Development of a Hardware in the Loop Setup with High Fidelity Vehicle Model for Multi Attribute Analysis

Jae Sung Bang[1]   Tae Soo Kim[1]   Suk Hwan Choi[1]   Raphael Rhote-Vaney[2]   Harikrishnan Rajendran Pillai[2]

[1]Eco-Vehicle Control System Development Team, Hyundai Motor Group, South Korea, `aeromec@hyundai.com`
[2]MBSE Engineering Services, Siemens PLM, USA, `raphael.rhote-vaney@siemens.com`

Abstract

This paper describes a novel model based real time simulation approach to test, validate and calibrate electronic controllers for Hybrid Electric Vehicle (HEV) applications. The performance of the Hybrid Control Unit (HCU) needs to be evaluated on multiple vehicle attributes such as fuel economy, acceleration and drivability objectives. The multi attribute evaluation requires a higher level of detail for the vehicle simulation model where the energy flow and drivetrain dynamics are represented accurately. Given the high mechatronic content and the strong interactions among the various controllers in HEVs, it becomes necessary to simulate many of the vehicle controllers on the real time platform. The higher fidelity vehicle model coupled with the realistic behavior model of the controller network poses challenges in setting up the real-time Hardware In the Loop (HIL) test platform where the vehicle level attributes can be studied. The real time simulation setup process, its challenges and the methods used to overcome these challenges are described in this paper.

Keywords: hardware in the loop, Amesim, hybrid electric vehicle

1 Introduction

The expectations from the consumer has transformed radically in the recent years from the advent of enhanced driver support, better fuel efficiency and improved powertrain technologies. Automotive manufacturers and suppliers are confronted with ever greater complexity as a result of increasing numbers of products and options, shorter technology cycles and the increasing pressure to innovate. At the same time they need to balance the needs and demands of customers, investors, regulators, non-governmental organizations and even the general public (Pwc, 2014). The passenger vehicles are being transformed to mechatronic machines with high electronic and software contents. Companies cannot afford to test such complexity in a hardware prototype thoroughly because of the extremely high costs associated with design changes far down the development cycle (Boehm,2005) and the extraordinary lead time associated with such a task.

Companies are using model-based simulation approaches to design and test such high complexity mechatronic systems. Testing the controller unit comprehensively before testing in the prototype requires real-time Hardware in the Loop (HIL) test platforms. Most of the existing literature in the HIL area (Nabi, 2004; Ramaswamy, 2004; King, 2004; Allende, 2015; Hafiz, 2015; Bovee, 2015; Isermann, 1999; Basrah, 2015; Wei, 2004) refers to the usage of HIL testing in the context of controller logic validation and testing.

It is difficult to find a study that focuses on real time HIL simulation of a complete closed loop vehicle model that captures both fuel economy and drivability phenomenon accurately. It is in this scenario, that the process we have created (i.e. performing vehicle level multi-attribute analysis with controller in the loop on HIL) becomes unique.

The paper describes the process of real time closed loop HIL validation of a hybrid electric vehicle model with the objective of validating the supervisory hybrid control algorithm on the basis of both fuel economy and drivability characteristics of the vehicle.

The second section in this paper describes the hybrid electric vehicle architecture, scope of the multi-attribute analysis and the load cases used to study the multiple attributes. The Amesim© model used to describe the Hybrid Electric Vehicle (HEV) powertrain, the controller model architecture and the process followed for modeling and validating the system is described in the third section. The real-time model generation process and the partitioning of the model to execute on the multi-core platform to optimize execution performance are discussed after that. This is followed by results discussion and conclusions.

2 Multi-attribute analysis of hybrid electric vehicles

This section describes the architecture of the hybrid electric vehicle under consideration, functionality of the Hybrid Control Unit (HCU), other controllers and the

test cases. Within the scope and context of this paper, multi-attribute analysis refers to the simultaneous analysis of fuel efficiency and drivability characteristics of the vehicle. The entire simulation setup has been created in order to study the impact of the HCU strategy on the drivability and fuel efficiency related aspects of the vehicle.

