Virtual Integration for hybrid powertrain development, using FMI and Modelica models

Lionel Belmon[1] Yujung Geng[1] Huaqiang He[2]

[1]: Global Crown Technology, Beijing, China
Lionel.Belmon@globalcrown.com.cn

2: Dongfeng Commercial Vehicles Technical center, Wuhan, China
hehuaqiang@dfcv.com.cn

Abstract

Dongfeng Commercial Vehicles (DFCV) is developing powertrain controls for a hybrid light truck. To support this development, a virtual integration platform is being implemented, using Modelica models and Functional Mock-up Units (FMUs) for the engine/EMS, gearbox, MCU/e-motors, driveline, tyres and longitudinal dynamics. Simulink models and/or c-code of the Hybrid Control Unit (HCU) and Transmission Control Unit (TCU) are also integrated in the platform to achieve closed-loop simulation. The virtual integration allows reproducing accurately the overall vehicle behavior and is used for optimization of gearshifts, hybrid mode switches and hybrid drive strategies.

Keywords: Hybrid powertrain, FMI, Control software

1 Motivation and objectives

The control systems of hybrid powertrain are generally implemented using several ECUs networked together. Functions are distributed through the controllers. Typically, the powertrain controllers will include a Hybrid Control Unit (HCU), an Engine Management System (EMS), a Transmission Control Unit (TCU), a Motor Control Unit (MCU), and a Battery Management System (BMS). Other controllers from chassis systems (ESP/ABS) might also interact with the powertrain.

The development of such a system requires considering the interactions of all main components together, namely engine, gearbox, actuation system, e-motors, battery, driveline, tyres. For the control software, it also implies that functions cannot be developed independently but are now inter-dependent and distributed over several ECUs. This poses challenges to classical development processes.

Dongfeng Commercial Vehicles is developing such a hybrid powertrain for application in a light truck. We give a schematic of the system under consideration in Figure 1. DFCV wishes to have an efficient tool for integrating HCU and TCU control logic, optimizing parameters and performing system testing. This tool should be available for function developers, should be cost-efficient and deployable.

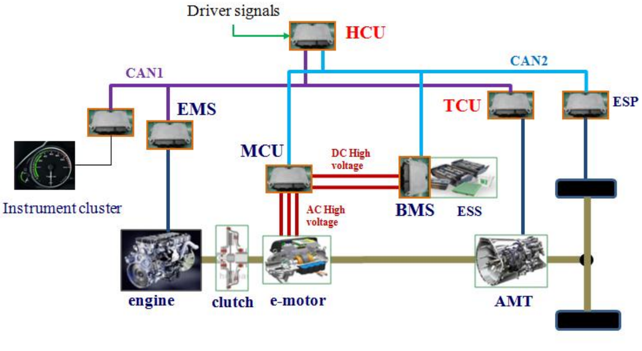


Figure 1 : schematic of the hybrid powertrain

The objective of the project described in this paper is thus to establish a *virtual hybrid powertrain* where subsystems and controllers can be simulated together on a standard PC. Function developer can then easily verify on their PC the behavior of any changes in software or parameters. This virtual powertrain can also be used for other applications, such as large coverage testing or system/parameters optimization.

DOI
10.3384/ECP14096419

For creating this *virtual hybrid powertrain*, the following tools are used:

ITI SimulationX – plant models authoring
SimulationX is a Modelica platform that provides powerful commercial libraries for powertrain modeling and electrical systems modeling [1]. The tool fully support the FMI standard and application has been demonstrated for powertrain applications [2]. Models can be exported to a standard format and can be executed without license restrictions. This is an important feature for deployment of the *virtual powertrain.*

Simulink – HCU and TCU software authoring
Dongfeng Commercial Vehicles developed HCU and TCU control software using a model-based approach in Simulink. Simulink floating-point models can be compiled and integrated in the virtual powertrain. This would be then a “model-in-the-loop” setup. It is also possible to use the final production-code (fixed point) and calibration parameters for integration inside the virtual powertrain. This method could be described then as Virtual ECUs. [3]

QTronic Silver – Integration platform
QTronic Silver is an integration platform, widely used for powertrain applications [4],[5],[6]. Silver provides the simulation core, an interactive GUI dashboard, numerous interfaces and tools for integrating plant models and control software or even performing ECU chip simulation. The Virtual ECUs described in the previous paragraph are built with the Silver Basic Software (SBS) technology.

