Proceedings
of the 3rd International Modelica Conference,
Linköping, November 3-4, 2003,
Peter Fritzson (editor)

Leo Laine, Johan Andreasson
Chalmers Institute of Technology; Royal Institute of Technology, Sweden:

Modelling of Generic Hybrid Electric Vehicles

pp. 87-94

Modelling of Generic Hybrid Electric Vehicles

Leo Laine, CTH, Sweden, laine@mvs.chalmers.se
Johan Andreasson, KTH Vehicle Dynamics, Sweden, johan@fkt.kth.se

Abstract

The software development of the control functions will be a large part of the work when developing future vehicles. Therefore, it is of great importance to be able to reuse the control architecture for different hardware configurations. In this work, a generic[1] control architecture for Hybrid Electric Vehicles has been modelled with Modelica. Functional decomposition was used to develop the generic control architecture. Functions are identified and placed into a hierarchical partitioning structure. Three functional levels are suggested; main control level, subsystem level, and actuator/sensor level. The main control contains a driver interpreter, energy management, vehicle motion control and a strategic control. These main functions are made independent of hardware and of hybrid configuration. The subsystem level contains driver interface, chassis, power supply and auxiliary systems. Two models, a parallel and a series hybrid electric vehicle, are used to demonstrate the implemented architecture.

1 Introduction

In order to handle the complexity of several actuators/sensors interacting in future Hybrid Electric Vehicles (HEVs) and to allow easy change of hardware configuration, a control architecture with suitable functional partitioning is necessary.

There are three main types of architectures for partitioning; centralised, hierarchical, and peer, as shown in Figure 1. The centralised architecture collects information from all sensors and computes references for all actuators. The benefit is that all signals are available simultaneously. The drawback is the lack of modularity that makes it hard to add new functionality. The hierarchical structure consists of a top level control block and several low level control blocks. This allows good modularity and also a central controller is available to coordinate the interaction between the actuators/sensors. The peer-to-peer architecture is the most modular one, but without a coordinator between the different actuators/sensors conflicts will be hard to avoid.

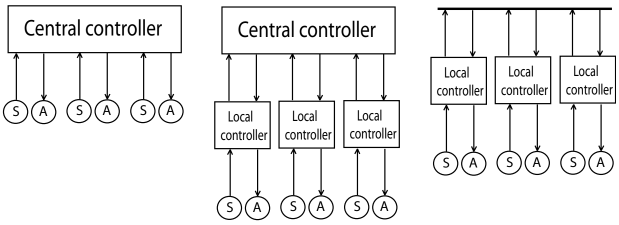


Figure 1: Centralised, hierarchical and peer-to-peer architecture.

The architecture should be generic and work for several types of HEV configurations such as parallel, serial, and split etc and must therefore be modularised. It must also fulfill the requirements on interfaces between automotive suppliers and manufacturers so that brand specific qualities can be kept in-house. For both these demands, the hierarchical control architecture is suitable.

The purpose with the suggested control architecture is to easily handle the variety of vehicles that the authors believe will be found a decade from now and further on. These future vehicles could be serial HEVs with fuel cell as primary power unit, and with wheel units that can apply driving, steering, and suspension forces independently. However, to be really useful, the architecture must also be able to handle today's vehicle in a well defined way.

Modelica [1] was chosen as a platform for test and validation of ideas concerning generic modelling of HEVs. The aim is to study how HEVs can be modelled as a complete system and combine different areas of interest, such as: control, energy management, and vehicle dynamics. The first step is to evaluate if the suggested generic control architecture really is generic by modelling different hardware configurations with Modelica. The second step is to study how the specific strategies within Main Control should be designed. Finally, the sensibility to faults and inaccuracies will be studied. In this paper the ideas behind the architectures are first briefly described[2] and then the implementation

---

[1] Generic: hardware and configuration independent

[2] See [2] for a more thorough explanation.