The configuration of the vehicle in this paper is referred to as a Transmission Mounted Electric Device Hybrid Electric Vehicle (TMED HEV). As seen in Figure 1, the architecture is composed of a Hybrid Starter and Generator (HSG), an engine, an engine clutch, an electric motor and a transmission. The HSG is connected to the engine by a pulley and the main role of the HSG is to start the engine, control the engine speed to engage the engine clutch and charge the battery by using engine power. The engine clutch plays an important role in the mode of the vehicle such that when the clutch is disengaged, the engine power is not transferred to the driving wheels and only motor power drives the vehicle. This setting is
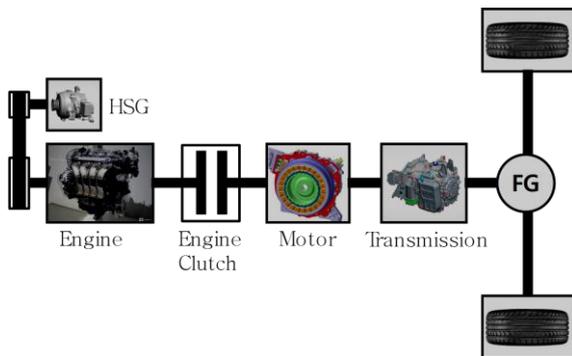
Figure 1. HEV Architecture

known as the Electric Vehicle (EV) mode. When the clutch is engaged, the engine power and the motor power drive the vehicle in HEV mode. During the HEV mode, the assistance by the electric motor ensures that the operating point of the engine is maintained to have optimum fuel efficiency. The 6-speed automatic transmission is similar to a conventional transmission except for the fact that it does not have a torque converter to reduce energy loss.

The optimal control of the operation of the vehicle is managed by an array of control units. Out of these, the HCU is the supervisory controller that computes the torque demanded by the driver and optimally distributes it among the engine, electric motor and the HSG. The torque demand from the driver is estimated by the HCU by considering multiple factors such as the accelerator pedal position, the brake pedal position, vehicle speed, gear ratio and torque interventions by other controllers. For example, if the battery state of charge is sufficiently high and the torque demand is less than the maximum motor torque at the current motor speed, then the HCU generally opts for the EV mode of operation in which all of the demanded torque is supplied by the electric motor.

On the other hand, in the case of HEV mode, the HCU distributes the demanded torque between the electric motor and the engine by considering the brake specific fuel consumption of the engine and the efficiency of the electric motor at that particular operating point. Hence it can be seen that the HCU plays a critical role by ensuring the driver demands are met and simultaneously ensuring optimal fuel efficiency.

In addition to the supervisory HCU, there are other subsystem level controllers (explained in the next sections) that also have to be modeled to an appropriate level of detail to ensure the simulation results are realistic and comparable to vehicle test data. For instance, the Engine Management System (EMS) controls fuel, air and spark in order to produce the commanded torque from HCU. If this feature is not modeled correctly, it can lead to significant deviations in the prediction of fuel economy and battery state of charge.

In order to validate the performance of the model across the two attributes of drivability and fuel economy, the following test cases are utilized:

- For fuel economy analysis:
  - o FTP drive cycle
  - o US06 drive cycle
  - o NEDC drive cycle
- For drivability analysis:
  - o Fixed gear HEV mode tip & creep
  - o HEV/EV mode change

3   System modeling for multi-attribute analysis

This section highlights the modeling aspects of the plant model using Amesim© software and the controller models   using MATLAB®/SIMULINK®/STATEFLOW®. A brief description of the process followed to validate the models is also provided.

