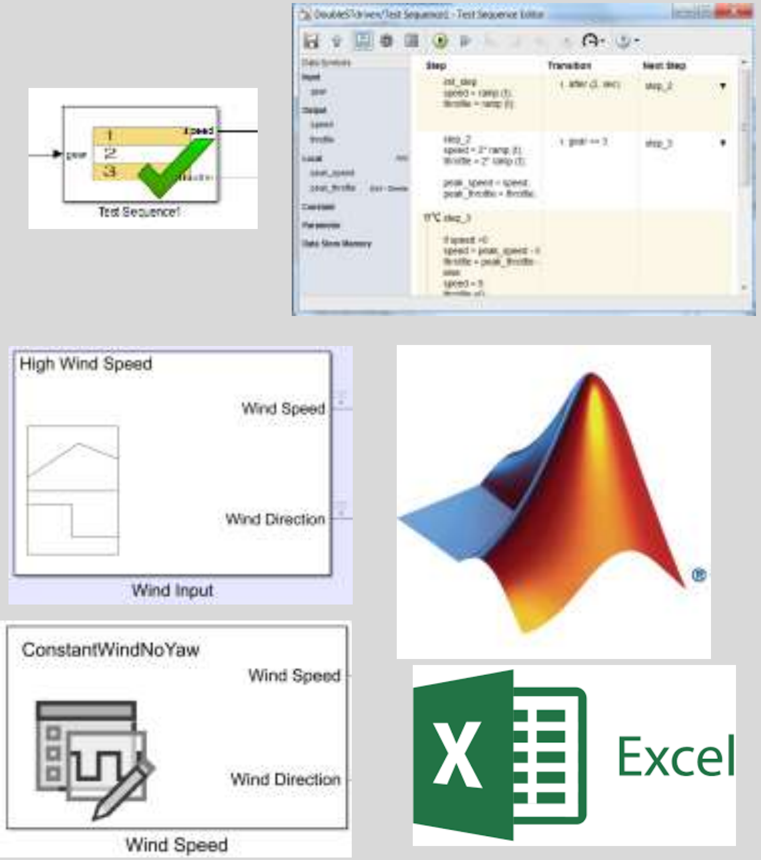
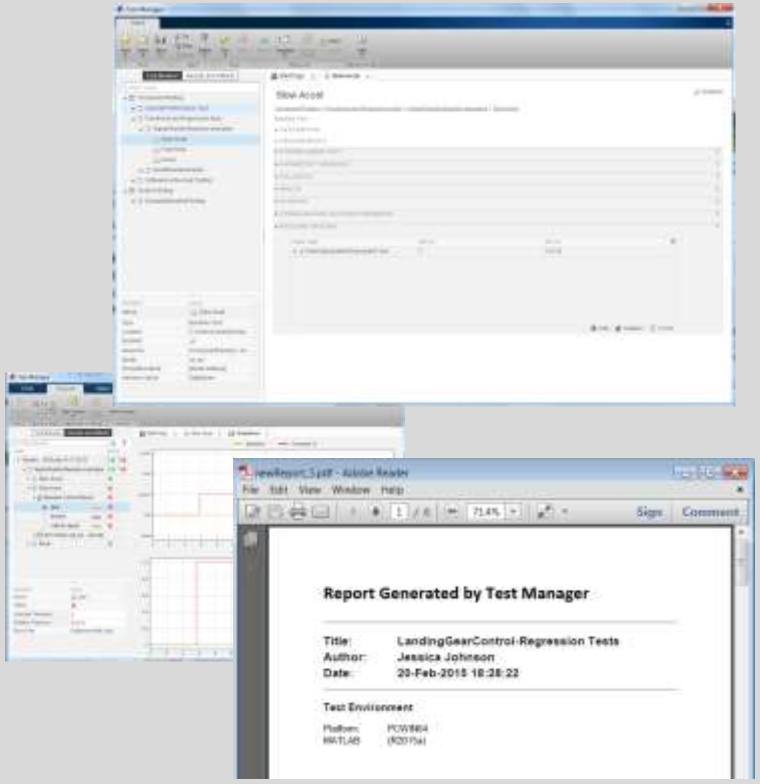
Saves you time:

- 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

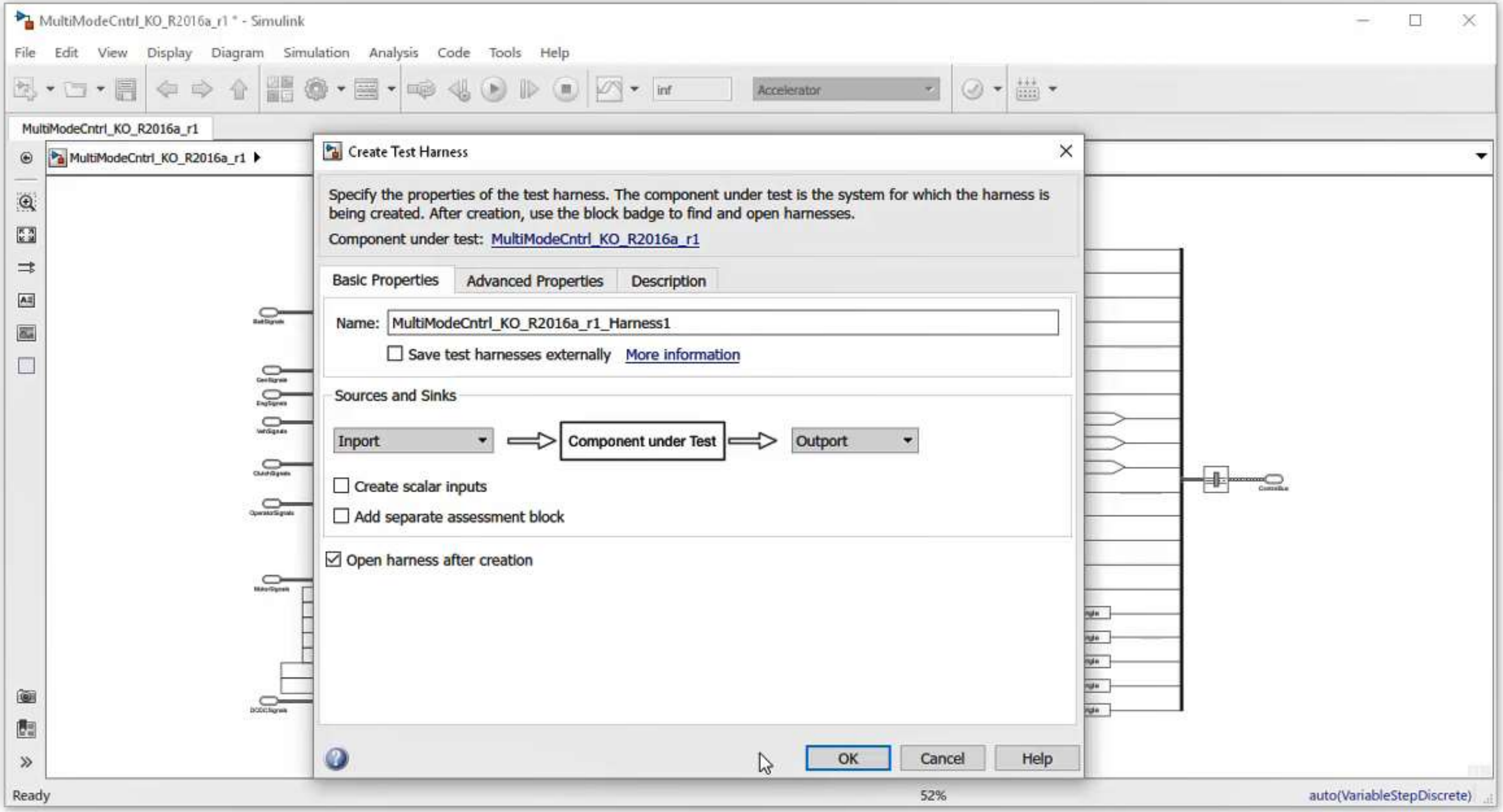
Simulink Test Overview

1. Test Harnesses	2. Test Stimulus Integration	3. Test Manager
• Synchronized, simulatable test environment	• Inputs and assessments based on logical, temporal conditions	• Author, execute, manage test cases • Review, export, report
 Double-click to open the GUI and select an input maneuver Copyright 1997-2014 The MathWorks, Inc. Main Model Component under test Test Harness	 Test Sequence1 Step Transition Next Step High Wind Speed Wind Speed Wind Direction Wind Input ConstantWindNoYaw Wind Speed Wind Direction Wind Speed Excel	 Report Generated by Test Manager Title: LandingGearControl-Regression Tests Author: Jessica Johnson Date: 20-Feb-2015 18:28:22 Test Environment

Agenda

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

Creating Test Harness



MultiModeCntrl_KO_R2016a_r1 * - Simulink
File Edit View Display Diagram Simulation Analysis Code Tools Help
inf
Accelerator
MultiModeCntrl_KO_R2016a_r1
MultiModeCntrl_KO_R2016a_r1
Ready
52%
auto(VariableStepDiscrete)

Create 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: MultiModeCntrl_KO_R2016a_r1