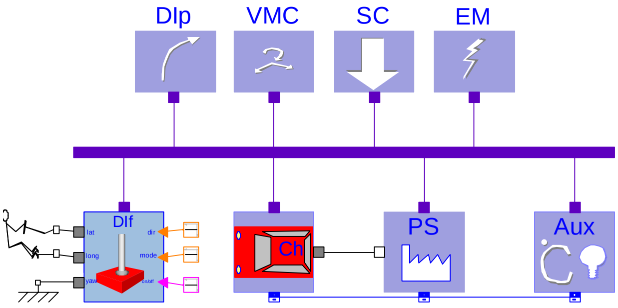


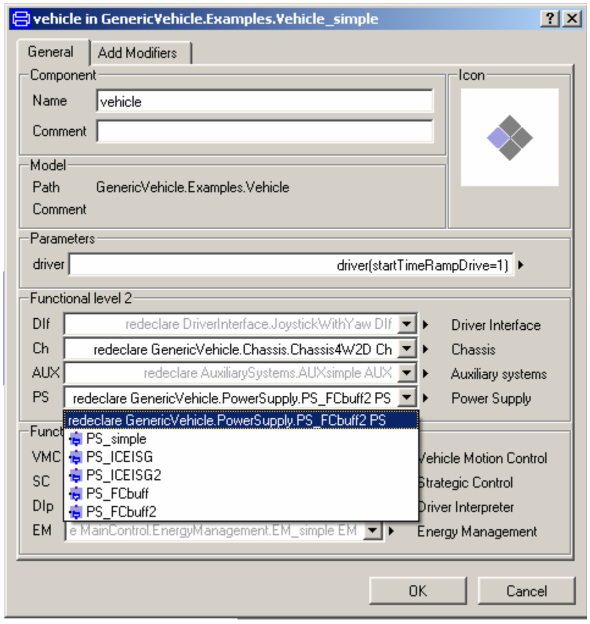
Figure 2: Main model architecture illustrating the main functions within functional levels 1 and 2.

in Modelica is discussed.

2 Main model architecture

The main model architecture is divided into different functional levels. The highest functional level is called main control and includes the following functions; Driver Interpreter (DIp) interprets the driver's demands as a desired path, Vehicle Motion Control (VMC) that controls the vehicle according to these demands and Energy Management (EM) assures that this is done in an energy efficient way. Additionally there is the Strategic Control (SC) which finalizes the orders from Vehicle Motion Control and Energy Management to the lower functional levels. It is only Strategic Control that can send orders to lower functional levels. This to uphold the causality of the orders. If a critical state is recognised by Energy Management or Vehicle Motion Control, Strategic Control will give priority to suggested signals from either part. The functional level 2 contains the following: Driver Interface (DIf), Chassis (Ch), Power Supply (PS), and Auxiliary Systems (Aux).

In Figure 2 the main model architecture implemented in Modelica shows functional levels 1 and 2. All functions exchange generic signals via a bus, and the chassis, power supply and auxiliary systems are coupled with standardised mechanical and electrical connectors. This allows each model to be changed without having to redesign the others. In Figure 3 this is illustrated by a menue that shows how different HEV configurations can be set up. Figure 4 shows the signal flow between functional levels 1 and 2. Auxiliary systems and Driver Interface are here excluded for simplicity.


vehicle in GenericVehicle.Examples.Vehicle_simple

Component
Name: vehicle
Comment:

Model
Path: GenericVehicle.Examples.Vehicle
Comment:

Parameters
driver: driver(startTimeRampDrive=1)

Functional level 2
DIf: redeclare DriverInterface.JoystickWithYaw DIf — Driver Interface
Ch: redeclare GenericVehicle.Chassis.Chassis4W2D Ch — Chassis
AUX: redeclare AuxiliarySystems.AUXsimple AUX — Auxiliary systems
PS: redeclare GenericVehicle.PowerSupply.PS_FCbuff2 PS — Power Supply

redeclare GenericVehicle.PowerSupply.PS_FCbuff2 PS
PS_simple
PS_ICEISG
PS_ICEISG2
PS_FCbuff
PS_FCbuff2

Funct
VMC — Vehicle Motion Control
SC — Strategic Control
DIp — Driver Interpreter
EM: e MainControl.EnergyManagement.EM_simple EM — Energy Management

Figure 3: The generic vehicle menue easily allows changing the Power Supply.