QTronic TestWeaver – Large coverage testing
Powertrain systems, and in particular hybrid powertrain systems are systems difficult to test because of the very large number of system states and larger number of state transitions. For instance, a relevant test campaign should test all gearshifts, in various slopes. It should also test all hybrid mode transitions, under various State Of Charge (SOC). The test space is huge and test scripts/manual testing is not an efficient method. QTronic TestWeaver is an intelligent test system that can generate test cases to increase test coverage, drive the system under test to uncovered states and report problems/bugs when these are met. TestWeaver has been successfully applied in a large number of powertrain projects [5],[7],[8].

2 Plant models

The plant models to be developed will be used for development support of HCU and TCU. This means that all subsystems and all remaining controllers will be modeled in the plant. This includes in particular control models for EMS, MCU and BMS.

We can define several general requirements for the plant models :

- Simulate all required bus&sensors signals (>200)
- Simulate EMS,MCU and BMS logical functions
- Simulate the necessary physics
- Simulate fast enough for convenient use
- Simulation should be accurate enough to support optimization of relevant system parameters

We describe now how plant models are developed.

2.1 CAN buses

The hybrid powertrain under consideration use 2 CAN networks, CAN1 and CAN2. Dongfeng Commercial Vehicle provided the complete list of CAN messages. The 2 CAN networks are implemented as Modelica connectors.
The connectors are then used in the simulated controllers for EMS, MCU and BMS. They are also used for defining I/O variables of the FMU that will be exported from the vehicle model. The CAN1/CAN2 networks have together around 200 signals. In SimulationX, the network is considered as ideal without losses or delays besides the ECUs task cycle time. Non-ideal behavior of the CAN network is implemented and simulated in QTronic Silver which supports special features for this.

2.2 Automated Manual Transmission gearbox

The AMT model consists of input and output inertias, gear stages and synchronizers. Drag torques and efficiency losses are also included. The AMT stiffness is lumped inside the synchronizers hubs.

DOI 10.3384/ECP14096419

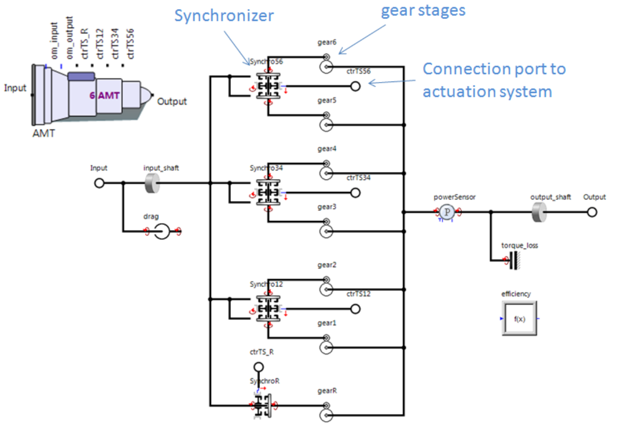


Figure 2 : AMT gearbox model

The synchronizer models are complex behavioral models that fully reproduce the synchronization process, with synchronization torque depending on actuator force. The synchronizer behavioral model has been compared to full contact-based synchronizers models in [9]. Some results of the comparisons are given in Figure 3. The detailed synchronizer model with dog-clutch gear contacts has been validated in [10].

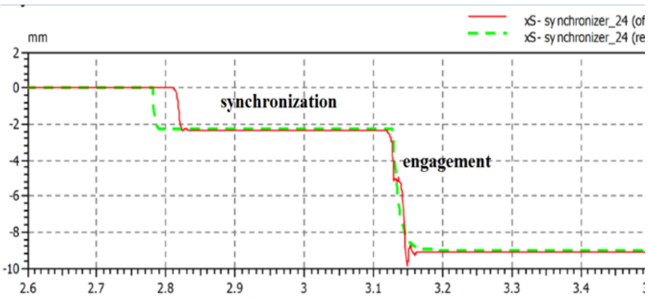


Figure 3 : Comparison of synchronization and engagement process for detailed (red)/functional (green) synchronizer models

We can conclude that the behavioral model computes correctly the synchronization time and the engagement process. The behavioral model has the advantage of requiring few parameters and of being a “fast-running” model. It has been applied in the past in real-time simulations for HiL [9]. The synchronization time is an important quantity for the gearshifting simulation and must be simulated correctly.

