Simulink Test를 이용한 Powertrain Blockset 기반의 동적 검증

홍 혁 기
Agenda

- HEV P2 Model
- Requirement Management in the Simulink Environment
- How to implement VnV Activities using Simulink Test with HVE P2 Model
  - Case1: HVE P2 Performance
  - Case2: HVE P2 Fuel Economy
Nomenclature for HEV Topology

- **P2** parallel HEV
  - P# = Electric machine locations
HevP2Reference Application
HevP2Reference Application
HevP2Reference Application
HevP2Reference Application
HevP2Reference Application
HevP2Reference Application
Powertrain Blockset / Simulink Test Collaboration

- **Plus**
  - Provide starting point for engineers to build good plant / controller models
  - Provide open and documented models
  - Provide very fast-running models that work with popular HIL systems

- **Needs**
  - Systematic V&V workflow using Powertrain Blockset
  - Repeatable and Automatic Process
Model Based Design for Powertrain Blockset

- Simulink Requirement
  - Test Requirement Authoring
  - Test and Modeling Status Check

- Powertrain Blockset
  - Vehicle modeling
  - Simulation

- Simulink Test
  - Test Authoring and Automation
  - Execution and Report Generation
Model Based Design for Powertrain Blockset

HEV P1 Reference Application

Traceable
Author Requirements in Simulink

Author

Track

Manage
Import and Export with External Tools

External Requirements

 import

 Update

 Export

 Unlock
 Edit/Supplement

 ReqIF
 Requirements Interchange Format

 req
 Requirements

 R2019a
**Import External Requirements**

---

**BATTERY MANAGEMENT SYSTEM (BMS) REQUIREMENT**

"Copyright 2018 - 2019 The MathWorks, Inc."

---

**1. OVERVIEW**

This document describes a requirement specification for Battery Management System (BMS). The BMS system provides functionalities to control the battery pack model.
HEV P2 Test Requirement

Test Suite

Index | ID | Summary
--- | --- | ---
1 | #1 | Performance
1.1 | #2 | Zero to sixty
1.2 | #3 | Passing time
1.3 | #10 | Differential Gear ratio
2 | #4 | Fuel Economy
2.1 | #5 | Average fuel economy
2.2 | #6 | Battery SOC Min/Max
2.3 | #7 | Battery SOC Delta
2.4 | #9 | Drive schedule accuracy
The time to reach 60 MPH for the WOT drive cycle shall be less than 20 seconds.
The time to accelerate from 40 to 60 MPH in the WOT drive cycle shall be less than 5 seconds.
Two types of variants should be applied.
- Variant1 : 3.0
- Variant2 : 3.32
The following requirements shall be met for the FTP75, US06 and HWFET drive profiles.
<table>
<thead>
<tr>
<th>Index</th>
<th>ID</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>#1</td>
<td>Performance</td>
</tr>
<tr>
<td></td>
<td>#2</td>
<td>Zero to sixty</td>
</tr>
<tr>
<td></td>
<td>#3</td>
<td>Passing time</td>
</tr>
<tr>
<td></td>
<td>#10</td>
<td>Differential Gear ratio</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>#4</td>
<td>Fuel Economy</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>#5</td>
<td>Average fuel economy</td>
</tr>
<tr>
<td></td>
<td>#6</td>
<td>Battery SOC Min/Max</td>
</tr>
<tr>
<td></td>
<td>#7</td>
<td>Battery SOC Delta</td>
</tr>
<tr>
<td></td>
<td>#9</td>
<td>Drive schedule accuracy</td>
</tr>
</tbody>
</table>

The average fuel economy at the end of the drive profile shall be 32 MPG.
The battery state of charge shall be greater than 20% and less than 90% for the entire drive cycle.
The absolute value of the difference between the battery state of charge at the start of the drive profile and the end of the drive cycle shall be less than 1%.
The actual vehicle speed shall be no more than 2 mph greater than the scheduled vehicle speed within 1 second of the given time.
Simulink Test

Test Harness

Test Manager

Test Sequence
Test Case Templates

- **Simulation Testing**
- **Baseline Testing : Regression Test**
- **Equivalence Testing**

**Temporal Assessment**
- Whenever is true
- Becomes true
- Becomes true and stays true for at least
- Becomes true and stays true for at most
- Becomes true and stays true for between

MathWorks
AUTOMOTIVE CONFERENCE 2019
Simulink Test Configuration

- Traceability
- Model and Harness
- Simulation Iteration with Parameter Change
- Simulation Input: Excel/Mat
- Measurement
- Criteria check
- MCDC, Condition...
Dynamic Test 1 - Performance