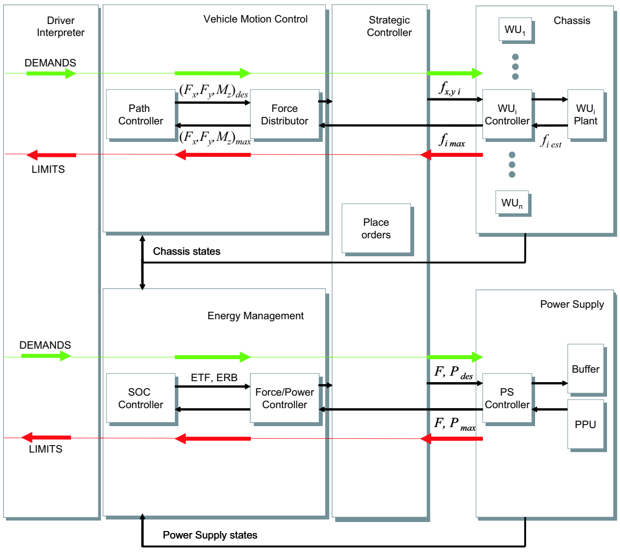


Figure 4: Signal between functional levels 1 and 2. Only signals to Power Supply and Chassis are shown for simplicity.

3 Modelica implementation

The Modelica implementation is gathered in the Modelica library `GenericVehicle`. According to Section 2 the main model consist of nine functions and in the library, these represent a sub-packages each. `DriverInterpreter`, `VehicleMotionControl`, `StrategicControl` and `EnergyManagement` cover the functional level 1. Additionally there are `DriverInterface`, `Chassis`, `PowerSupply` and `AuxiliarySystems` for level 2. Finally the `Bus` package contain the models necessary for the information exchange.

3.1 `DriverInterpreter`

The Driver Interpreter communicates with the driver interface by interpreting the driver's signals and by sending proper feed-back. The driver's intentions are interpreted as a desired path, taking into account limitations set up by the Vehicle Motion Control and external inputs such as e.g. cruise control. The desired path is defined by the velocity v, the vehicle's slip angle β, and the curvature ρ.

3.2 `VehicleMotionControl`

The Vehicle Motion Control includes a controller that follows the desired path by the derivation of desired global forces (F_x, F_y, M_z). These forces are then distributed between the wheels within the allowed limits for each wheel unit. Thus, there is an optimisation task and a control task. These are currently handled as described in [3].

3.3 `EnergyManagement`

Energy Management controls the energy flow from the Primary Power Unit (PPU) and the flow to the Buffer (Bf). The simple version of Energy Management calculates a State-Of-Charge (SOC) target by considering the vehicle speed, see Equation 1. By comparing SOC target with actual SOC simple strategies are used to calculate how much Electrical Regenerative Braking (ERB) and how much Electrical Traction Force (ETF) should be applied. Both parameters are nominal values. The desired tractive force and the total desired power needed from PS are the signals sent to Strategic Control which places the orders to Power Supply. EM Simple also calculates a power limit value for Auxiliary Systems.

$$SOCTarg = C_0 - C_1 \cdot e^{\frac{|v(t)|}{C_2}} \tag{1}$$

where $v(t)$ is the current vehicle speed, and $C_0 = 0.75$, $C_1 = 0.1$, and $C_2 = 6$ are constants.

3.4 `StrategicControl`

The Strategic Control is responsible for the commands from level 1 to level 2 and handles the priorities between VMC and EM. The simple SC only places the orders to functional level 2. Strategies about safety and reliability will be located at SC, checking the critical state signals from EM and VMC.

3.5 `DriverInterface`

The driver interface contains the actuators and sensors that the driver can influence. These could be steering wheel and pedals as well a joystick. DIf is here seen as a full drive by wire subsystem. The longitudinal, lateral and yaw signal are measured and then sent to DIp.

3.6 `Chassis`

The chassis (Ch) is thought of as a body onto which a number of wheel units are mounted. Each wheel is then considered as an autonomous unit and is by default decoupled from the other wheels. Depending on the linkage carrying the wheel as well as the available actuators, there are different possibilities to generate ground contact forces. A very simple example is a wheel with only brakes and no steering possibility and passive suspension, while other wheel units may have drive, steering, camber control and active damping.

The Modelica implementation is based on the `VehicleDynamics` library [4] components for three dimensional Multi Body System (MBS) chassis modelling. Additionally the `PlanarMultiBody` library [5] has been used to model simpler planar chassis models. The latter are suitable when influences of load transfer due to roll or pitch can be neglected since these models speeds up simulation time considerably.

The distributed forces from the SC is realised at each wheel unit that also sends information about maximum achievable force. For a future vehicle with independent wheel units, this is straightforward, but today's vehicles uses many passive components that in some case limits the wheel motion and also couples the wheels together. To deal with this, *restrictors* are introduced to limit the degrees of freedom (DOF) of the wheel.