2.3 Clutch model

The clutch model can be separated in 3 subcomponents:

Friction/torque model:

The torque capacity of the clutch is defined as a function of the clutch actuator position. This is implemented through a look-up table. Clutch wear is included in the modeling. Finally this torque capacity is used in a stick/slip friction model.

Spring force on actuator side:

The clutch actuator has to overcome the conical spring force. The spring is non-linear and has a strong hysteresis. This is implemented using a hysteresis table from SimulationX

Thermal model:

The clutch friction surfaces temperature is an important quantity to simulate. This is done by computing losses in the clutch during slip and using thermal capacities and heat transfer models.

2.4 Gearbox actuators

The gearbox actuation system is a pneumatic actuation system controlled by solenoid valves. The model is created using the SimulationX pneumatic library, where pressure/temperature are computed using mass/energy balances and compressible flow equations. The solenoid valves receive PWM signals from the TCU. The pneumatic actuation model includes all solenoid valves, gearshift cylinders and clutch cylinders. A selector gate model is also implemented. We give the overview of the pneumatic actuation model in Figure 4

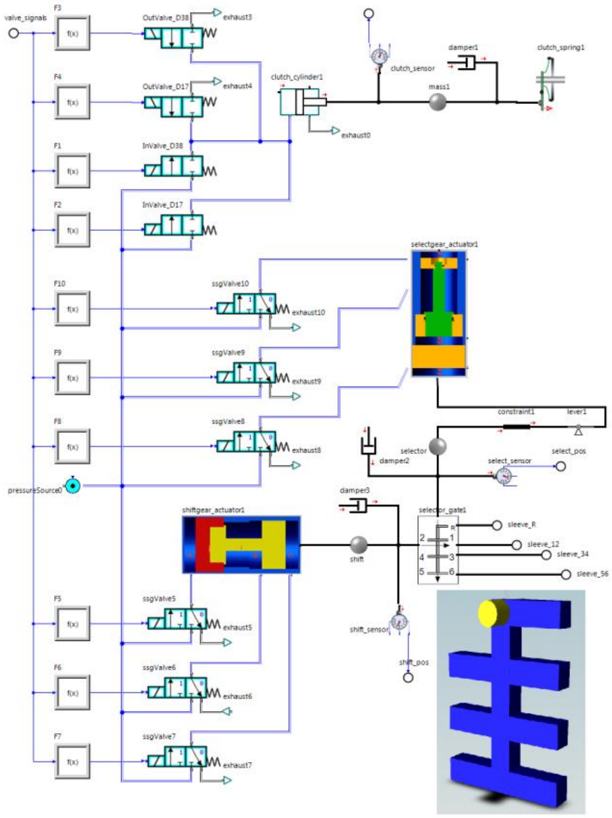


Figure 4 : Pneumatic actuation model – integrated in the vehicle model

The pneumatic actuation model can reproduce the clutch actuation dynamics. In particular, the clutch actuation is a closed control with a position feedback. The model can be used for tuning of control gains and control laws.

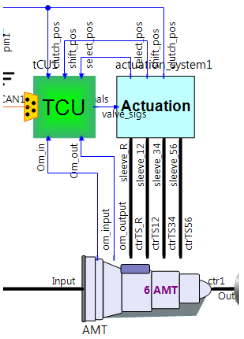


Figure 5 : AMT with "packaged" actuation system and a soft-TCU model

2.5 Engine and EMS

The engine and EMS models are based on tables defining the torque/rpm capacity of the engine, along with the fuel consumption. Turbocharger dynamics are so far neglected, but the turbocharger delay in boost pressure will probably be implemented in a model revision. Besides this rather simple table-based modeling, a state-chart is also added for simulating the logical behavior of the EMS.