Basic Properties | Advanced Properties | Description

Name: MultiModeCntrl_KO_R2016a_r1_Harness1

□ Save test harnesses externally More information

Sources and Sinks

Inport → Component under Test → Outport

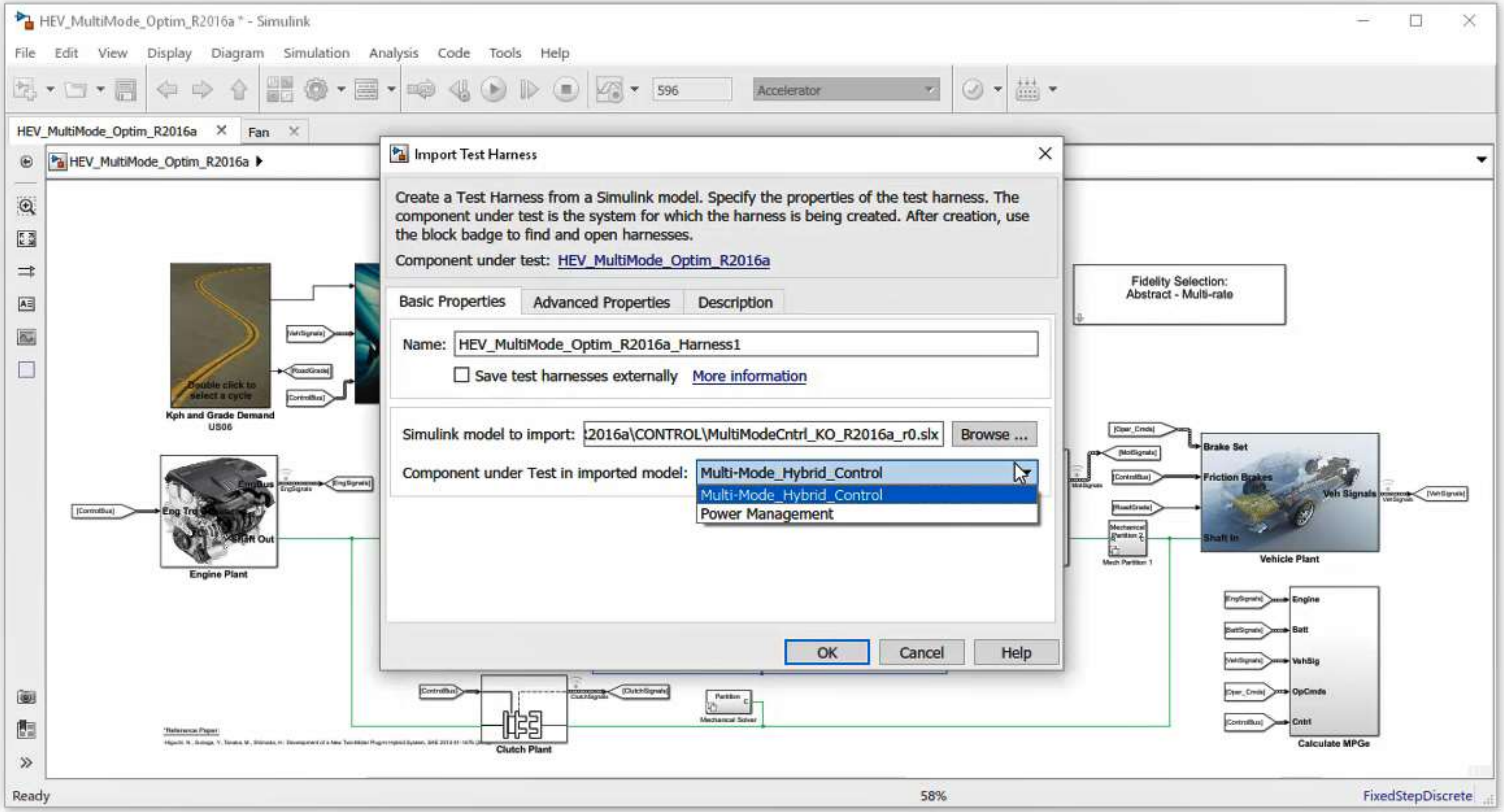
□ Create scalar inputs

□ Add separate assessment block

☑ Open harness after creation

OK Cancel Help

What if you already have a harness model....



HEV_MultiMode_Optim_R2016a * - Simulink

Import Test Harness

Create a Test Harness from a Simulink model. 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: HEV_MultiMode_Optim_R2016a

Name: HEV_MultiMode_Optim_R2016a_Harness1

□ Save test harnesses externally More information

Simulink model to import: :2016a\CONTROL\MultiModeCntrl_KO_R2016a_r0.slx

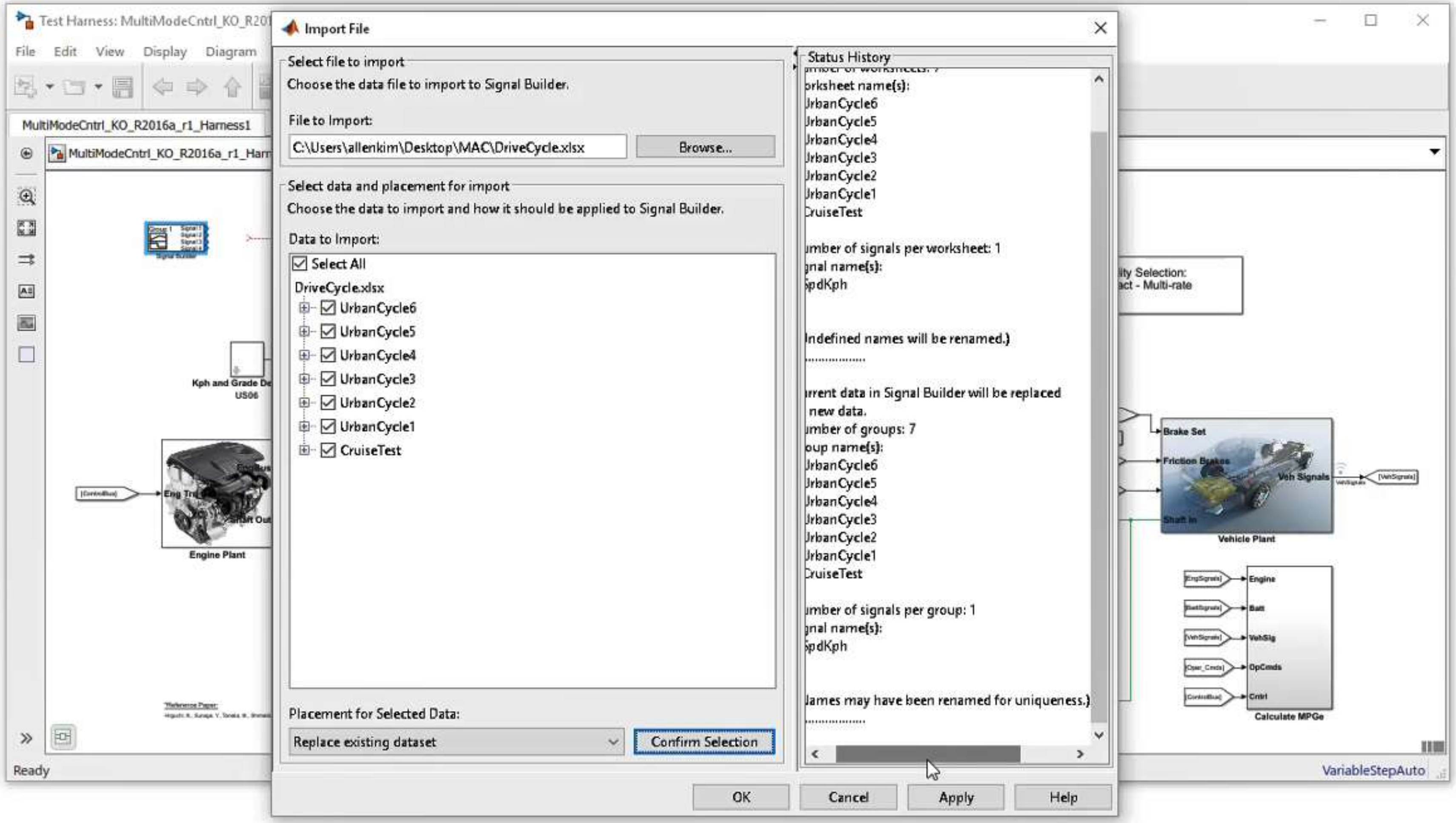
Component under Test in imported model: Multi-Mode_Hybrid_Control

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



Select file to import

Choose the data file to import to Signal Builder.

File to Import: C:\Users\allenkim\Desktop\MAC\DriveCycle.xlsx

Select data and placement for import

Choose the data to import and how it should be applied to Signal Builder.

Data to Import:

☑ Select All

DriveCycle.xlsx
- ☑ UrbanCycle6
- ☑ UrbanCycle5
- ☑ UrbanCycle4
- ☑ UrbanCycle3
- ☑ UrbanCycle2
- ☑ UrbanCycle1
- ☑ CruiseTest

Placement for Selected Data: Replace existing dataset

Status History

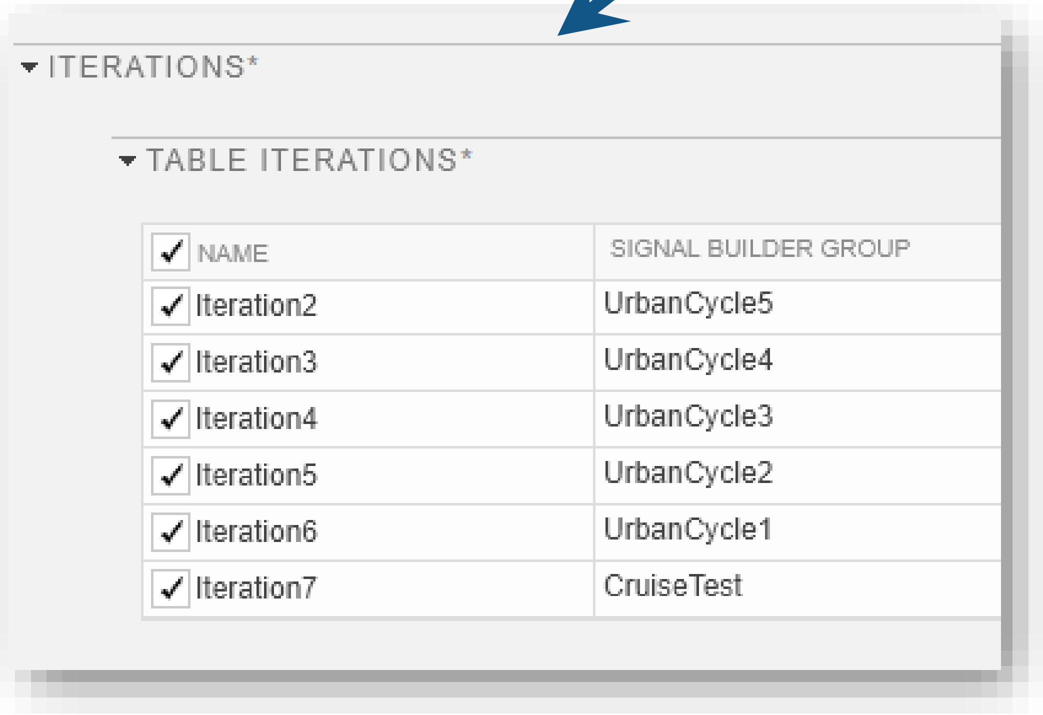
```
orksheet name(s):
JrbanCycle6
JrbanCycle5
JrbanCycle4
JrbanCycle3
JrbanCycle2
JrbanCycle1
CruiseTest

umber of signals per worksheet: 1
gnal name(s):
SpdKph


Indefined names will be renamed.)
....................

urrent data in Signal Builder will be replaced
 new data.
umber of groups: 7
oup name(s):
JrbanCycle6
JrbanCycle5
JrbanCycle4
JrbanCycle3
JrbanCycle2
JrbanCycle1
CruiseTest

umber of signals per group: 1
gnal name(s):
SpdKph


Names may have been renamed for uniqueness.)
....................
```

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...