Since the HCU is to be tested against Fuel Economy and Drivability requirements, a dynamic model of the vehicle using Amesim© is developed with the physics needed to capture the energy flows and conversion from fuel to mechanical and electrical energy. In order to also address drivability requirements, the level of details for the description of the elements involved is chosen to capture natural frequencies in the 0 to 20Hz range which corresponds to the frequencies that can be felt by the driver as seen in Figure 2

Initially, the vehicle model is developed focusing on these considerations, the real-time capabilities not being part of the requirements considered at the time. In order to capture the targeted frequency content, all the mechanical elements, whose modes are known to fall within that range, are included in the model (i.e. drivetrain, engine 3D block and mounts, suspension and

chassis). The 6DOF engine block on its mounting system and the corresponding rigid body modes are shown in Figure 3.

In order to use the vehicle model for predicting Fuel Economy, the efficiencies and energy losses of the main components should be modeled. Being able to track and account for the power flows is key to understanding how the fuel energy is converted to the mechanical energy delivered to the wheels. Once this torque/energy balance is achieved, the engine brake torque and speed are used to calculate the corresponding fuel consumption. The BSFC map used to predict the engine fuel consumption
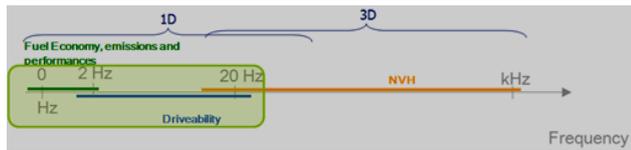
Figure 2. Frequency range of model
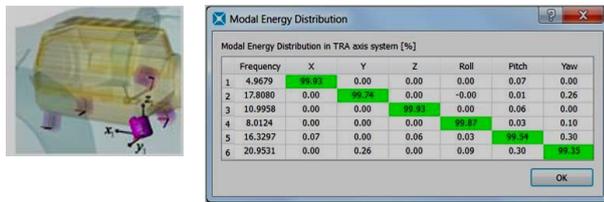
Figure 3. Modal energy distribution

HCU is linked to the rest of the controllers through Controller Area Network (CAN) and these controllers in turn are connected to each other and to the plant hardware.

As stated previously, HCU is the high level supervisory control unit that manages the torque split between the internal combustion engine and the electric motor according to the operating conditions. The HCU computes and sends the main signal of each controller such as the engine torque command, the motor torque command, the battery charge and so on. The other controllers carry out the command from the HCU. The Engine Management System (EMS) controls the fuel quantity, air quantity and ignition timing in order to realize the torque command from the HCU. The EMS also estimates the amount of fuel consumption. The Motor Control Unit (MCU) controls the electric motor by controlling the current. The MCU also ensures a reduction in driveline oscillations by appropriately controlling the motor torque and also shifts the operation point of the engine to achieve better fuel efficiency. The HSG, which controls the engine speed for engine clutch engagement and which is used to charge the battery, is also controlled by the MCU. The Transmission Control Unit (TCU) determines the shift point and provides the commands to actuate the clutches and the brakes within the automatic transmission. The
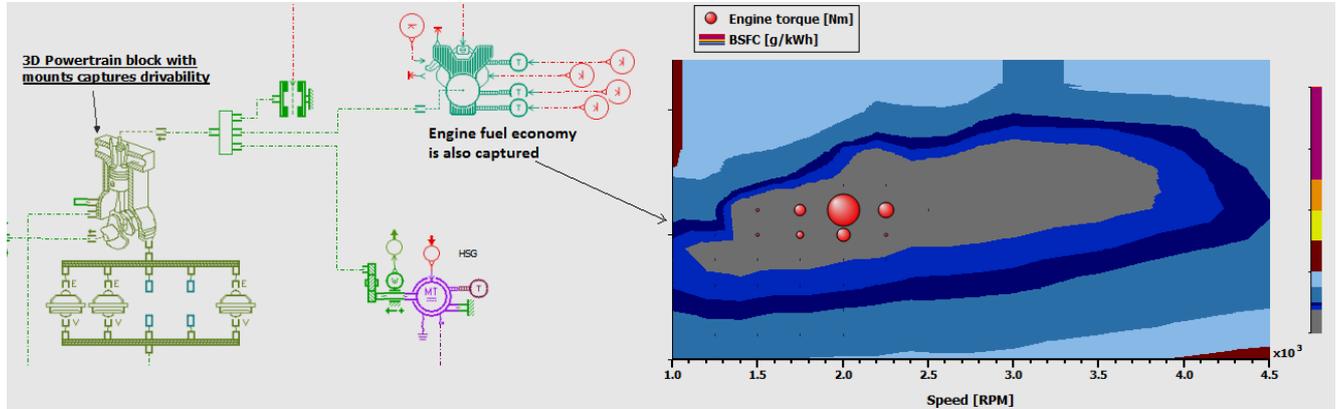
Figure 4. Model captures drivability and fuel economy