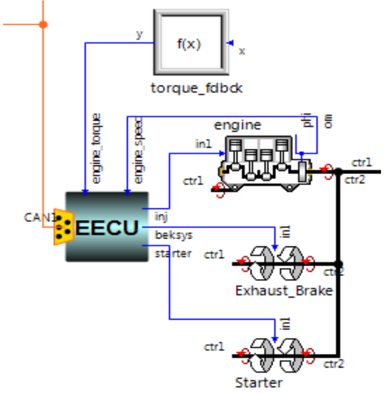


Figure 6 : Engine and EMS model

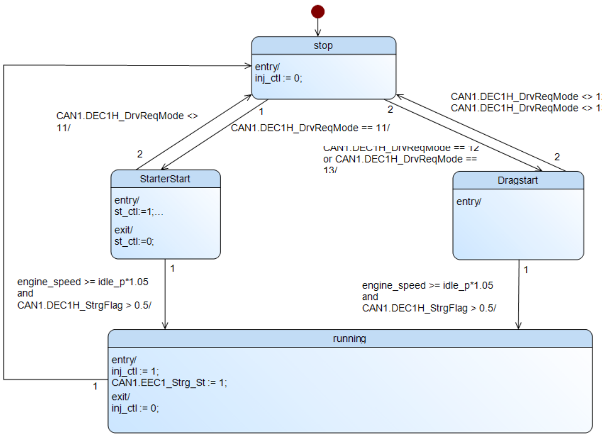

stop
entry/
inj_ctl := 0;
CAN1.DEC1H_DrvReqMode <>
11/
1
2
CAN1.DEC1H_DrvReqMode == 11/
CAN1.DEC1H_DrvReqMode <> 1
CAN1.DEC1H_DrvReqMode <> 1
CAN1.DEC1H_DrvReqMode == 12
or CAN1.DEC1H_DrvReqMode ==
13/
2
2
StarterStart
entry/
st_ctl:=1;...
exit/
st_ctl:=0;
Dragstart
entry/
1
engine_speed >= idle_p*1.05
and
CAN1.DEC1H_StrgFlag > 0.5/
1
engine_speed >= idle_p*1.05
and
CAN1.DEC1H_StrgFlag > 0.5/
running
entry/
inj_ctl := 1;
CAN1.EEC1_Strg_St := 1;
exit/
inj_ctl := 0;
1

Figure 7 : EMS state-chart

The EMS model must simulate the various operation mode of the engine, according to CAN requests. For instance, the EMS have a self-start mode and a "drag start" mode where the e-motor will start the engine by closing the clutch. The state-chart editor of SimulationX is thus very valuable for creating such models.

2.6 E-motor and MCU

The e-motor model is based on an energy/power approach. The AC 3-phase modeling is not considered and we only focus on the DC interface. The model is thus based on an efficiency approach in which the current on the DC side is computed as a function of DC voltage and motor torque. As for the EMS, a state-chart is also included in the MCU model to represent the logical behavior of the MCU software.

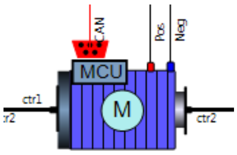


Figure 8 : e-motor and MCU model

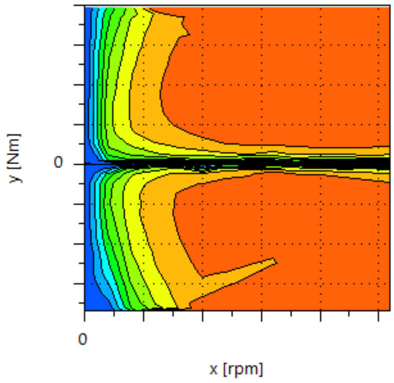


Figure 9 : efficiency map in the e-motor

Motor cooling circuit is so far not modeled in detail because the focus is on HCU and TCU.

2.7 Battery and BMS

The battery model is based on a Open Circuit Voltage table and on parasitic and polarization resistance/capacity. The diagram of the model is shown below.