- Requirement
  - The time to reach 60 MPH for the WOT drive cycle shall be less than 20 seconds.
  - The time to accelerate from 40 to 60 MPH in the WOT drive cycle shall be less than 5 seconds.
  - Two types of variants should be applied. (Variant1 : 3.0 / Variant1 : 3.32)

- Configuration
  - Test Template
Dynamic Test 1 - Performance

- Requirement
  - The time to reach **60 MPH** for the WOT drive cycle shall be less than 20 seconds.
  - The time to accelerate **from 40 to 60 MPH** in the WOT drive cycle shall be less than 5 seconds.
  - Two types of variants should be applied. (Variant1 : 3.0 / Variant1 : 3.32)

- Configuration
  - Test Template
  - Test Measurement
    - Target Speed
    - Actual Speed
  - Parameter Variant Control
    - Parameter Override
function customCriteria(test)

    try
        vs = test.sltest_simout.logout.get(1).Values;
    end

    vIndex = find(vs.Data>60,1);
    if isempty(vIndex)
        v40Time = Inf;
    else
        v40Time = vs.Time(vIndex);
    end

% Requirement 2
% 60 MPH in the WOT drive cycle shall be less than 5 seconds.
    wotTest.VERIFICATION_Criteria(test);
    vIndex = find(vs.Data>60,1);
    v60Time = vs.Time(vIndex);
    wotTest.WOT_0_60_MAX_TIME, sprintf('0-60 test\n'));
    verifyLessThan(v60Time-v40Time, wotTest.WOT_40_60_MAX_TIME, sprintf('40-60 passing test\n'));
Dynamic Test for HevP2Reference Application
At any point of time, if `actual_speed > 0` becomes true and stays true for at least 10.6 seconds then, starting from end of min-time, with no delay, `actual_speed > 0`.
Temporal Requirement Change

The time to reach 60 MPH for the WOT drive cycle shall be less than 20 seconds.
Temporal Requirement Change

The time to reach 60 MPH for the WOT drive cycle shall be less than 10.6 seconds.
Test Assessments: Formalize and execute requirements

Zero to Sixty

The speed shall at least reach 60MPH after 10.6 sec passes

\[(|x_1 - x_2| \geq x_3)^\varepsilon \land \square_{[0,t_1]}(|x_1 - x_2| \geq x_3) \rightarrow \square_{[0,t_2]}x_4\]
At any point of time, if \( tv > 0 \) becomes true and stays true for at least 10.6 seconds then, starting from end of min-time, with no delay, \( actual\_speed > 60 \) must be true.
Temporal Assessment

Expected Behavior

Actual Result

Explanation

Assessment T

At any point of time, if actual_speed > 0 becomes true and stays true for at least 10.0 seconds then, starting from end of min-time, with no delay, actual_speed > 60 must be true.

Assessment failed when triggered at 0.1 s.
- Trigger condition (actual_speed > 0) is true at 0.1 s.
- Expected response condition to be true at 10.7 s.
- i.e. at 'end of min-time' of trigger.
- Expected 'with no delay, (actual_speed > 60)' must be true' to be true at 10.7 s, actual value is false.
Dynamic Test for HevP2Reference Application
Dynamic Test 1 - Fuel Economy

- **Requirement**
  - The average fuel economy at the end of the drive profile shall be 32 MPG.
  - The battery state of charge shall be greater than 20% and less than 90% for the entire drive cycle.
  - The absolute value of the difference between the battery state of charge at the start of the drive profile and the end of the drive cycle shall be less than 1%.
  - The actual vehicle speed shall be no more than 2 mph greater than the scheduled vehicle speed within 1 second of the given time.

- **Configuration**
  - Test Template
  - Test Measureme
    - Target Speed
    - Actual Speed
    - SOC
    - MPG
function customCriteria(test)
    fuelEconomy = test.slttest_simout.logslout.get(1).Values.Data;
    soc = test.slttest_simout.logslout.get(2).Values.Data;

    % Requirement : Average fuel economy
    test.verifyGreaterThan(fuelEconomy(end), 32, 'Fuel Economy');

    % Requirement : SOC Min/Max
    soc_min = min(soc);
    test.verifyGreaterThan(soc_min, 20, 'Min of SOC');

    soc_max = max(soc);
    test.verifyLessThan(soc_max, 90, 'Max of SOC');

    % Requirement : SOC delta
    test.verifyLessThan(abs(soc(end)-soc(1)), mpgTest.SOC_DELTA, 'State of Charge Delta');

mpgTest.customCriteria(test);
Dynamic Test for HevP2Reference Application
Key Takeaway

- Verification and Validation Process for Powertrain System
- Flexible and Automatic Verification and Validation
- Various Range of Applications
Thank you!