from engine torque and speed is shown in Figure 4.

It is noteworthy that the plant model architecture mirrors that of the physical hardware so as to capture the functions and physics needed to address the requirements of interest and capture the corresponding physical phenomena. The complete model being sizeable, it is not presented in this paper but it shall be introduced during the conference.

Now moving on the controller modelling aspects, a summarized version of the overall vehicle controller network architecture is outlined in Figure 5. The figure corresponds to the system that is modeled within the scope of this particular project. It can be seen that the

Battery Management System (BMS) monitors the state of the battery at all times and decides the charging/discharging limits based on the operating point. The BMS also computes an estimate of the state of charge (SOC) of the battery. The Anti-lock Braking System (ABS), within the context of this project, computes the total braking torque required based on the driver input and then splits them appropriately so that a certain percentage of the total braking requirement is supplied using the hydraulic brakes and the rest is produced by the motor as regenerative torque. The above controller network structure (excluding the HCU), along
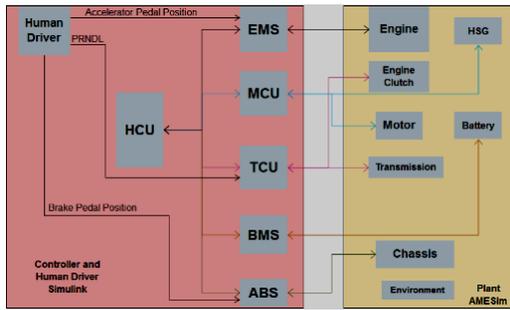
Figure 5. Model Architecture
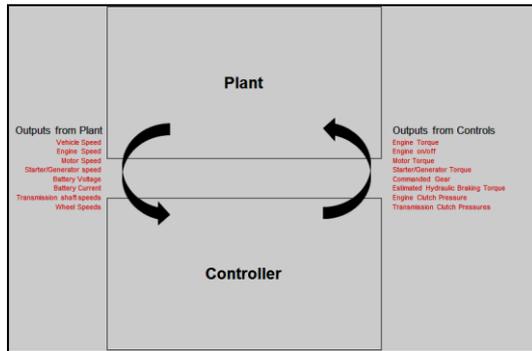
Figure 6. Plant-Controls Interface

Amesim© plant modeling is carried out at the component level first, which is then assembled to obtain the complete vehicle model. Validation is also ensured at both the component and system level of the plant. The controller models are also built and validated in a systematic way, from unit level to system level The process of controller model building – from unit level logic, to complete 'X'CU, to the entire controller is highlighted in Figure 7.

The validation process also follows the development process – open-loop unit level validation is followed by open-loop system level validation. Once this is completed for both the plant and control model, closed-loop validation is performed on the desktop. The final validation step is performed on the HIL bench using the HCU hardware. The entire validation process is concisely represented in Figure 8.

4   Real-Time Model Generation and Setup

Initially developed to capture drivability phenomena like clutch Judder or Shift surge, the Amesim© model contains excessively high natural frequencies and very small time-constants that cannot be handled by a fixed

Figure 7. Controller modeling approach

with a simplified implementation of the algorithm which includes the main function of each controller, is modeled in Simulink.

