Vehicle System Control Software Implementation and Validation for a Hybrid Powertrain

Yanan Zhao, Yulei Chen, Chat Nguyen, Raju Nallapa and Ray Spiteri

Vehicle & Battery Controls
Research and Advanced Engineering
Introduction

In the automotive industry, with the advent of various alternate fuel vehicles, vehicle control software development is becoming a focal point. The challenges associated with the development of such systems include hardware integration, model development, cycle timing, and model validation. The ever-increasing challenge demands improvements in the software development and validation process.

For Hybrid Electric Vehicles (HEVs), the Vehicle System Control (VSC) is a functional supervisory controller that interfaces with other vehicle systems. For the Dual Drive hybrid powertrain development project, an SVT enhancement was developed for the VSC implementation.
Overview of the Hybrid Powertrain

• The Dual Drive system has two propulsion torque paths – the engine and the front-axle motor deliver propulsion torque to the front axle through a transmission; – for the rear axle, the propulsion torque is delivered by the rear-axle motor.

• There are three main driving modes in vehicle operation – Electric Drive: the vehicle is propelled electrically using the battery to power the rear-axle motor. – Series Drive: the engine drives the CISG to generate electricity for the ERAD to propel the vehicle – Hybrid Drive: the engine is the primary driving source while the rear-axle motor can provide additional electric boost torque if necessary.
Overview of VSC Software Architecture

- The Integrated Modeling Environment (IME) is used for the VSC control software development. It provides the necessary framework and tools to develop and manage complex engineering projects.

- The IME supports the development of real-time systems, enabling engineers to simulate and test systems under various conditions.

- The IME facilitates the integration of various components, allowing for efficient and seamless system development.

- The scalability of the IME is a key feature, enabling the development of systems of varying complexity and size.

- The Torque Manager, a critical component in the system, can be decomposed into features, and each feature contains subsystems running at different task rates.
Overview of VSC Software Architecture (2)

• At each task rate the subsystem is composed of Function Identifications (FIDs), and a FID represents the lowest level functional unit with a unique ID which implements specific function requirements of a feature.
VSC Software Validation Process

- Software validation process is part of the model-based control development process which includes:
  - unit testing;
  - model-in-the-loop (MIL) testing;
  - hardware-in-the-loop (HIL) testing;
  - vehicle testing.

The evolution of controls software validation strategy with different test platforms illustrates that the software functions are tested gradually:
- From simulation to real-time hardware implementation;
- From unit level (FID), to component level (feature) and then to vehicle level.
Unit Testing

- tests each implementation of the algorithms at the lowest level of functionality;
- is usually performed using open-loop desktop simulation within the Simulink environment;
- incorporates developing test harness and generating test cases;
- tests the algorithms and outputs based on various conditions as defined from functional requirements.
Unit Testing: Process

• For VSC implementation with the IME environment, unit testing and FIDs integration process can be split into two paths:
  – For hand-coded FIDs, PolySpace testing tool can be used first to perform automatic ANSI C basic checks and run time error checks.
The use of PolySpace testing tool increases software quality, productivity, and reduces warranty costs by detecting errors at an early stage. The test enhances the proposed control development process for the mixed C/Simulink implementation.
Model-in-the-loop Environment

- In the model-based design process, a plant model provides the basis for both the MIL and HIL testing.
- Vehicle Model Architecture (VMA) is a standard modeling structure at Ford – all key vehicle functions are represented with subsystems in the VMA; for subsystems that have their own controllers ... of these controllers; these controller models simulate the sensing, control and actuation local to each subsystem.

- For the MIL testing, the plant model is integrated with the actual VSC control algorithms.

- A HIL plant model is obtained from the MIL plant model.

VehSysCtrl (Vsc)
Transmission (Trn)
Steering (Ste)
Powerplant (Pwp)
Online (HIL)
Environment (Env)
Electrical (Ele)
Driver (Drv)
Driveline (Dln)
Chassis (Cha)
Bus
Brakes (Brk)
Auxiliaries (Aux)
Ms_GloblBus
<Ms_CtrlGloblBus>
<Ms_PlntGloblBus>
<Ms_Drv GloblBus>
<Ms_Env GloblBus>
Ms_MainBus
<Ms_HilGloblBus>
Model-in-the-loop Testing

- **MIL testing**
  - provides closed-loop desktop simulation to test the integrated VSC functions;
  - allows developers to validate features and interactions between FIDs with vehicle response against user specified input signals and calibrations.

**Test inputs used for a MIL test include:**

- **Driver inputs,**
  - from a standard drive cycle, or a user pre-defined drive cycle, or interactive user operated drive cycle;
- **Overridden inputs,**
  - signals/calibrations defined with specified ranges or values;
- **Inputs provided by vehicle plant feedback.**

**A feature Design Verification Method (DVM)**

- contains test plans to validate VSC features against feature design requirements;
- is used to setup the tests, conduct the tests, and determine whether the desired results are met.
Hardware-in-the-loop Testing

- In HIL environment, the auto-generated C code for the integrated VSC software exists in MicroAutoBox rapid prototyping module. The rest of the vehicle system including the plant and other controllers are running in a simulator.

- HIL testing allows the developers to test VSC functionality with hardware in real-time and also provides the capability to test the I/O interface.

- It can significantly reduce the cost and usage of the vehicle while detecting potential issues at an early stage.
Hardware-in-the-Loop Testing (2)

- The VSC HIL consists of the following hardware:
  - dSPACE mid-size simulator;
  - dSPACE MicroAutoBox;
  - break-out-box;
  - dSPACE DS830 multilink panel;
  - signal conditioning box;
  - shifter;
  - PC.

- For the Dual Drive HEV, the VSC HIL was developed as a platform to validate:
  - VSC hardwired inputs and outputs;
  - low and high level driver software;
  - CAN inputs and outputs;
  - signal conditioning box;
  - feature level functions;
  - high level functions.

- HIL testing uses the same DVM as used in the MIL testing.
Vehicle Testing

- In vehicle testing, all major VSC functions were tested in the prototype vehicle.
- The New European Driving Cycle (NeDc) was a good tool for testing.
- In addition, the actual hardware components can provide the nonlinear inputs to the system and present dynamic interactions in many degrees of freedom which is not possible in the MIL/HIL environment.
Simulation vs. Target Validation

- A Simulation vs. Target (SVT) validation method was developed to validate the response and functionality of the VSC in simulation to be identical with the target environment.
- In SVT implementation:
  - In the target (vehicle/HIL) environment, a test (e.g., Wide Open Throttle) is conducted and its inputs and outputs are captured at a specific time.
  - Then, the SVT simulation is designed to capture the simulation outputs.
  - After that, simulation and target outputs are compared graphically.

- The SVT process was used as part of the VSC validation. Some output discrepancies were observed, and the related VSC implementation defects were fixed.

![Diagram of VSC Validation Process](image-url)
Conclusions

• With the complex systems and rapid pace of control software development, a robust control software validation process was developed for the Dual Drive hybrid powertrain. • All software feature functional requirements were successfully validated with achieved functional readiness gateway milestone. • This process can be generalized to model-based control software validation with mixed C/Simulink environment.