3.6.1 `Wheel Units`

At each Wheel Unit (WU), the force commanded by the SC should be generated. To avoid saturation, the wheel unit provides the VMC with information about it current limitations. From the desired forces, the desired steering angle and wheel spin velocity are calculated [3].

To generate the wheel spin velocity, the wheel unit checks how much rotational torque is available directly at the drive shaft from the PS and then coordinates the available actuators to meet the desired order.

In Figure 5, three different WU models are shown, illustrating the variety of modelling detail. The left-

---

[3] Details are found in [3].

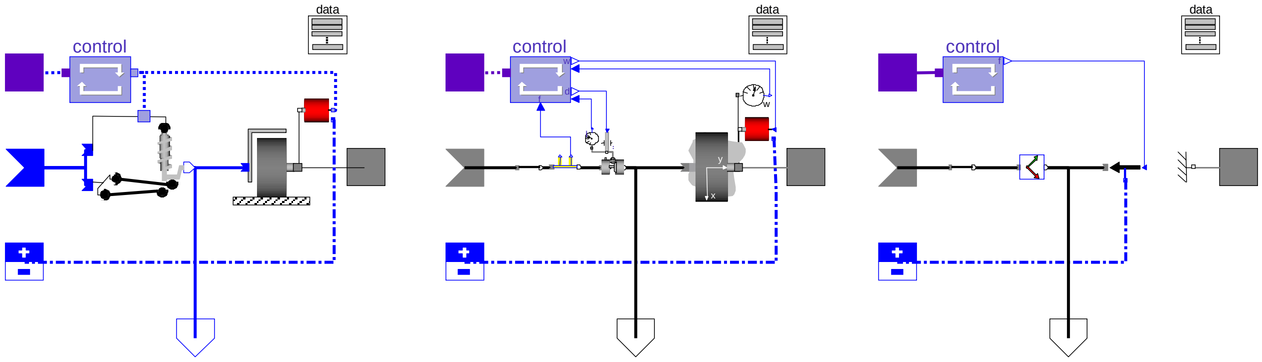


Figure 5: Wheel units with different level of detail. Left: A 3D MBS model of a control, linkage and wheel with an electric motor, middle: a 2D MBS model with linkage replaced by a steering joint, and right: an ideal Wheel Unit that generates the desired forces directly.

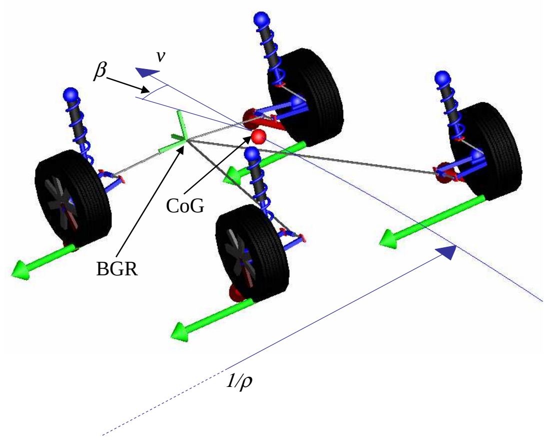


Figure 6: Screen shot of an animation showing a four wheeled HEV with independent wheel corners. The path (ρ, v, β) is indicated as well.

most example is a full 3D-model of a wheel and a linkage, e.g. a MacPherson or a DoubleWishBone. Here, a linkage suggested in [6] is used and an animation view of a vehicle with these wheel unit models can seen in Figure 6.

In many cases, when the details of the linkage are of less importance, simulation time can be reduced by using a simpler model as illustrated in Figure 5, middle. The linkage is reduce to an equivalent king-pin (steer) axis and no vertical motion is considered.

Still, these two models have in common the need to find steering angle and wheel spin velocity. The model in Figure 5, right, instead applies the desired forces directly.

3.6.2 Restrictors

As mentioned earlier, it is straightforward to use the WU concept as long at each wheel is independent of the others. This is not the case for today's vehicles and the restrictors are used to describe these relations. Typical restrictors are rack steerings and differentials that constrains the steering angle and the force distributed from the PS, respectively. To make the VMC aware of their existence, they are connected to the bus and send information about a) between what WU they act and b) how they act. Active restrictors also receive information about the WU state and commands to figure out how they should act. In Section 4, the usage of restrictors is exemplified.

3.6.3 Bodies

The body is the frame that carries the WUs. It also sends information about its states to the VMC. The reason it is treated as a separate unit, and not just as a least common divider of all chassis, is because there will be an extension that handles more than one body, coupled by restricors. Typical cases when this is relevant are tractor-trailer combinations, articulated buses and vehicles with a frame that cannot be considered as rigid.