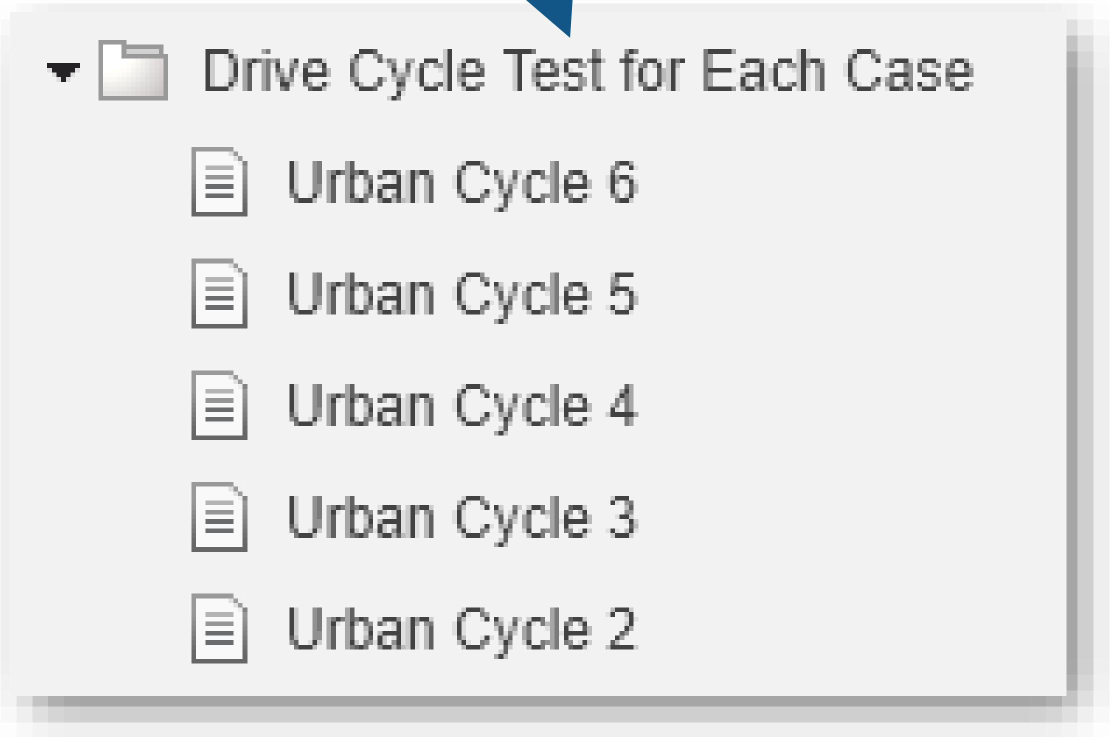
When should I use iterations vs multiple test cases?



▾ ITERATIONS*

▾ TABLE ITERATIONS*

☑ NAME	SIGNAL BUILDER GROUP
☑ Iteration2	UrbanCycle5
☑ Iteration3	UrbanCycle4
☑ Iteration4	UrbanCycle3
☑ Iteration5	UrbanCycle2
☑ Iteration6	UrbanCycle1
☑ Iteration7	CruiseTest

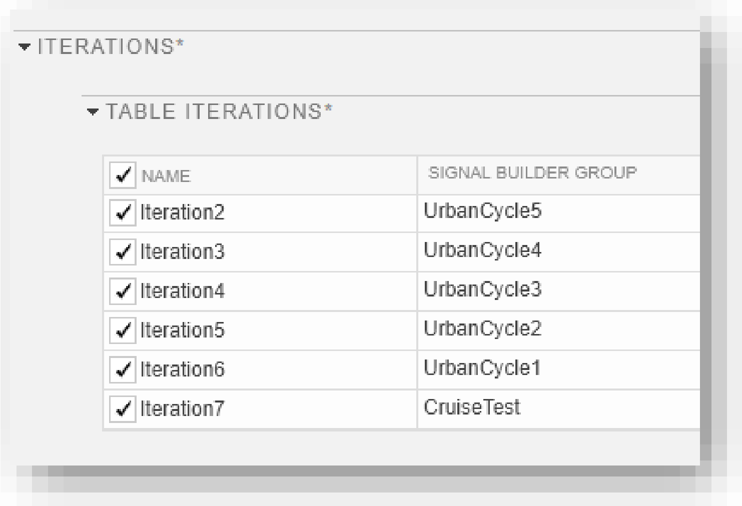


▾ Drive Cycle Test for Each Case

- Urban Cycle 6
- Urban Cycle 5
- Urban Cycle 4
- Urban Cycle 3
- Urban Cycle 2

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

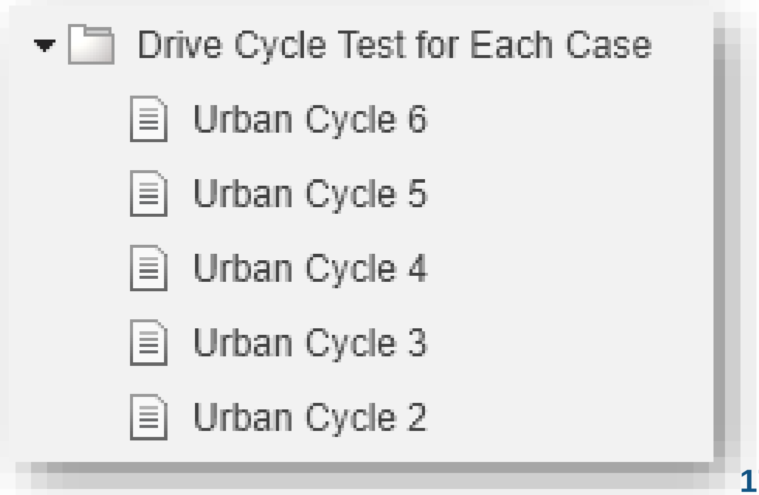


ITERATIONS*

TABLE ITERATIONS*

☑ NAME	SIGNAL BUILDER GROUP
☑ Iteration2	UrbanCycle5
☑ Iteration3	UrbanCycle4
☑ Iteration4	UrbanCycle3
☑ Iteration5	UrbanCycle2
☑ Iteration6	UrbanCycle1
☑ Iteration7	CruiseTest

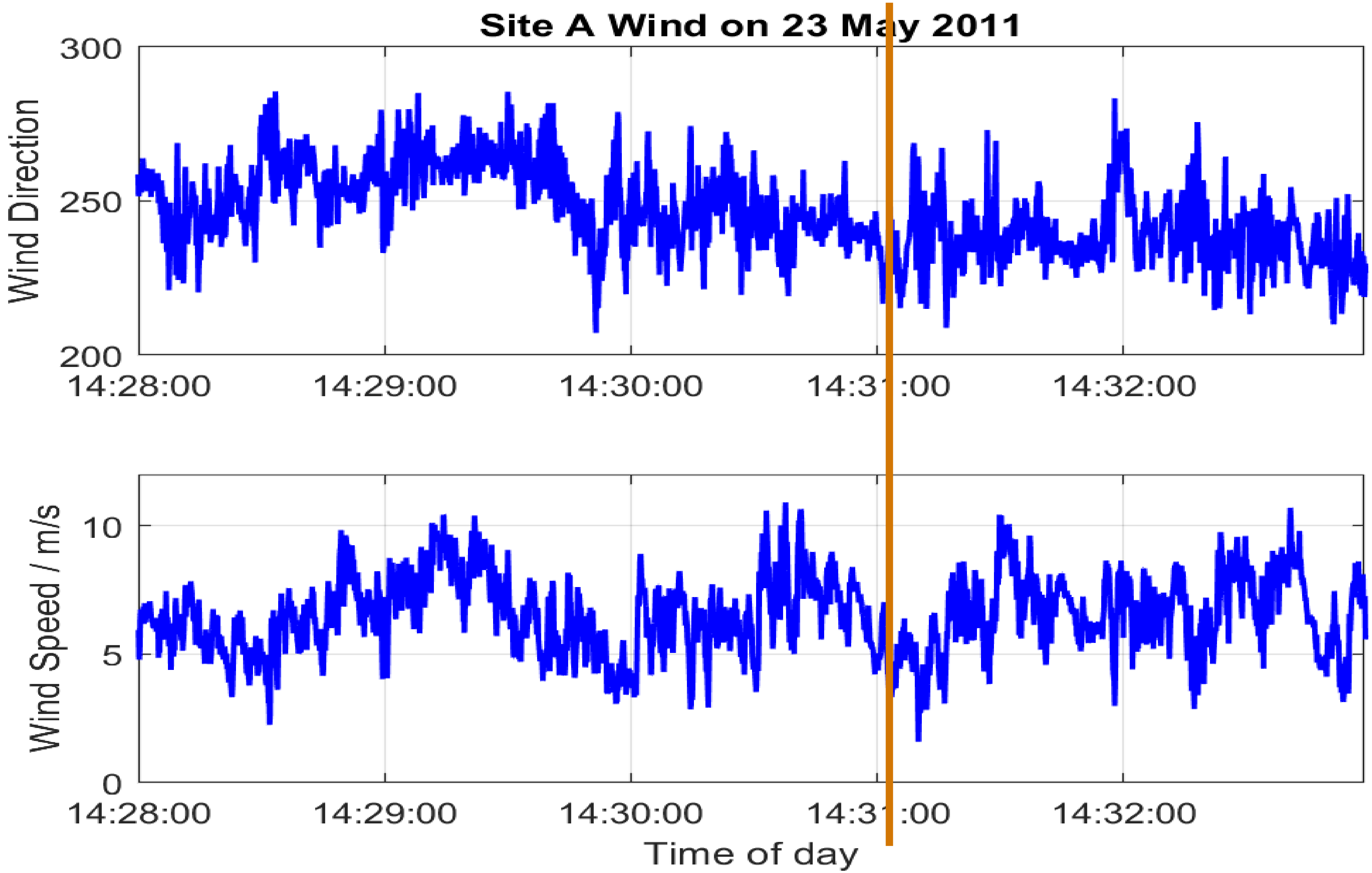
- Use separate test cases if:
  - Need independent configuration control
  - Different model/harness/test type or callbacks
  - Relate to distinct requirements
  - Distinct control of coverage



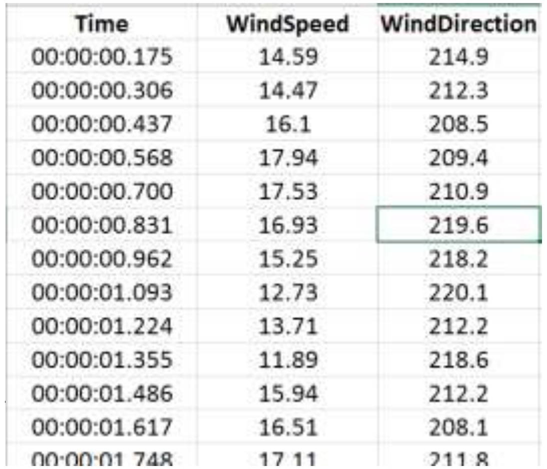
Drive Cycle Test for Each Case
- Urban Cycle 6
- Urban Cycle 5
- Urban Cycle 4
- Urban Cycle 3
- Urban Cycle 2