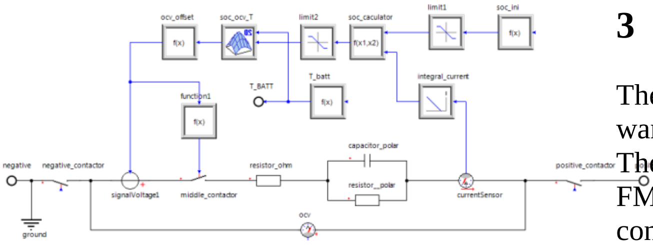


Figure 10 : Diagram of the Battery model

As for the EMS and MCU, the Battery Management System (BMS) uses a state-chart for handling the various states and transitions of the BMS software. The model is finally packaged as shown in Figure 11.

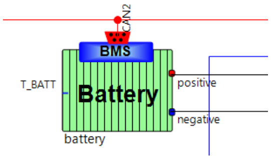


Figure 11 : Packaged battery and BMS model

2.8 Vehicle model overview

We give an overview of the vehicle model in Figure 12.

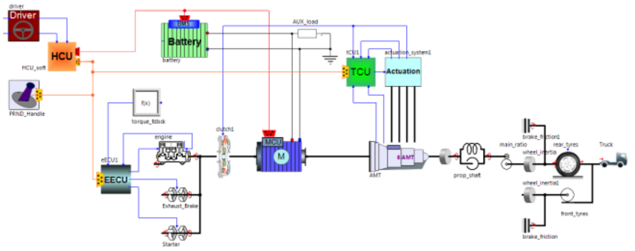


Figure 12 : Diagram view of vehicle model

The vehicle model includes tyre with slip and driveline/propeller shaft with stiffness. The stiffness of the driveline is so far lumped and set according to an equivalent stiffness computed from CAD drawings. This assumption will probably be reviewed in future model updates.

2.9 Export of the model as an FMU

The plant model input/outputs are defined in the Code Export wizard of SimulationX. In particular the definition of CAN buses is helpful. The model is exported so that exported FMU variable names correspond exactly to the names defined in the CAN bus definition. This will be an important property when doing integration in Silver.

3 Hybrid powertrain integration

The integration flow of the HCU/TCU control software and plant is summarized in the figure below. The plant model is exported from SimulationX as a FMU, using the Functional Mock-up Interface. The control models are built using Simulink Coder and Silver Basic Software scripts.

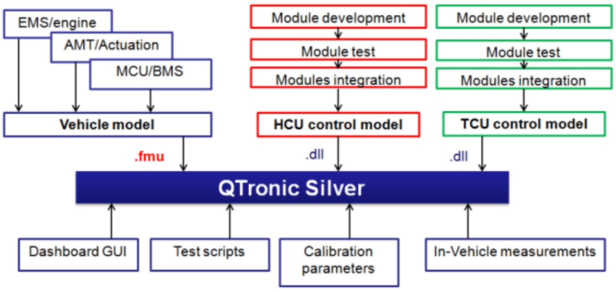


Figure 13 : Integration flow for plant model and controllers

Silver-built controller models also provide access to all internal signals and all internal parameters of the Simulink models. This provides very valuable support for analysis, debugging and calibration of controllers.

4 Validation/calibration of models

4.1 BMS – MCU system validation

The High voltage circuit of BMS, MCU and DC accessories load is validated independently by using comparison between in-vehicle measurements and simulation results. The model unknown/uncertain parameters are first calibrated. These unknown parameters are typically the polarization resistance and polarization capacity and some MCU control software properties. The model used for this calibration/validation work is shown in Figure 14. Boundary conditions such as motor speed and motor torque target are imposed according to vehicle measurements. Some results are presented in Figure 15 and Figure 16.