Since the vehicle plant model is developed using the Amesim© software and controller models are developed using the MATLAB®/SIMULINK®/STATEFLOW® suite, it is very important to define the interface between these two entities early on in the development process. Effort is taken to ensure that the virtual interface is similar to the actual interface implemented on the real vehicle, to the extent possible. The signal interface between Amesim© (plant model) and MATLAB® (controller model) is shown in Figure 6.

Modeling and validation of the plant model and controls are done following a systematic procedure with confirmation and validation ensured at each stage. The

time-step solver. In order to use in an HIL setup several steps have to be taken for the model to get rid of the unnecessary high frequency while keeping the lower frequency content used to capture drivability phenomena. The first step is to identify the largest frequency or mode and the main contributing states. Fig. 9 shows the Modal Projection Tool that is used to that effect.

In a second step the part of the model contributing to this mode is simplified. Then the new results are compared to that of the original model to validate that differences in time-domain and frequency-domain are acceptable while the model is being significantly reduced. In the present case, the inertia of the HSG, connected to the engine flywheel inertia via a very stiff and over-damped shaft, generates the 47MHz mode shown in Figure 9. This may not be a problem for a

variable time step solver but that will be a problem for a fairly typical fixed time-step. In this example the two inertias can be lumped together and the very high mode disappears. This process is then repeated until all the unnecessary high frequencies have been removed. Figure 10 shows natural frequencies of the system after model reduction and the drivability results after vehicle model reduction is plotted in Figure 11. The frequency, magnitude, vehicle speed, and acceleration are very similar to vehicle test data, and the results are sufficiently accurate to predict the behavior of the powertrain.

The following subsection highlights certain aspects on the controller side that needs to be taken care of before deploying on a real-time bench.

The different controllers are to be executed at various sample periods and also the communication intervals among the different controllers vary. All of these factors have been taken into consideration for controller modeling. For the purpose of desktop Model in Loop (MIL) simulation, the controller models and the plant model (Amesim© S-function) are within the same Simulink .mdl file. However, for the HIL simulation, the plant model (Amesim© S-function) is simulated in a separate mdl file. The structure of HIL model is shown in Figure 12.

Figure 8. Validation Approach

Figure 9. Natural frequencies before Transmission model reduction

Figure 10. Natural frequencies after Transmission model reduction
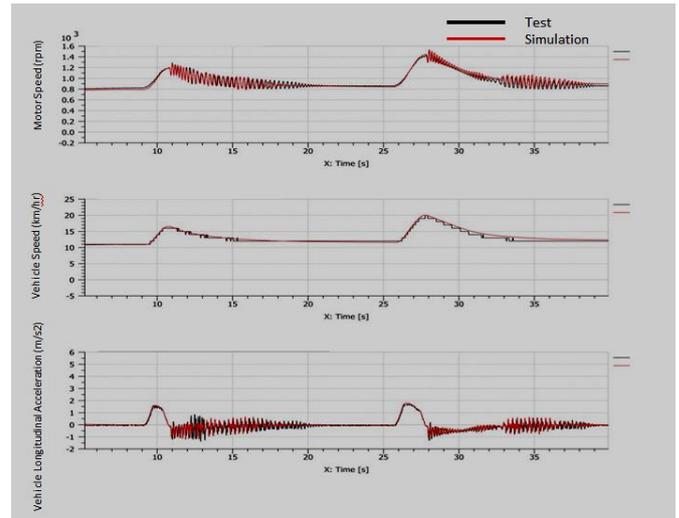
Figure 11. Drivability Results
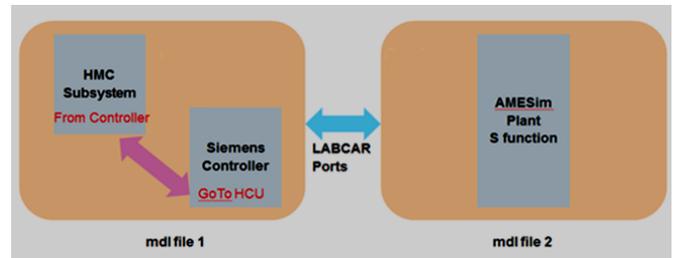
Figure 12. HIL model structure