Example 2: Create a test case using real-world recorded data

My data


Site A Wind on 23 May 2011
Wind Direction
300
250
200
14:28:00
14:29:00
14:30:00
14:31:00
14:32:00
Wind Speed / m/s
10
5
0
14:28:00
14:29:00
14:30:00
14:31:00
14:32:00
Time of day

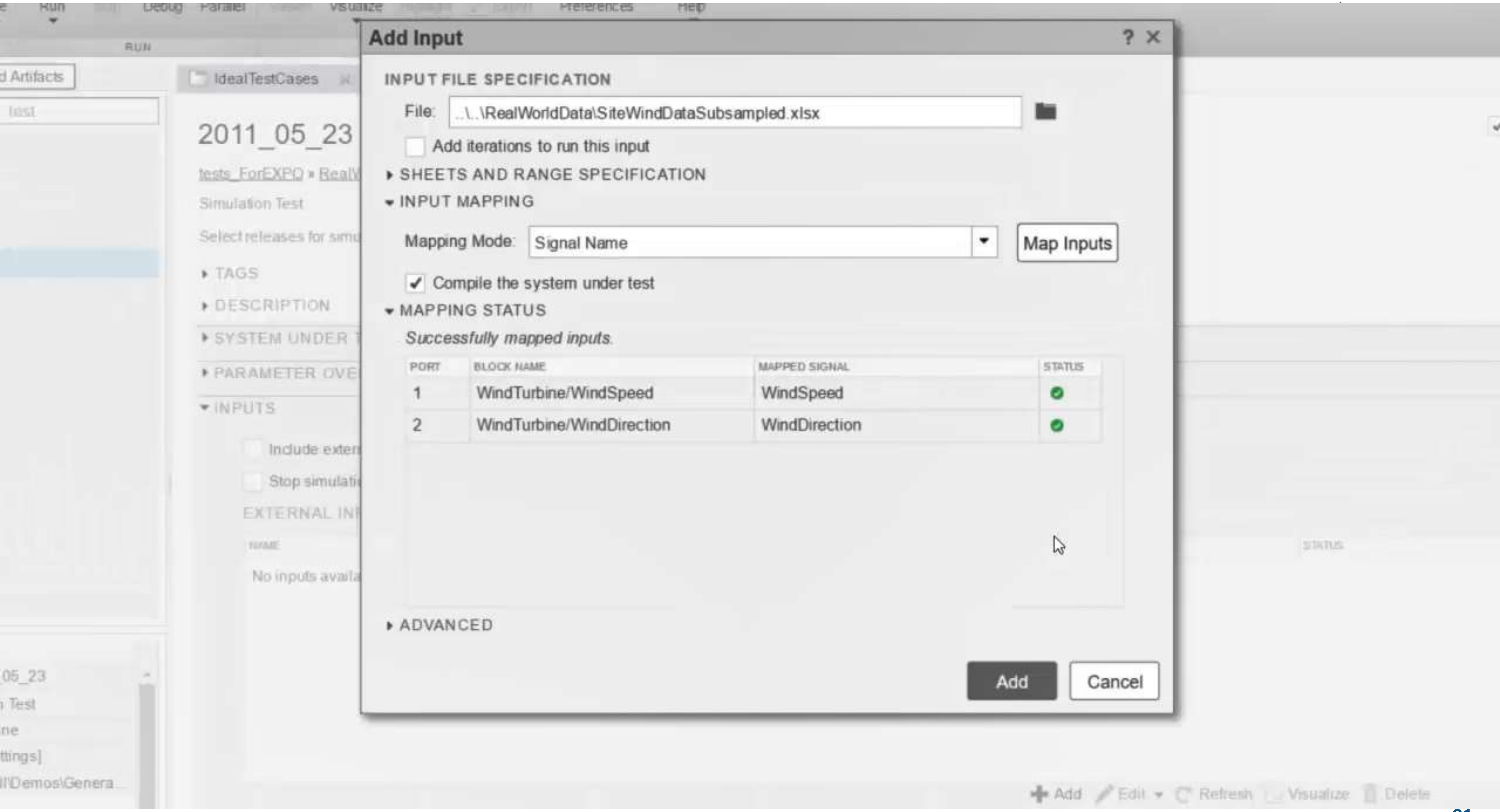
Importing time-stamped data from Excel or text files



Time	WindSpeed	WindDirection
00:00:00.175	14.59	214.9
00:00:00.306	14.47	212.3
00:00:00.437	16.1	208.5
00:00:00.568	17.94	209.4
00:00:00.700	17.53	210.9
00:00:00.831	16.93	219.6
00:00:00.962	15.25	218.2
00:00:01.093	12.73	220.1
00:00:01.224	13.71	212.2
00:00:01.355	11.89	218.6
00:00:01.486	15.94	212.2
00:00:01.617	16.51	208.1
00:00:01.748	17.11	211.8

```
% 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



Add Input

INPUT FILE SPECIFICATION

File: ..\..\RealWorldData\SiteWindDataSubsampled.xlsx

□ Add iterations to run this input

▸ SHEETS AND RANGE SPECIFICATION

▾ INPUT MAPPING

Mapping Mode: Signal Name

☑ Compile the system under test

▾ MAPPING STATUS

Successfully mapped inputs.

PORT	BLOCK NAME	MAPPED SIGNAL	STATUS
1	WindTurbine/WindSpeed	WindSpeed	●
2	WindTurbine/WindDirection	WindDirection	●

▸ ADVANCED

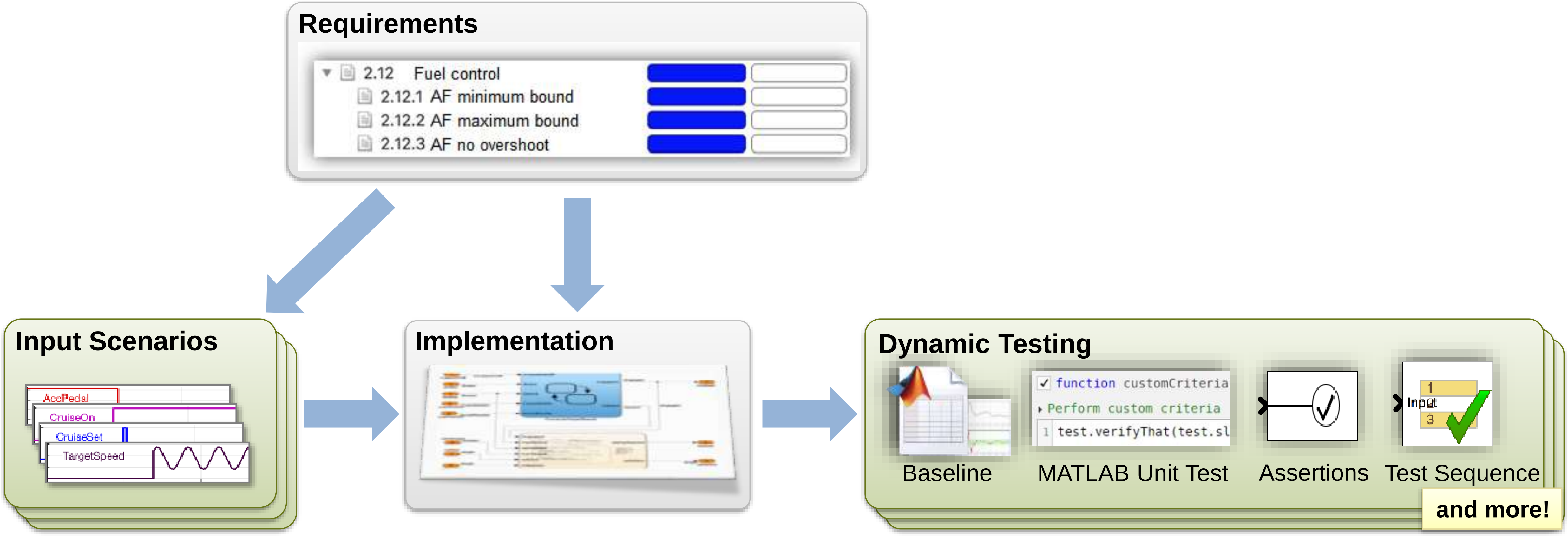
What have we done so far....

- 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

- 2.12 Fuel control
  - 2.12.1 AF minimum bound
  - 2.12.2 AF maximum bound
  - 2.12.3 AF no overshoot

Input Scenarios

Implementation

Dynamic Testing