3.7 PowerSupply

The conventional power train concept with a combustion engine, transmission, and drive line is not a valid description for a HEV. The HEV concept includes handling of a major electricity source in combination with a conventional or parts of a conventional power train. A more suitable name of this function is Power Supply. The PS includes both the Primary Power Unit and a buffer and can be anything from an internal combustion engine to a fuel cell. The buffer can be an electric buffer such as a battery, super capacitor or a mechanical one e.g. flywheel.

3.8 `AuxiliarySystems`

The Auxiliary systems is a gathering of all systems that are not involved in the vehicle's motion. Examples are air conditioning and lights. Aux calculates the actual power needed and sends this information to EM. EM limits the maximum power available for the Aux and PS provides the needed electricity by a standardised electrical connector.

3.9 `Bus`

The `Bus` contain generic information and orders that are exchanged between functional levels 1 and 2. The signals are named after their origin as exemplified below:

`EM_Pauxlimit` EM calculates a maximum power limit for Aux.

`SC_Pauxlimit` SC finalise the order to Aux.

`Aux_Pactual` The actual power consumption from Aux.

It is important that the signals are made hardware independent to allow easy change of functions. The Modelica implementation is based on the bus connectors available in the standard library. All models of a specific function e.g. EM, VMC, PS, and Ch share the same base, defining the `send` and `receive` signals.

The signals on the bus give an idea of what information is necessary for any kind of hardware configuration for the specific function.

4 Examples

To demonstrate the suggested architecture's ability to handle different hardware configurations, two different HEV configurations have been implemented. The first one is a parallel HEV with wheel motors on the front wheels, see Figure 7, left. As indicated in the figure, the front and rear wheels are constrained by restrictors. The front wheels have a rack steering that couples the steering angle of the two wheels. At the rear wheels there is also a rack steering, but in this case the range is set to 0, making the vehicle front wheel steered. Additionally, there is also a differential that distributes the driving torque from the PS.

The second case is a series HEV with wheel motors on all wheels, see Figure 7, right. Here no restrictors are used and each wheel is individually controlled.

The body weight and inertia for both cases is relevant for a sports utility vehicle. For both cases, the same models within functional level 1 are used.

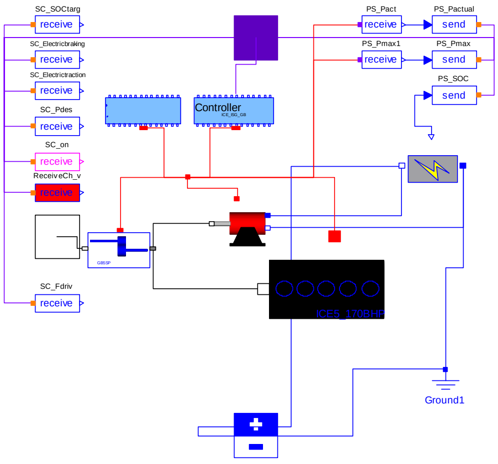


Figure 8: PS with ICE, ISG, and a GB.

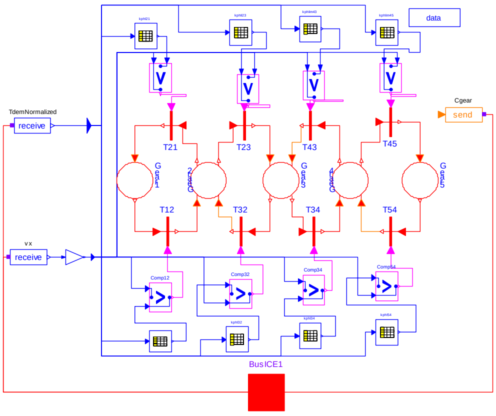


Figure 9: Petri net used for the shift strategy for the 5 speed gear box.

4.1 The parallel HEV case

The parallel HEV is equipped with PS containing an Internal Combustion Engine (ICE) as a PPU, and an Integrated Starter Generator (ISG), automated manual Gear Box (GB), and a battery, see Figure 8. There is a local controller that coordinates ICE, ISG and GB. The gearshift strategy is based upon a petri net which uses actual vehicle speed and desired torque for the boolean expressions. The petri net is shown in Figure 9.

The ICE model uses one dimensional look up tables for maximum and minimum torque. The fuel consumption is calculated by using the actual torque and rotational speed as input for a two dimensional look up table. The model is shown in Figure 10.