Here, mdl file 1 contains the controller blocks. This model is executed using a fixed time step solver. Internal clocks within this model take care of the separate trigger rates required for the different control systems. The second mdl file contains the plant model S function and this file is executed at a fixed time step which is smaller than the controller models. Since the system model is split into two separate mdl files for

Figure 13. Closed loop MIL model structure
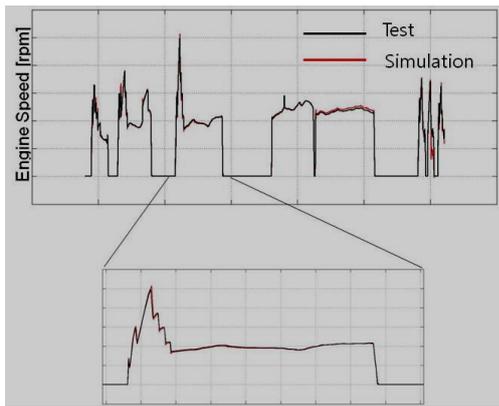
Figure 14. MIL Results

controller models and plant model, it is possible to simulate them on two separate cores on the HIL bench.

Once the real time plant model and controller models are ready, the closed loop setup is validated against test data on desktop. For this purpose, an S function of the HCU is utilized.

The overall model structure used to test the system closed loop on the desktop is outlined in Figure 13. It can be seen that the entire controller model components are in the Simulink environment and the plant model is implemented as a MATLAB® S function. The entire system is simulated in fixed time step using the Simulink solver. The HCU is also implemented as a MATLAB® S function. This setup enables a closed loop fixed step simulation on the desktop before it is deployed on the HIL bench.

Figure 14 shows the result of the MIL simulation using the HCU s function. The engine speed error between test and simulation is extremely small. This close behavior between model and test data on desktop ensures the accuracy of the model before deployment on HIL.

5 HIL Test Bench Results

The novel simulation model on the HIL bench is used to analyze the impact of HCU algorithm change or calibration change on the fuel efficiency and drivability aspects of the HEV. The HIL bench enables evaluation of the performance of the HEV without real vehicles thereby reducing the development time significantly.

The HIL bench used for the multi-attribute analysis and the verification/validation of HCU is shown in Figure 15. The HIL bench is composed of the real-time PC, the operating PC, I/O board and HCU. Using LABCAR© software, the vehicle model and controller model suggested in this paper are built and the INCA© software is used for measurement of signals and calibration.

Some of the issues related to HIL deployment is outlined in this paragraph. The main issue that can cause problems during HIL execution is model overrun. This means that all of the computations associated with the

system model are not being completed within the allocated fixed time step. When this is encountered, the model can be split and run on multiple cores. This would entail having two separate mdl files with LABCAR© ports for communication between them. Simulation debugging can also be a challenge in this case because of the number of constituents in the system − plant, controller models, HCU hardware, interfaces, different cores etc.

The results presented in this section reflect the Sonata HEV vehicle. Various versions of the prototype HCU are used to correlate the test result and the simulation result. In the figures, red line corresponds to the test data from the real vehicle and the blue line corresponds to the simulation data.