```
✓ function customCriteria
▸ Perform custom criteria
test.verifyThat(test.sl
```

Baseline

MATLAB Unit Test

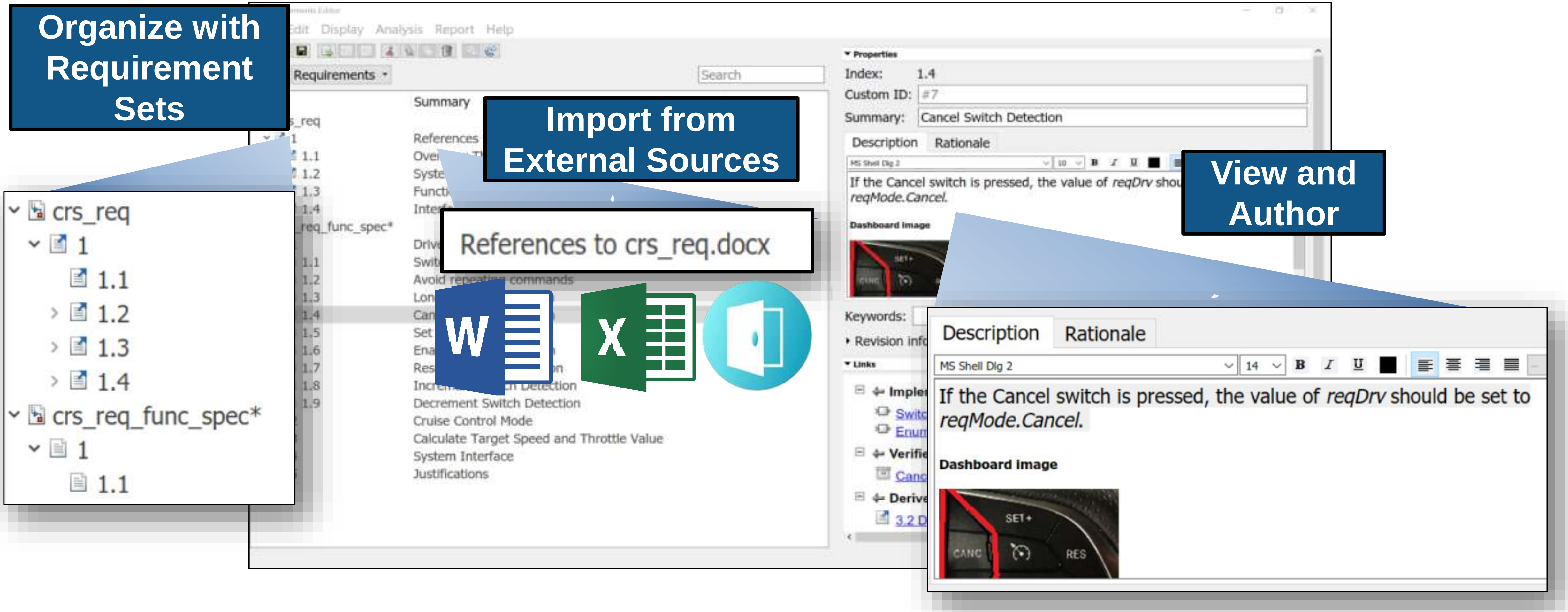
Assertions

Test Sequence

and more!

Requirements Editor in Simulink Requirements

Manage and Organize Requirements



Organize with Requirement Sets

Import from External Sources

View and Author

Index: 1.4

Custom ID: #7

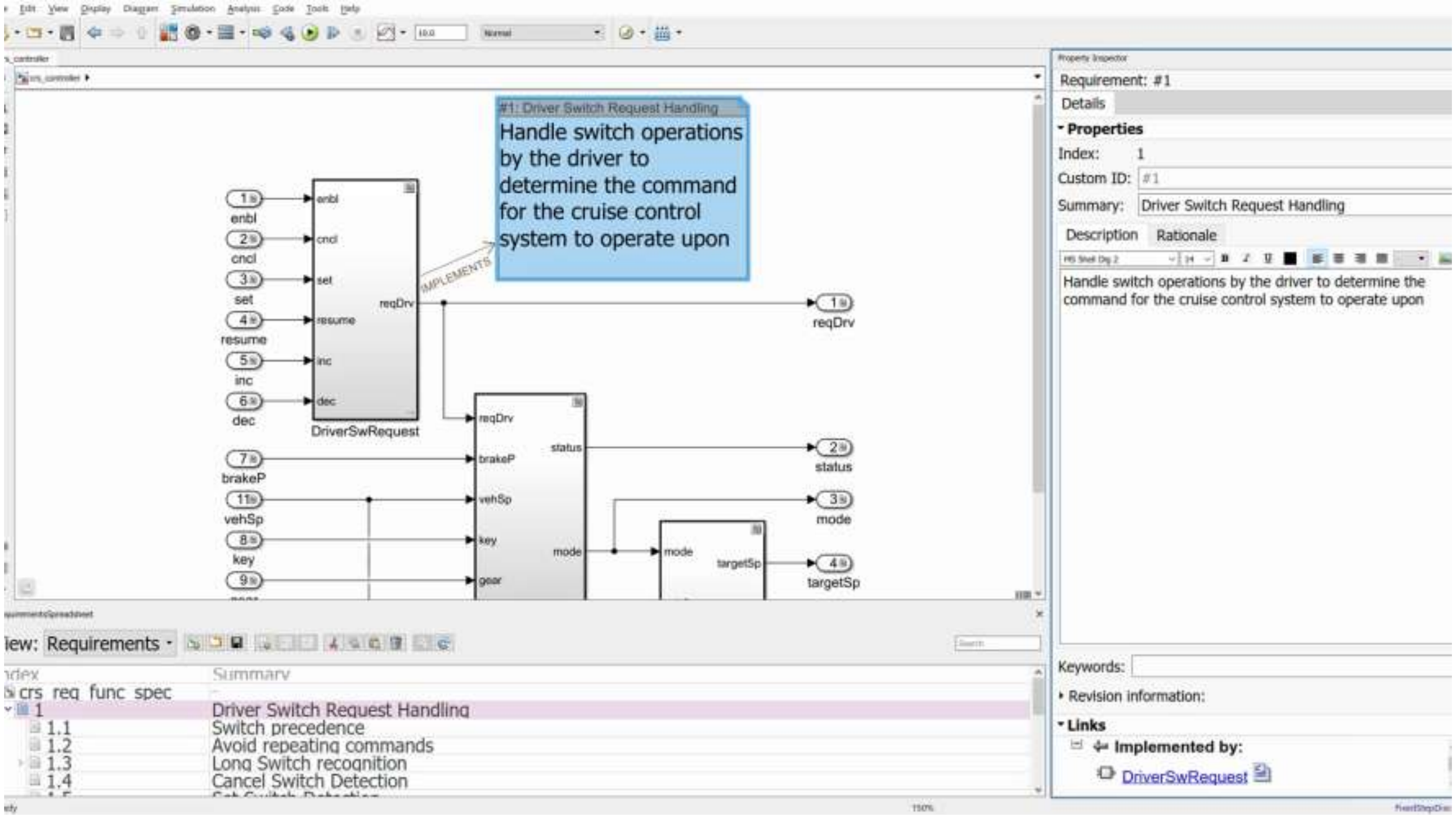
Summary: Cancel Switch Detection

If the Cancel switch is pressed, the value of *reqDrv* should be set to *reqMode.Cancel.*

Dashboard image

Requirements Perspective in Simulink Requirements

R2017b



#1: Driver Switch Request Handling

Handle switch operations by the driver to determine the command for the cruise control system to operate upon

Requirement: #1

Details

- Properties

Index: 1

Custom ID: #1

Summary: Driver Switch Request Handling

Description | Rationale

Handle switch operations by the driver to determine the command for the cruise control system to operate upon

Keywords:

- Revision information:

- Links

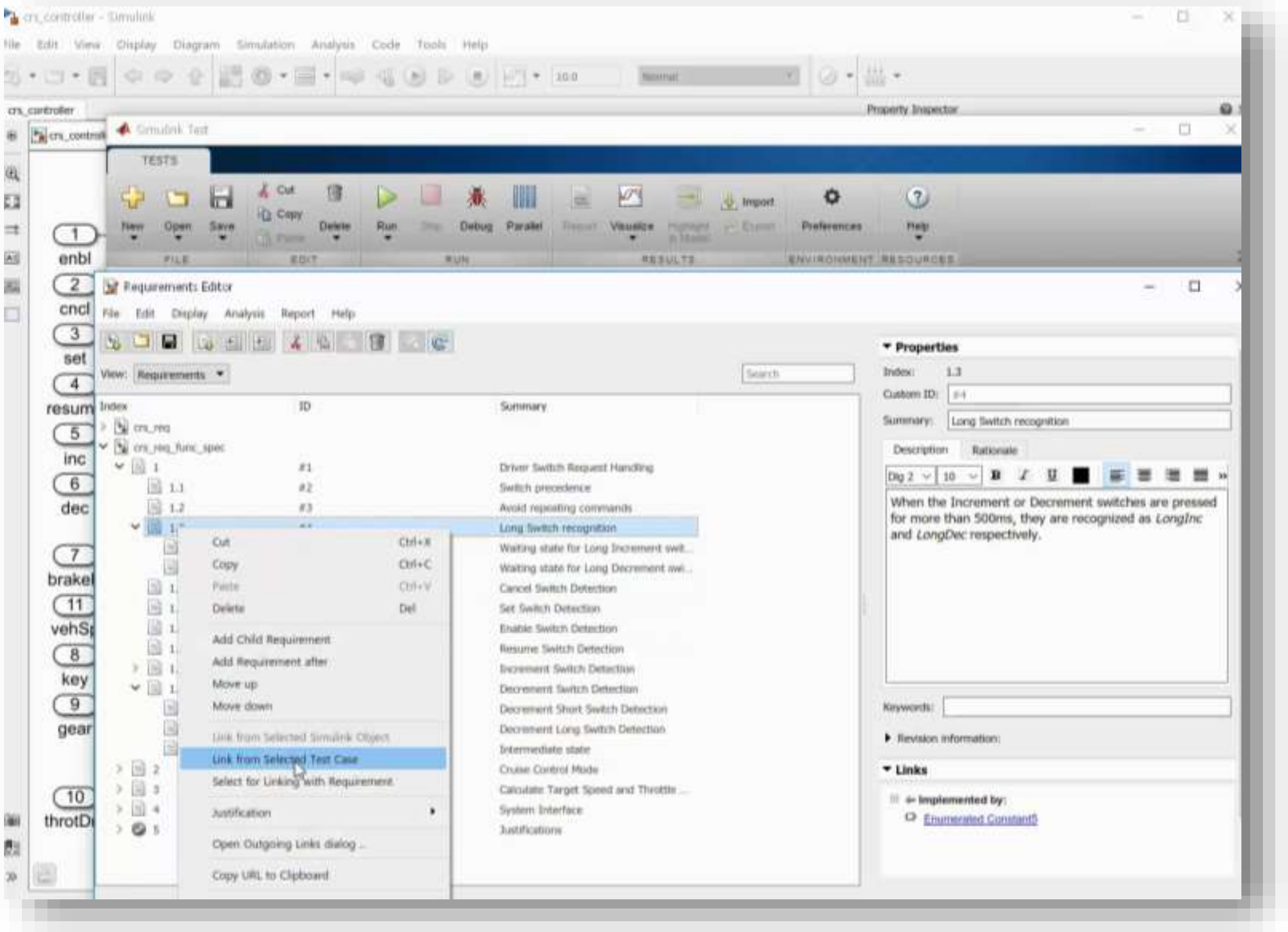
Implemented by:

DriverSwRequest

iew: Requirements

ndex	Summary
crs_req_func_spec	–
1	Driver Switch Request Handling
1.1	Switch precedence
1.2	Avoid repeating commands
1.3	Long Switch recognition
1.4	Cancel Switch Detection

Track Implementation and Verification



View: Requirements

Index	ID	Summary
crs_req		
crs_req_func_spec		
1	#1	Driver Switch Request Handling
1.1	#2	Switch precedence
1.2	#3	Avoid repeating commands
1.		Long Switch recognition
		Waiting state for Long Increment swit...
		Waiting state for Long Decrement swi...
1.		Cancel Switch Detection
1.		Set Switch Detection
1.		Enable Switch Detection
1.		Resume Switch Detection
1.		Increment Switch Detection
1.		Decrement Switch Detection
		Decrement Short Switch Detection
		Decrement Long Switch Detection
		Intermediate state
2		Cruise Control Mode
3		Calculate Target Speed and Throttle ...
4		System Interface