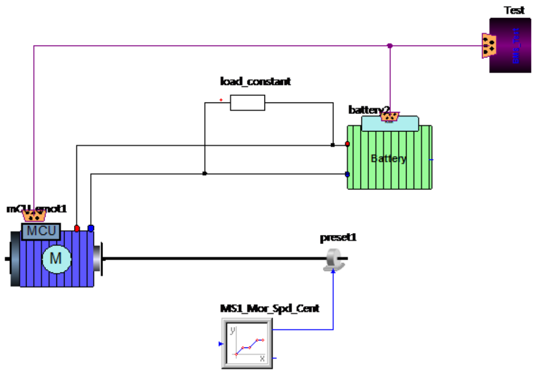


Figure 14: High voltage Electrical system validation model

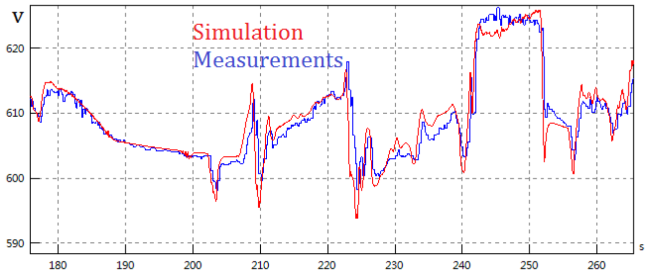


Figure 15 : Battery voltage, comparison of in-vehicle measurement and simulation results

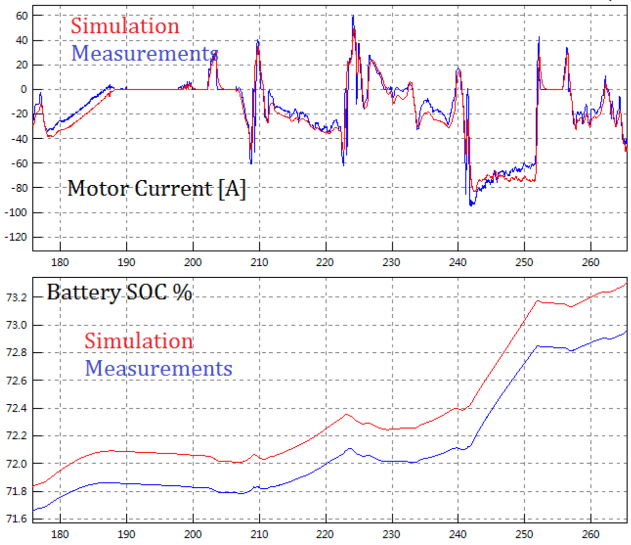


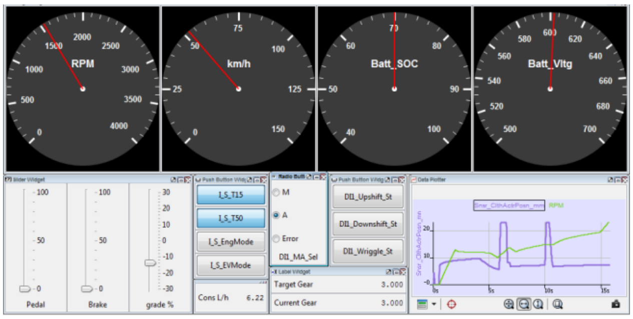
Figure 16 : Comparison of in-vehicle measurements and MCU/BMS currents and Battery State Of Charge

The results for the BMS/MCU validation gives a very good agreement for the MCU current, battery SOC and battery voltage, including transients effects on the battery voltage. Difference between simulated SOC and recorded SOC is a static offset due to initial value of SOC.

4.2 Complete vehicle simulation

A special attention was given first to increase simulation speed. For instance unnecessary events in the exported FMU were removed, some high frequencies dynamics were neglected because irrelevant for the work at hand. However, since we are using variable step solvers, quantifying exactly simulation speed is not possible since it will depend on the driving case. We give however rough estimates:
-When the vehicle is idle or in steady-state without gearshifts, simulation runs 20x faster than real-time.
-When the driving sequence involves numerous gearshifts, simulation runs 3x faster than real-time.

We give in Figure 17 a partial view of the graphical controls and displays used in the QTronic Silver, where the plant model, HCU and TCU are integrated together, with around 250 signals exchanged at 10ms time cycles between controllers and plant. The complete vehicle validation is still under progress at the day of writing.



Cons L/h 6.22

Target Gear 3.000

Current Gear 3.000

Figure 17 : Extract of dashboard/instruments for interactive simulation in QTronic Silver

5 Applications and benefits

The above virtual powertrain will be used for several applications that we list below.

