Development and Testing of AMT Control Strategy Using Model Based Design Method

By: Ajitsinh A. Yadav
Engineering Research Centre
Tata Motors
Contents

1. Introduction: Background and Target
2. Transmission Control Module (TCM)
3. Gear Shift Subsystem Block Diagram
4. System Requirements: A Brief Overview
5. Scope of Work
6. Control System Design
7. Plant Model
8. LABCAR: Hardware-in-Loop System (HIL)
9. Rapid Prototyping
10. Rapid Prototyping: MotoHawk Interface
11. System Layout for HIL Testing
12. Verification and Validation (V&V) Test Plan
13. Requirement Traceability and Consistency Check
14. V&V: Model Coverage
15. Recalibration, Fault Detection and HIL Testing
16. Mathworks Tools in Development Process
Introduction: Background and Target

Parallel Hybrid Concept:

**Target:** Development of Transmission Control Submodule for a parallel hybrid prototype
Transmission Control Module

- The main function of the transmission control unit is to control the hybrid transmission (refer fig. in slide above) and the gear shift mechanism for the multi-speed gearbox which is a part of hybrid transmission.

- The gearshift mechanism consists of two electromechanical actuators for selection of gear rail (1-2, 3-4, 5-6) henceforth called ‘Select Actuator’ and for shifting from neutral into a gear henceforth called ‘Shift Actuator’.

- It executes the following tasks:
  - Gear shift strategy (decide when to shift gear)
  - Gear shift execution (drive/control actuators)
  - Power split mechanism lock/unlock
  - Motor control for synchronization while shifting gears

- For this presentation we focus on a subsystem of TCM, which drives and controls actuators to execute gear shift
Gear Shift Subsystem Block Diagram

- The block diagram below shows a subsystem from the transmission control module.
- We are going to focus our attention on this subsystem for the rest of the presentation.
- A brief overview of system requirements is given in the slide below.
Scope of Work

We followed a proprietary ‘5 Level Model Based Development V Model’ framework for product development, testing and validation.

Following activities are carried out:

- Requirements: Definition and Management
- Proof of Concept/Model Development using Model Based Design tools, MATLAB, Simulink and Stateflow
- Model in Loop Testing
- Rapid Prototyping
- Hardware in Loop Testing
System Requirements: A Brief Overview

Following is a list of some of the system requirements.

- System should be able to drive and control actuators to engage desired gear
- System should inform about status of gear shift i.e. gear shift in progress/completed
- Feedback about gear engaged eg. 1,2,3 etc
- Shift time constraint eg. Shift within ‘X’ secs.
- Position accuracy of actuators should be within tolerance
- Repeatability
- Other diagnostic and error ‘flag’ requirements

Example:

When the system completes a gearshift(refer section ‘Terminology’ above for definition of ‘gearshift complete’), the function shall set the value of output signal Gear_Shift_Status equal to 1.
Control System Design

- Top Down approach was used for system design
- Top level system model was created using empty ‘shells’ for subsystems
- Subsystems are interfaced using buses
- Buses are defined using bus objects
- Bus objects allowed better control and traceability
- Bus objects are under revision control
- Simulink Projects helped in organizing files and setting up workspace
- For version control we used Simulink’s built in source control tools
Plant Model

Plant model of a seven speed manual transmission was created

Plant Model was created using Simscape and Simscape Driveline toolboxes in Simulink

Simscape greatly simplified modeling process because:

- It allows use of physical components instead of modeling based on governing equations
- Connections are bi-directional
The following components/effects were modeled:

- Electromechanical actuators with encoders
- Shift linkages, shift fork, shift sleeve, detent mechanism
- Synchronizers, dog clutch, gear pair
- Custom components also created for plant model in Simscape with support from Mathworks team
- In the model, the force to be applied by the shift actuator for synchronization was a function the inertias in the system, synchronizer parameters and the speed difference between input and output gears
- The current drawn by the actuator is proportional to the force required for synchronization
LABCAR: Hardware-in-Loop System (HIL)

- LABCAR is the HIL system for automotive ECUs developed by ETAS GmbH
- We have used this system for our HIL testing activities
- LABCAR architecture consists of the following system components:
  - Real-time simulation target, model configuration, signal I/O cards, operator interface, extensions
- LABCAR supports MATLAB, Simulink and Simulink Coder
- LABCAR uses its own Target Language Compiler (TLC). However Simscape code does not pass through TLC.
- Simscape pre-compiled libraries are not used by LABCAR
- Hence, static runtime libraries needed by code generated by Simscape need to be compiled during code generation

Rapid Prototyping

- The control strategy developed for the gear shift subsystem is deployed on MotoHawk Controller.
- MotoHawk Controller is an embedded control module developed by Woodward Inc. for automotive applications.
- MotoHawk is an application development tool built on top of Matlab and Simulink.
  - It is used to program MotoHawk Controllers.
  - It leverages the powerful capabilities of Simulink to develop and deploy application software on MotoHawk Controllers.
    - It has its own library of components which can be accessed through Simulink Library Browser.
- MotoTune is the application used for calibrating the controller.