5		Justifications

Properties

Index: 1.3

Custom ID: #4

Summary: Long Switch recognition

When the Increment or Decrement switches are pressed for more than 500ms, they are recognized as *LongInc* and *LongDec* respectively.

Keywords:

Revision information:

Links

Implemented by:

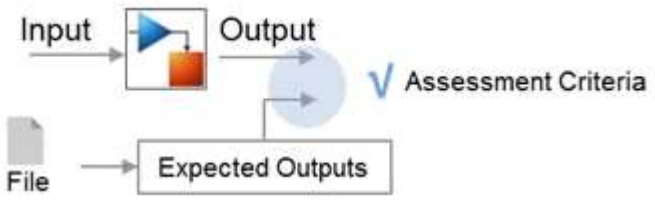
Enumerated Constant5

Example 1:
Baseline test

Test types in Test Manager

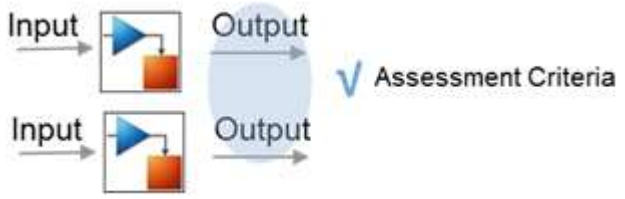
- Baseline Test

Ex) Regression test



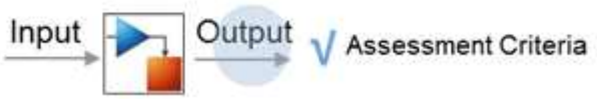
- Equivalence Test

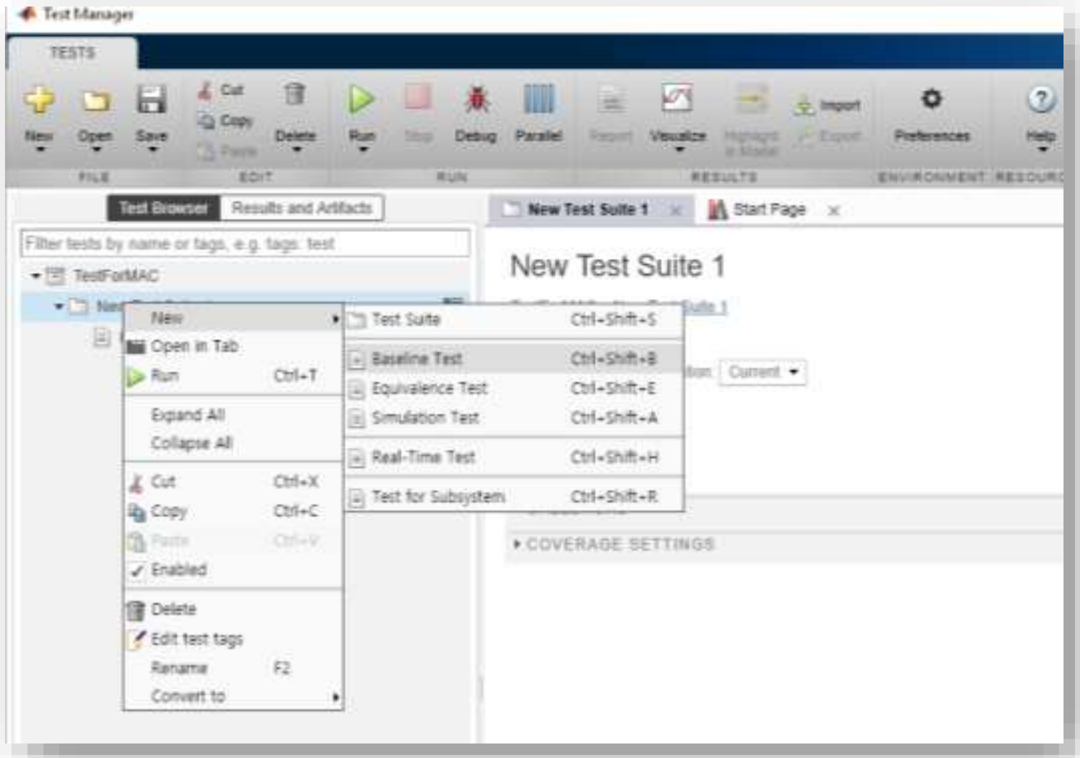
Ex) Back-to-Back test like SIL, PIL



- Simulation Test

Ex) Verifying algorithm with logical criteria



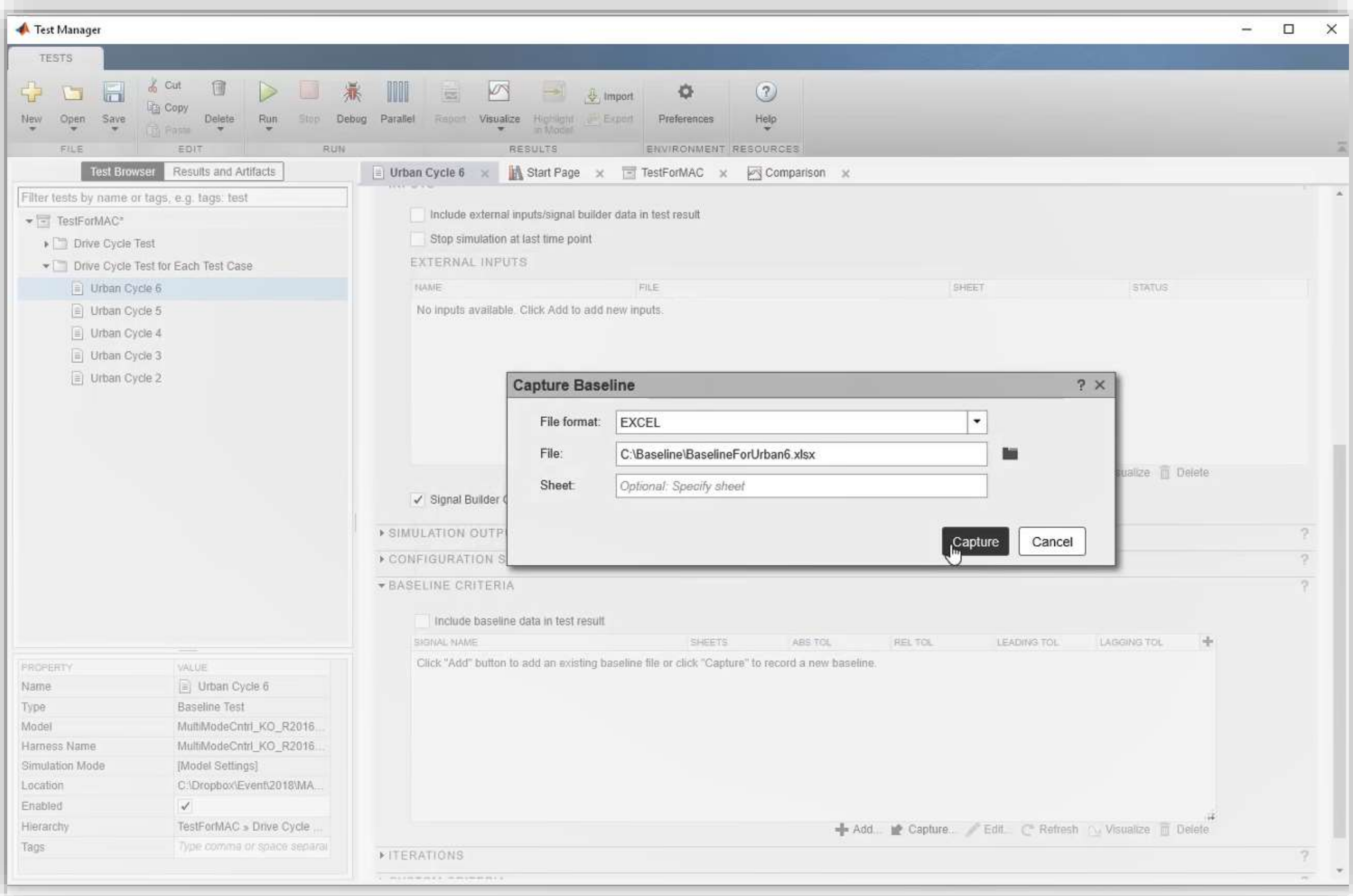


- **Challenges**
  - Not easy to predict expected result
  - Hard to make time-series input data

- **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



Filter tests by name or tags, e.g. tags: test

- TestForMAC*
  - Drive Cycle Test
  - Drive Cycle Test for Each Test Case
    - Urban Cycle 6
    - Urban Cycle 5
    - Urban Cycle 4
    - Urban Cycle 3
    - Urban Cycle 2

PROPERTY	VALUE
Name	Urban Cycle 6
Type	Baseline Test
Model	MultiModeCntrl_KO_R2016...
Harness Name	MultiModeCntrl_KO_R2016...
Simulation Mode	[Model Settings]
Location	C:\Dropbox\Event\2018\MA...
Enabled	✓
Hierarchy	TestForMAC » Drive Cycle ...
Tags	Type comma or space separat

□ Include external inputs/signal builder data in test result

□ Stop simulation at last time point

EXTERNAL INPUTS

NAME	FILE	SHEET	STATUS

No inputs available. Click Add to add new inputs.

☑ Signal Builder

▸ SIMULATION OUTPUT

▸ CONFIGURATION S

▾ BASELINE CRITERIA

□ Include baseline data in test result

SIGNAL NAME	SHEETS	ABS TOL	REL TOL	LEADING TOL	LAGGING TOL

Click "Add" button to add an existing baseline file or click "Capture" to record a new baseline.

▸ ITERATIONS

Capture Baseline

File format: EXCEL

File: C:\Baseline\BaselineForUrban6.xlsx

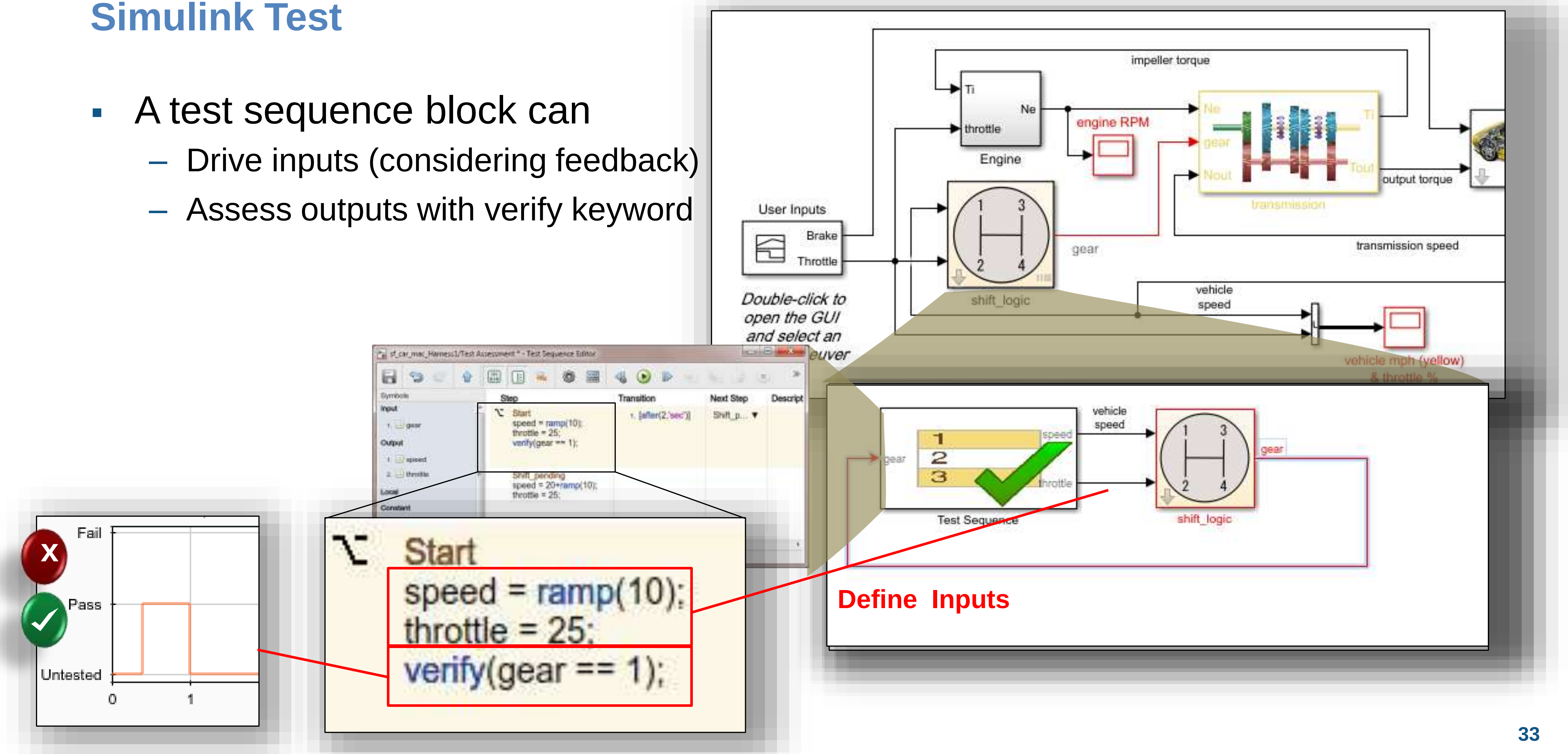