5.1 Control software development support

The virtual powertrain in QTronic Silver has very desirable properties to support the work of function developers for the TCU and HCU.

Accurate plant models from SimulationX

The plant models from SimulationX are accurate and represent the overall behavior of the complete vehicle, including all CAN signals and details of EMS, BMS, MCU control logic. The controllers can then be accurately checked and tuned against such plant models.

Integration of calibration and measurements

Silver provides easy integration of in-vehicle measurements standard formats. This allows test engineers to provide data to function developers. Function developers can easily analyze problems met in the vehicle, define fix and corrections, and verify effects in the simulation.

Integration of actual production c-code

During this project, we are currently integrating floating-point Simulink models. A next step will be the integration of actual production c-code, with fixed point arithmetic, tasks, along with A2L definition. The final production code used in the TCU and HCU can thus be fully tested, using Virtual ECUs.

Control software Debugging in Silver

Simulink models built from Silver can access all internal signals and parameters of a model. This provides powerful function for debugging issues in control logic, arguably more comfortable and efficient than natively in Simulink.

5.2 Large coverage testing

Once an accurate/realistic virtual powertrain is available, it is possible to use it for large coverage testing using TestWeaver. QTronic TestWeaver can generate 1000's of test scenarios overnight and find problems and issues in the system. This increases system quality and system safety early in the development cycle. Compared to hand-written test scripts, TestWeaver scenario generation is a systematic process that will explore a large number of combinations and states.

In the case of the hybrid truck under consideration, state coverage objectives include gearshifts and hybrid mode transitions.

5.3 System optimization

The virtual powertrain established in the project can be used for system optimization, including control software optimization and calibration optimization. This work is efficiently supported since any function developer can introduce modifications and quickly get results on the new implementation. Moreover, Silver also supports scripting so that selected parameters can be automatically optimized using numerical optimization routines.

6 Conclusions

A virtual powertrain for a hybrid truck is being established in collaboration with Dongfeng Commercial Vehicles. For creating this system, several tools are being used, including ITI SimulationX, QTronic Silver and Matlab/Simulink.

The Virtual powertrain provides accurate simulation of the complete vehicle, down to gearshift synchronization events. Battery SOC and e-motor currents have been validated against in-vehicle measurements.

This Virtual Powertrain is applied for system optimization, HCU/TCU control logic debugging and large coverage testing with TestWeaver. These tasks can be conducted efficiently by engineers using PC-based deployable simulation.

References

[1] www.itisim.com.

[2] A.Abel, T.Blochwitz, A.Eichberger et al. Functional Mock-up Interface in Mechatronic gearshift simulation for commercial vehicles, 9th International Modelica Conference, 2012, Munich.

[3] A.Junghanns, R.Serway, T.Liebezeit, M.Bonin. Building virtual ECUs quickly and economically. ATZ Elektronik, 03/2012, Volume7.

[4] E.Chrisofakis, A.Junghanns. Simulation-based development of automotive control software with Modelica. Dresden: Modelica international conference, 2011.

[5] M.Tatar, Schaich, Breitinger. Automated test of the AMG speedshift DCT control software. Berlin: 9th CTI Innovative Automotive Transmissions Symposium, 2010.

[6] J.Mauss, M.Simons. Chip simulation of automotive ECUs. 9th symposium Steuerungssysteme fur automobile Antriebe, 2012, Berlin

[7] N.Papakonstantinou, S.Klinger, M.Tatar. Test-driven development of DCT Control Software. 8th International CTI Symposium Innovative Automotive Transmissions, 2009, Berlin.

[8] M.Neumann,M.Nass,M.Tatar. Absicherung von Steuerungssoftware fur Hybridsysteme, Autoreg 2011, Friedrichshafen.

[9] L.Belmon, J.Yan. Modeling and simulation of DCT gearshifting for real-time and high-fidelity

analysis. SAE China-FISITA conference, F2012-C04-014, 2012

[10] A.Abel, U.Schreiber, Valsania, Fornelli. Simulation based design of gearboxes for high-performance sports cars. Modena:11th HTCES conference, 2005.