# MotHawk Control Solutions: https://www.woodward.com/WorkArea/DownloadAsset.aspx?id=2147484104
Rapid Prototyping: MotoHawk Interface
System Layout for HIL Testing

Rapid Prototyping Controller

- Shift Act. PWM signals
- Shift Act. Direction
- Select Act. PWM signals
- Shift Act. Direction
- Shift Act. 2 CH ENC feedback
- Select Act. 2 CH ENC feedback

Breakout Box

LABCAR
Hardware in Loop system
Verification and Validation (V&V) Test Plan

- Verification and Validation was carried out as per a prescribed framework
- Following activities were carried out

<table>
<thead>
<tr>
<th>Stage/Level in V Model</th>
<th>Tests Carried Out</th>
</tr>
</thead>
<tbody>
<tr>
<td>Model Development</td>
<td>Requirement Traceability</td>
</tr>
<tr>
<td></td>
<td>Consistency Checks</td>
</tr>
<tr>
<td></td>
<td>Check compliance to MAAB guidelines</td>
</tr>
<tr>
<td>MIL Testing</td>
<td>Functionality tests using test cases generated through DOE</td>
</tr>
<tr>
<td></td>
<td>Model Coverage, Decision Coverage</td>
</tr>
<tr>
<td></td>
<td>Formal Verification using Simulink Design Verifier</td>
</tr>
<tr>
<td>SIL Testing</td>
<td>Functionality tests</td>
</tr>
<tr>
<td>HIL Testing</td>
<td>Functionality tests</td>
</tr>
<tr>
<td></td>
<td>Test to verify compliance to set targets of shift time, shift accuracy, repeatability, fault detection etc.</td>
</tr>
</tbody>
</table>
Requirement Traceability and Consistency Check

Requirements Traceability Report for Gearshift_Controller_Submodule

Table of Contents:
1. Model Information for "Gearshift_Controller_Submodule"
2. Traceability Summary for "Gearshift_Controller_Submodule"
3. System - GearShiftActuatorController
4. Chart - Gear_Shift_Act_Loc

List of Tables
1. Gearshift_Controller_Submodule
2. Artifacts linked in model
3. Objects in Gearshift_Controller_Submodule - GearShiftActuatorController that have Requirements Traceability Links
4. Objects in "Gear_Shift_Act_Loc" that have requirements

Chapter 1. Model Information for "Gearshift_Controller_Submodule"

Requirements Consistency Checking

- Identify requirement links with missing documents
  - Passed

- Identify requirement links that specify invalid locations within documents
  - Passed

- Identify selection-based links having description fields that do not match their requirements document text
  - Passed

- Identify requirement links with path type inconsistent with preferences
  - Passed
V&V: Model Coverage

• Model Coverage checked using Simulink Design Verifier
• Detailed report gives instances where certain combinations have not occurred.
• Using state flow debugger we identify root cause of coverage issue and modify the model
• Following is a snapshot from the Model Coverage Report

<table>
<thead>
<tr>
<th>Model Hierarchy/Complexity</th>
<th>Test 1 Decision</th>
<th>Condition</th>
<th>MCDC</th>
<th>Execution</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. GearShift_Controller_Submodule</td>
<td>78 74%</td>
<td>78%</td>
<td>50%</td>
<td>100%</td>
</tr>
<tr>
<td>2. GearShiftActuatorController</td>
<td>77 74%</td>
<td>78%</td>
<td>50%</td>
<td>100%</td>
</tr>
<tr>
<td>3. Gear_Shift_Act_LoC</td>
<td>76 74%</td>
<td>77%</td>
<td>49%</td>
<td>100%</td>
</tr>
<tr>
<td>4. SF_GearShiftActuatorController-Gear_Shift_Act_LoC</td>
<td>75 74%</td>
<td>77%</td>
<td>49%</td>
<td>100%</td>
</tr>
</tbody>
</table>
Recalibration, Fault Detection and HIL Testing

- Repeatability in actuator displacement is ensured by recalibrating the displacement encoders.
- System recalibrates itself during operation
- Software has routines to detect faults in actuation system
- HIL tests were carried out to verify functionality of system
- HIL testing was also done to verify compliance to
  - set targets of shift time,
  - shift accuracy, repeatability,
Mathworks Tools in Development Process

• Mathworks has built an ecosystem of tools around Model Based Product Development Process
• Mathworks tools were used in all stages of product development including
  • Requirements linking, traceability and consistency check using Simulink Requirements
  • Project management and revision control
  • Controller and plant model development using Simulink, Stateflow and Simscape
  • Model-in-loop testing
  • V&V activities like checking against standards, checking against errors, proving properties etc.
  • Model and Code optimization by doing code coverage and decision coverage tests
• These tools provided clear visibility, helped reduce development time, helped develop robust and optimized code, allowed collaboration on development activities etc.
• Mathworks tools integrate seamlessly with third party tools while retaining most of their functionality
THANK YOU!