Figures 16, 17 and 18 show the full closed loop simulation results obtained from HIL simulation with HCU hardware connected. The HCU, vehicle and the controller models have the same calibration values. A subsection of the result from FTP drive cycle is shown in Figure 16. For reasons of security, the result for the entire time range is not provided. The vehicle speed error between the test data and simulation is below 2 km/hr. Since the error is less than the acceptable maximum speed error value of 3.2 km/hr, other variables of the simulation is expected to match the performance of the Sonata HEV for the FTP drive cycle. The engine torque error between the test and simulation is below 3%. The difference in the timing of engine on/off point is less than one second which is very small considering the whole range of the FTP drive cycle (1300s). The small difference in the engine on/off signal arises from the error in vehicle speed, SOC, and so on. The errors for SOC and the fuel consumption are also below 3%. Since all errors are within acceptable limits, the prediction of the fuel economy is possible for the FTP drive cycle. Figure 17 shows a subsection of the results of the Highway drive cycle. The errors for vehicle speed, engine torque, battery SOC and the fuel economy is below 3%. It is to be noted that the simulation model used for validating the highway drive cycle is exactly the same as the one used for FTP. The characteristics of the FTP and the highway drive cycles are vastly different. The FTP drive cycle can be considered to be somewhat mild in terms of the rates of acceleration and braking. The highway drive cycle, on the other hand is aggressive. Since the simulation model is able to match the vehicle performance for both of these cycles, any other cycle's fuel economy can be predicted. Finally, the result of NEDC drive cycle is shown in Figure 18. The results are sufficiently similar to the test data of the real vehicle and therefore the fuel economy for the NEDC is predicted by the developed model.

Figure 15. HIL bench for HCU
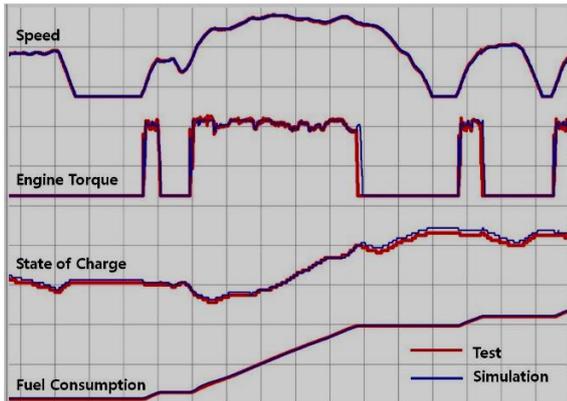
Figure 16. HIL FTP cycle results
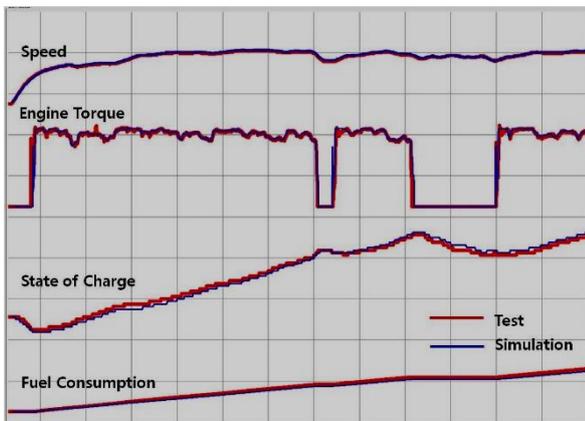
Figure 17. HIL Highway cycle results
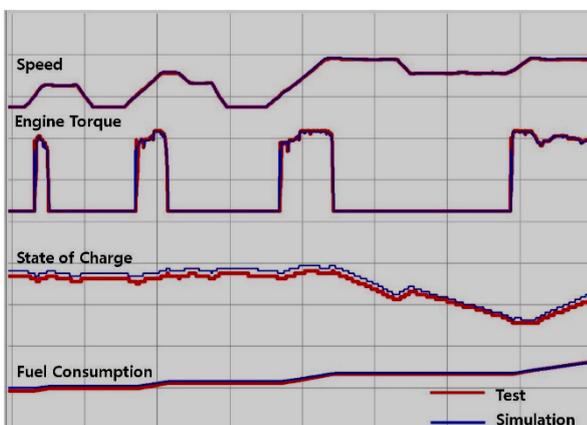
Figure 18. HIL NEDC cycle results

6 Conclusions

The results of the HIL simulation show a very close correlation between the real vehicle and the model. This confirms that closed loop vehicle simulation on an HIL bench can be used to validate multiple attributes like drivability and fuel economy. This process opens up new opportunities for similar multi-attribute studies on the HIL bench. This approach is novel when compared to existing studies that tend to focus on a single attribute using lower fidelity plant models for controls logic validation.