Sheet: Optional: Specify sheet

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



Double-click to open the GUI and select an maneuver

sf_car_mac_Harness1/Test Assessment * - Test Sequence Editor

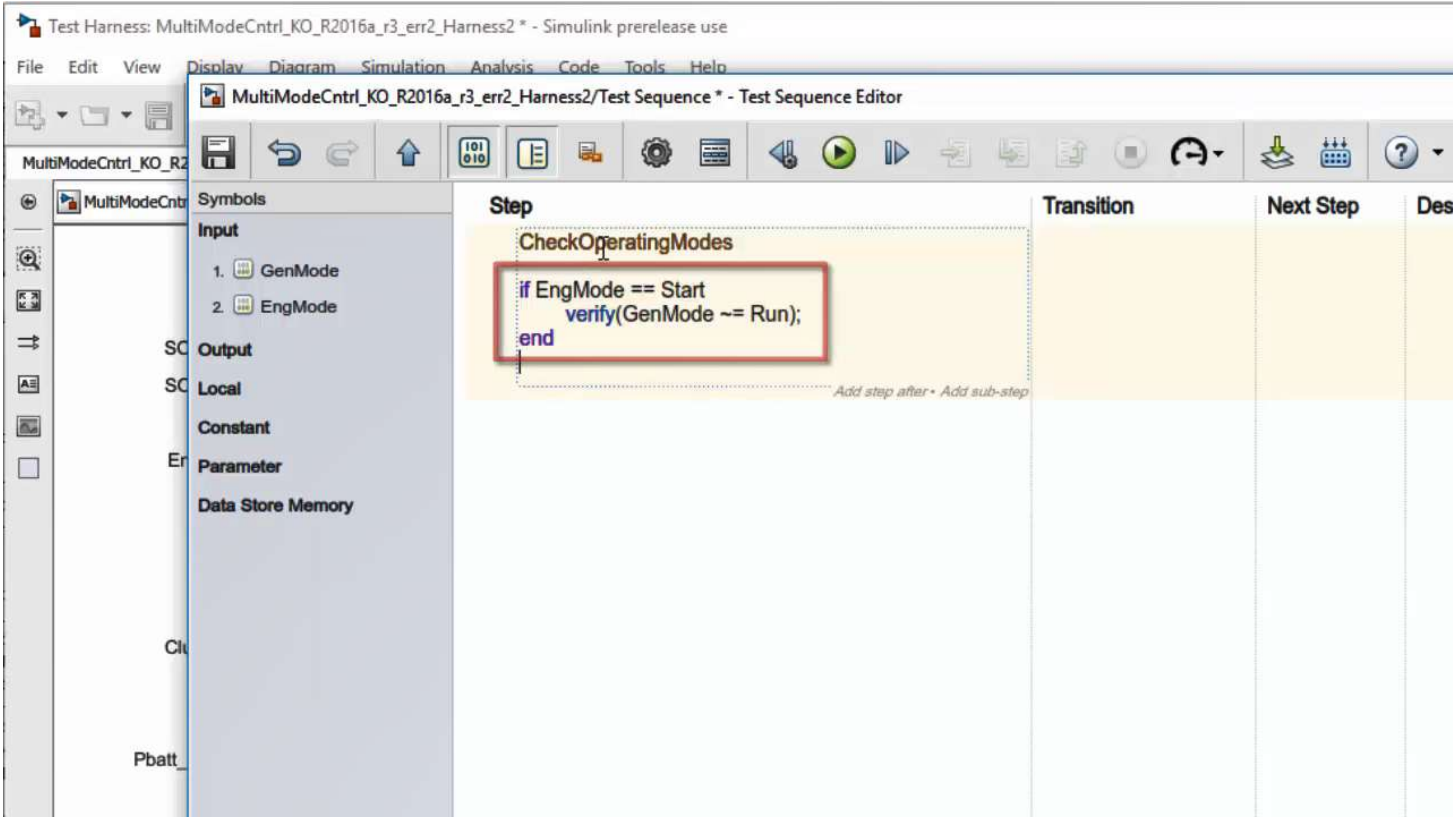
Step	Transition	Next Step	Descript

Start
speed = ramp(10);
throttle = 25;
verify(gear == 1);
	1. [after(2,'sec')]	Shift_p... ▼	

Shift_pending
speed = 20+ramp(10);
throttle = 25;			

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

Define Inputs



Test Harness: MultiModeCntrl_KO_R2016a_r3_err2_Harness2 * - Simulink prerelease use

MultiModeCntrl_KO_R2016a_r3_err2_Harness2/Test Sequence * - Test Sequence Editor

CheckOperatingModes

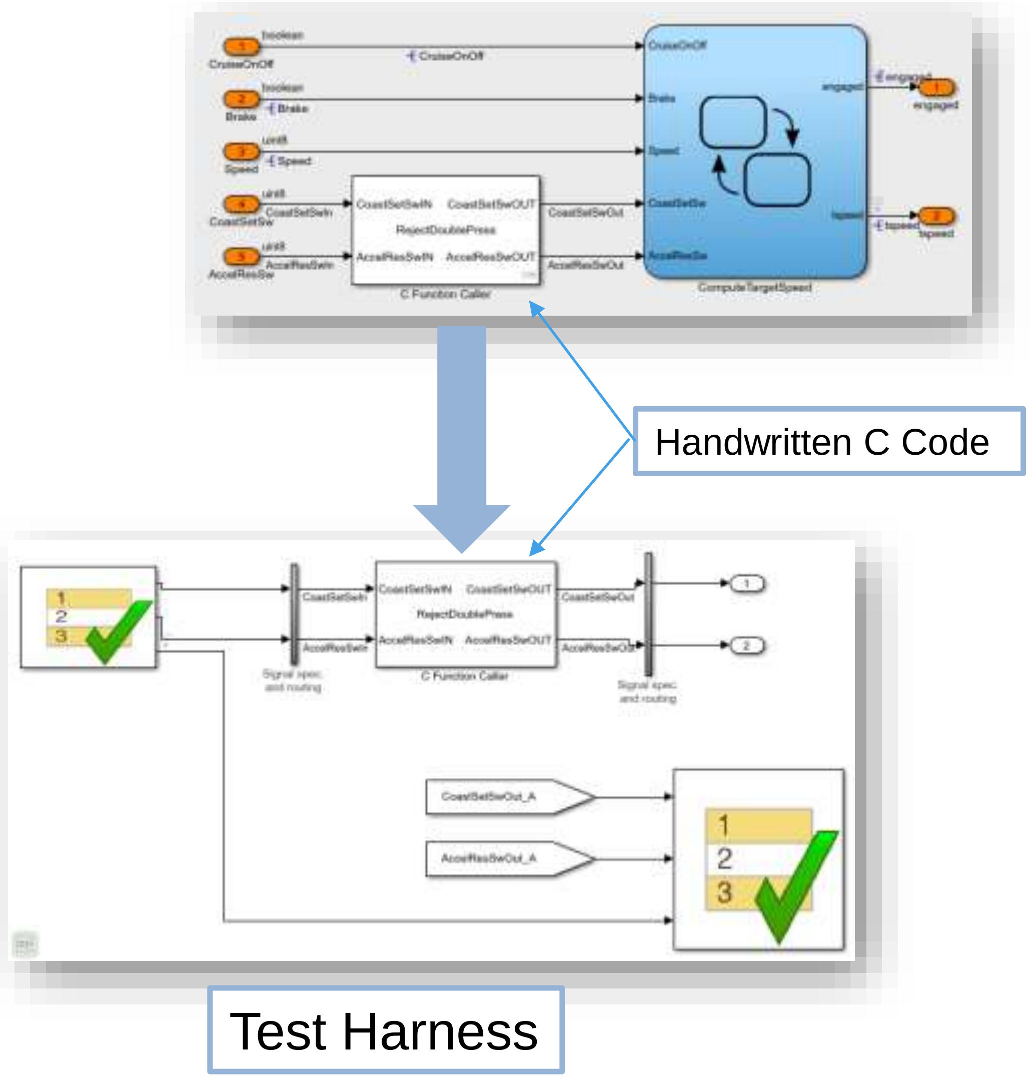
```
if EngMode == Start
        verify(GenMode ~= Run);
end
```

C Caller Block Support

R2018b

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



Handwritten C Code

Test Harness

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

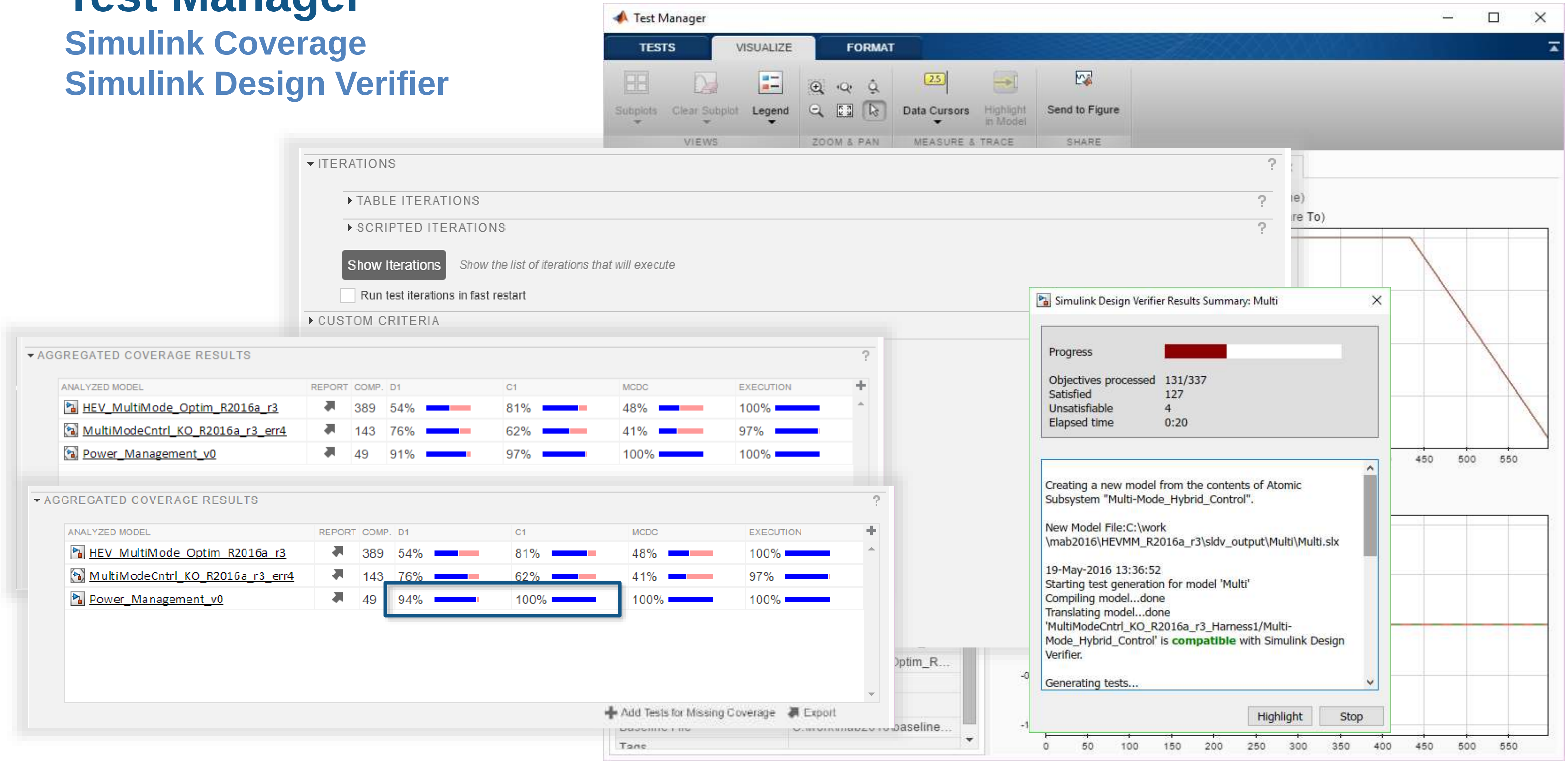
Agenda

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

Test Manager

Simulink Coverage

Simulink Design Verifier



ITERATIONS

TABLE ITERATIONS

SCRIPTED ITERATIONS

Show Iterations *Show the list of iterations that will execute*

□ Run test iterations in fast restart

CUSTOM CRITERIA

AGGREGATED COVERAGE RESULTS

ANALYZED MODEL	REPORT	COMP.	D1	C1	MCDC	EXECUTION
HEV_MultiMode_Optim_R2016a_r3		389	54%	81%	48%	100%
MultiModeCntrl_KO_R2016a_r3_err4		143	76%	62%	41%	97%
Power_Management_v0		49	91%	97%	100%	100%

AGGREGATED COVERAGE RESULTS

ANALYZED MODEL	REPORT	COMP.	D1	C1	MCDC	EXECUTION
HEV_MultiMode_Optim_R2016a_r3		389	54%	81%	48%	100%
MultiModeCntrl_KO_R2016a_r3_err4		143	76%	62%	41%	97%
Power_Management_v0		49	94%	100%	100%	100%

Add Tests for Missing Coverage Export

Simulink Design Verifier Results Summary: Multi

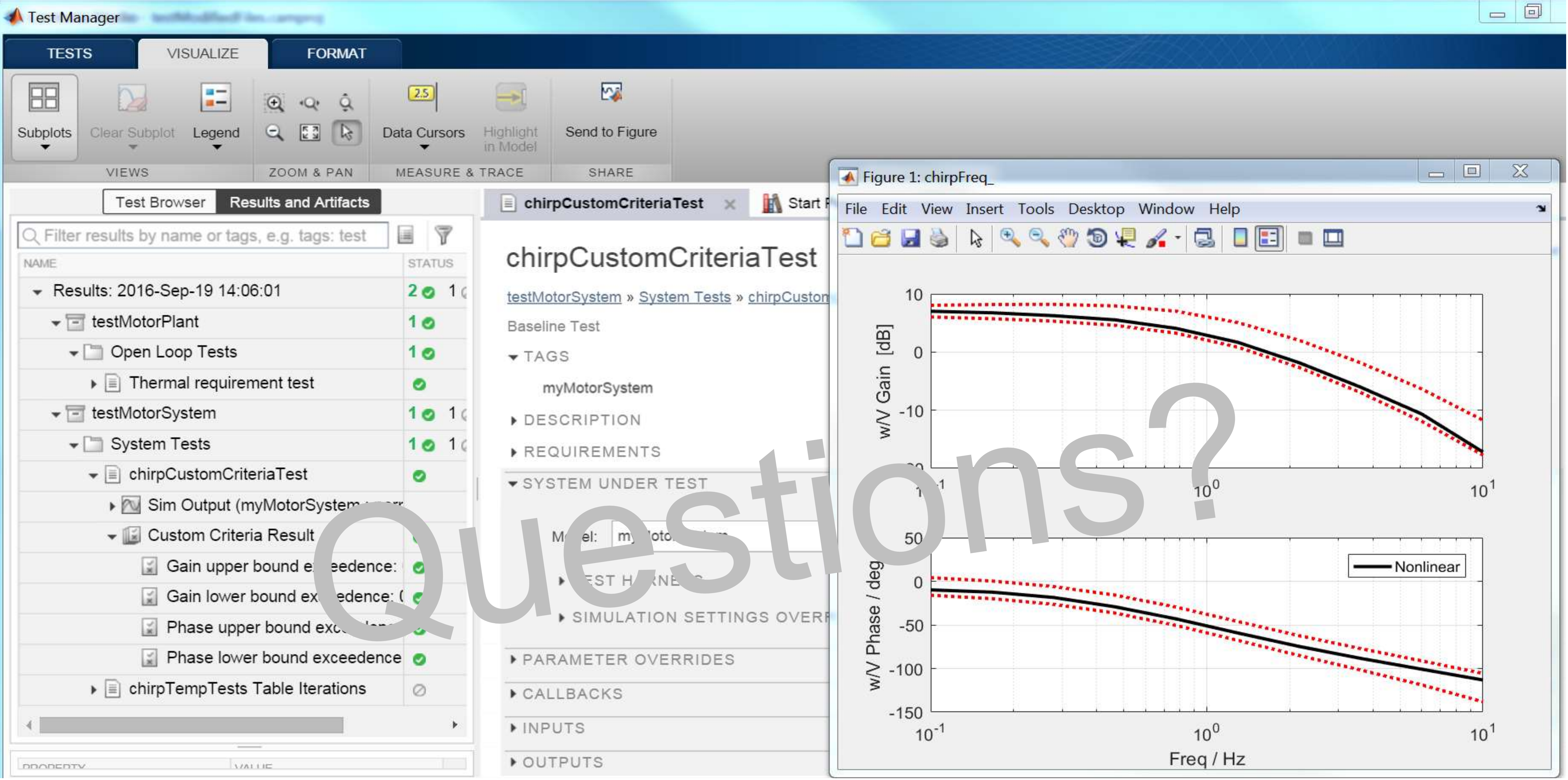
Progress	
Objectives processed	131/337
Satisfied	127
Unsatisfiable	4
Elapsed time	0:20

