Why Test?

Does the subsystem/system meet the design requirements?
The model worked last week… does it still?
The model / algorithm has been modified… is it still working ok?
Do these legacy models / libraries work in this new application?
Does it work real-time, integrated with hardware?
Does running the generated embedded code match running the model?
Testing to date…

- Create harness models
  - to exercise model references or subsystems

- Write MATLAB code
  - to run the tests

- Write more MATLAB code
  - to verify the test results

- Write some more MATLAB code / using Report Generator
  - to report on the test results

- Creating custom GUIs to manage running of tests
### Simulink Test Overview

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

**Main Model**

**Component under test**

**Test Harness**
Creating Test Harnesses
But what if…

- The component is in a library
- The reset should be relative to the time constant of the filter
- We want to verify a requirement that: filter output shall equal resetValue when reset is true
Release Notes – Test Harnesses

• Test Harnesses for Libraries R2016a
• External Test Harnesses R2016a
• External Test Harnesses with Requirements Linking R2016b
Test Sequence / Test Assessment block

Test Sequence
Release Notes – Test Sequence

- description field  R2016a
- tab complete & syntax highlighting  R2016a
- port reordering  R2016a
- support messages  R2016a
- “verify” statements  R2016a
Test Manager
Simulation Testing

- Requirement Based (Simulation) Testing
  - Does my design comply with my requirements?

- Equivalence Testing
  - Do these models match? Does generated code match this model?

- Regression (Baseline) Testing
  - Have I broken anything with the change I’ve just made?
Create a baseline test
Create a test for multiple parameter values and verify response against a custom criteria
Test Iterations

• Define by table or script
• Combine with Parallel Computing Toolbox &/or fast restart as appropriate
• Run via UI or programmatically
• Easy to re-run selected iterations
Reporting
Release Notes – Test Manager

- Parallel Computing Toolbox integration  R2016a
- Test iterations (parameter sweeps)  R2016a
- Simulink Real-Time integration  R2016a
- Tag test cases  R2016b
- Custom test criteria  R2016b
- MATLAB Unit Test integration  R2016b
Extend requirements-based tests to achieve full coverage

(Simulink Test + Simulink Design Verifier)
A file is modified;

What test(s) do I need to run?

(Simulink Test + Simulink Projects)
Related talks...

- Simulink Projects… Gavin Walker, masterclass, after lunch
- Connecting to hardware…. Nicolas Gautier, AT2, after lunch
- Verification of generated code… Richard Anderson, AT2, last session
Related demo stations...

Verifying Models and Code Through Automated Testing and Analysis

Design models in Simulink must be supported by appropriate levels of testing: effective methods that identify defects as early as possible. Efficient testing is just as important in speeding up development as having capable design tools. Come and discover the range of verification and validation techniques that can be applied to models and code, centered on the framework for simulation-based testing provided by Simulink Test™.

System Modelling, Project Management, and Simulation

System-level modelling and simulation requires flexibility in modelling style and testing frameworks. Through the example of a system-level aircraft model, discover how the latest capabilities of Simulink are used to ease modelling, testing, and project management tasks.

Technology Focus: System-level modelling, simulation
Simulink Test... makes testing easier...

Flexible
• Ease authoring different types of test
• Desktop or real-time

Scalable
• Hierarchical
• Integration with Parallel Computing Toolbox

Simple & Efficient
• Less time writing infrastructure code
• Integration with other V & V tools
• Automated reporting
chirpCustomCriteriaTest

Figure 1: chirpFreq

MATLAB EXPO 2016

Questions?