This setup enables Hyundai Motor Company (HMC) to quickly test the impact of different HCU algorithms and calibration values on the vehicle drivability and fuel economy. The overall development time for the HEV has been reduced and the performance of the HEV has been improved.

The vehicle model presented in this paper does not include thermal effects on the powertrain and also has limited predictive capabilities of the accessory loads.

As a next step, HMC and Siemens would like to include the effects of the temperature on the engine, electric motor and the battery. An improved model which includes the electrical energy consumption by the air conditioner, electric oil pump and the electric power steering will provide a better estimate of the energy consumption. In addition, if the road geometry and the vehicle lateral dynamics are added to the model, a variety of other test scenarios can be simulated accurately.

Acknowledgements

We would like to thank other team members (HMC, Siemens PLM colleagues in Lyon, Detroit, Chennai and Seoul) who indirectly contributed to this paper.

References

M. Allende, P. Prieto, B. Hériz, J. M. Cubert, and T. Gassman. Advanced Shifting Control of a Two Speed Gearbox for an Electric Vehicle. In *The 28th International Electric Vehicle Symposium and Exhibition*, pages 118-128, Korea, 2015.

M. S. Basrah, E. Velenis, and D. Cao. Four wheel torque blending for slip control in a hybrid electric vehicle with a single electric machine. In *ICSEEA*, pages 19-24, IEEE, 2015.

B. Boehm and V. R. Basili. Software defect reduction top 10 list. *Foundations of empirical software engineering: the legacy of Victor R. Basili*, 426(37):135-137, 2005.

K. Bovee, A. Hyde, M. Yatsko, M. Yard, M. Organiscak, B. Hegde, J. Ward, A. Garcia, S. Midlam-Mohler, and G. Rizzoni. *Plant Modeling and Software Verification for a Plug-in Hybrid Electric Vehicle in the EcoCAR 2 Competition*. SAE Technical Paper, 2015.

F. Hafiz, P. Fajri, and I. Husain. Effect of brake power distribution on dynamic programming technique in plug-in series hybrid electric vehicle control strategy. In *IEEE*

Energy Conversion Congress and Exposition (ECCE), pages 100-105, IEEE, 2015.

R. Isermann, J. Schaffnit, and S. Sinsel. Hardware-in-the-loop simulation for the design and testing of engine-control systems. *Control Engineering Practice*, 7(5):643-653, 1999.

P. J. King, and D. G. Copp. Hardware in the loop for automotive vehicle control systems development. In *UKACC Control 2004 Mini Symposia*, pages 75-78, IET, 2004.

R. Mura, V. Utkin, and S. Onori. Energy management design in hybrid electric vehicles: A novel optimality and stability framework. *IEEE Transactions on Control Systems Technology*, 23(4):1307-1322, 2015.

S. Nabi, M. Balike, J. Allen, and K. Rzemien. *An overview of hardware-in-the-loop testing systems at Visteon*. SAE Technical paper, 2004.

D. Ramaswamy, R. McGee, S. Sivashankar, A. Deshpande, J. Allen, K. Rzemien, and W. Stuart. *A case study in hardware-in-the-loop testing: Development of an ECU for a hybrid electric vehicle*. SAE Technical Paper, 2004.

X. Wei. *Modeling and control of a hybrid electric drivetrain for optimum fuel economy, performance and drivability*. Doctoral dissertation, The Ohio State University, 2004.

R. Hanna, and F. Kuhnert. *How to be No. 1: facing future challenges in the automotive industry*. PwC Autofacts, 2014.