Functional testing via statistical and behavioral signatures for Engine Management Systems Using Matlab/Simulink

Christo Prakash and Vijith Natarajan
Engine Powertain Control

www.continental-corporation.com
Conventional Test Procedure
Modified Test Procedure

Development Process

System Requirements

Conception

Software Requirements Specification

Software Design

SW Design Document

Coding

Module Verification

Code

Integrated Software

Integration and Regression Tests

Tested Integrated Software

Delivery to Customer

Author, © Continental AG

Continental
Model based design work flow

- Fixed point model
- Processor dependent model
- Floating point model
Model based design work flow

- Processor dependent model
- Floating point model
- Fixed point model

1. Requirements
2. Implement Floating Point Model
3. Prepare and Run Test cases
4. Model = Requirements
   - Yes: Store Test Results and Test Report
   - No: Iterate back to Implement Floating Point Model
Model based design work flow

1. Fixed Point conversion and Scaling
2. Run test cases again with same test inputs as floating point model
3. Fixed Point Results = Floating Point Results
   - Yes: Store Test Results and Test Report
   - No: If Mismatch Due to Scaling (Update the floating point model and repeat)
   - No: If Mismatch Due to design problem (Update the floating point model and repeat)
Model based design work flow

- Floating point model
- Processor dependent model
- Fixed point model

Diagram:

1. Generate Code
2. Test Software and Processor dependencies
3. If Results Matching with Fixed Point results
   - Yes: Generate Reports and document
   - No: Update Fixed Point Model
4. If Mismatch
   - Due to scaling: Update the floating point model and repeat
   - Due to design problem: Update the floating point model and repeat
Model based testing work flow

1. Store Test Log and Generate Reports
2. Test Execution
3. Prepare Test Cases
4. Test Vector Creation
5. Module Under Test
Model based testing framework

- **Test Manager** – Centralized environment for creating and managing Test vectors and Test cases
Identification of range of the cut off frequency for fuel tank level monitoring using model based testing
Matlab/Simulink model
Matlab/Simulink model
Simulation Results

Case 1: Cut off frequency 0.5

[Image of graphs showing sensor input signal and filtered output signal]
Simulation Results

Case 2: Cut off frequency 0.08

Sensor input signal

Filtered output signal
Simulation Results

Final Iterative simulation result

Filtered tank volume

Final tank volume
## Simulation Results

<table>
<thead>
<tr>
<th>Filter Coefficient</th>
<th>Signal strength</th>
<th>Performance</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.5</td>
<td>Noisy signal</td>
<td>😞</td>
</tr>
<tr>
<td>0.08</td>
<td>Filtered signal</td>
<td>😊</td>
</tr>
</tbody>
</table>

From the simulation results it is evident that our system is stable and defect free at the design stage itself.

Hence a drastic reduction in the effort spend on HIL /vehicle testing and design rework at later stages.
Challenges faced

Centralized data base management for the environmental model.

Onetime effort to create the environmental model and training the software developer.
Conclusion

The authors feel that with Model based testing the quality of software increases due to the simulation and test capability of MATLAB and the knowledge of the developer.

From the case study and the test framework used it is concluded that better understanding of the vehicle real time behavior and any unforeseen behavior problems can be corrected.

The effective model based testing showed the improved accuracy of the calibration in order to fine tune the vehicle performance.
Thank you for your attention!