```
Creating a new model from the contents of Atomic
Subsystem "Multi-Mode_Hybrid_Control".

New Model File:C:\work
\mab2016\HEVMM_R2016a_r3\sldv_output\Multi\Multi.slx

19-May-2016 13:36:52
Starting test generation for model 'Multi'
Compiling model...done
Translating model...done
'MultiModeCntrl_KO_R2016a_r3_Harness1/Multi-
Mode_Hybrid_Control' is compatible with Simulink Design
Verifier.

Generating tests...
```

Highlight Stop

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



Results: 2016-Sep-19 14:06:01 — 2 ✓ 1

- testMotorPlant — 1 ✓
  - Open Loop Tests — 1 ✓
    - Thermal requirement test — ✓
- testMotorSystem — 1 ✓ 1
  - System Tests — 1 ✓ 1
    - chirpCustomCriteriaTest — ✓
      - Sim Output (myMotorSystem
      - Custom Criteria Result
        - Gain upper bound exceedence: — ✓
        - Gain lower bound exceedence: (— ✓
        - Phase upper bound exceedence — ✓
        - Phase lower bound exceedence — ✓
    - chirpTempTests Table Iterations — ⊘

PROPERTY | VALUE

chirpCustomCriteriaTest

testMotorSystem » System Tests » chirpCustom

Baseline Test

TAGS

myMotorSystem

DESCRIPTION

REQUIREMENTS

SYSTEM UNDER TEST

Model: myMotorSystem

TEST HARNESS

SIMULATION SETTINGS OVERR

PARAMETER OVERRIDES

CALLBACKS

INPUTS

OUTPUTS

Questions?