The Chassis contains wheel units with wheel motors for the front wheels and the rear wheels have a differential restrictor applying the torque provided by PS, see Figure 7, right. In this case the PS supplies both mechanical torque and electrical power to the chassis.

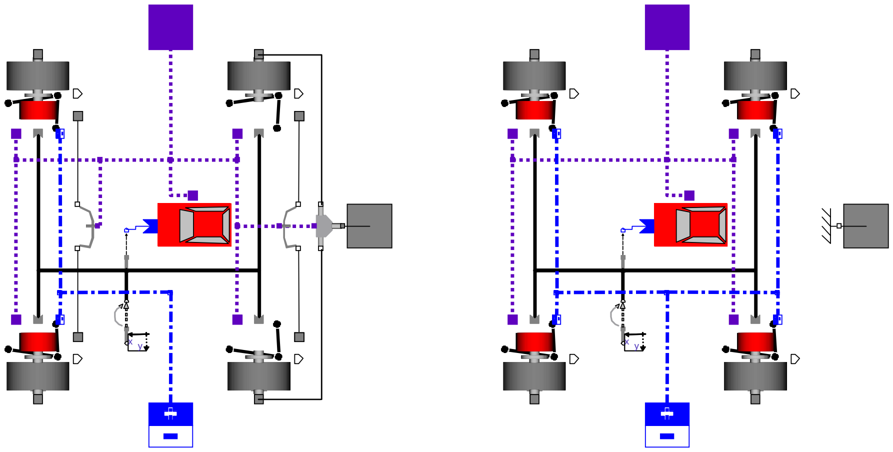


Figure 7: Chassis with independent wheel units used in the parallel HEV case, left and the serial HEV case, right.

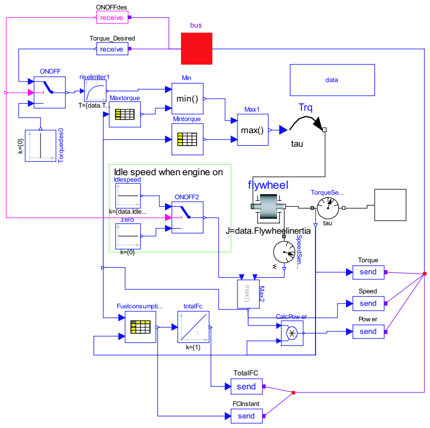


Figure 10: The ICE model.

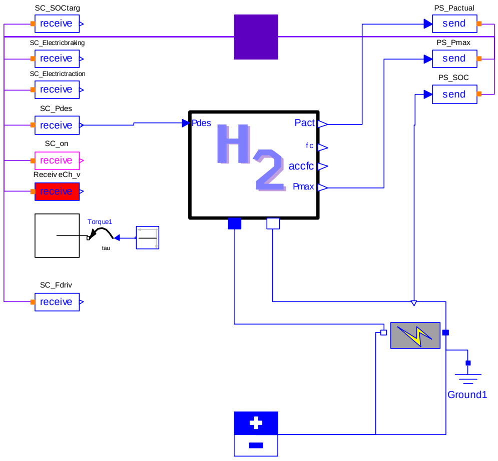


Figure 11: PS with FC and battery .

4.2 The serial HEV case

The Chassis contains wheel units with wheel motors on all four wheels. In this case PS submits only electric power to the chassis, see Figure 7, left. The PS contains a Fuel Cell (FC) as a PPU and a battery as buffer, see Figure 11.

4.3 Simulation

Figure 12 shows results from a ramp simulation of the parallel HEV vehicle starting from standstill. It is accelerated to 10 m/s in 4 s and then the velocity is kept constant for 1 s. Finally the velocity is decreased to stand still at $t = 8$ s. The first graph shows the desired speed from DIp and the actual speed. During the first 2 s of the deceleration the actual speed is higher than the desired. The second graph shows the actual torques from the ICE and ISG. Third graph shows the actual gear of the GB. Finally the fourth graph shows the SOC level of the battery.

The same simulation is also made for the serial HEV configuration, see Figure 13. The first graph shows the desired speed from DIp and the actual speed. The second graph shows the desired power and the generated power from FC. The third graph shows the actual SOC level of the battery. The fourth graph shows the instant and accumulated fuel consumption. During deceleration the FC is shut down.

The results show that it is possible to use the same VMC, EM, DIp and SC for both configurations. The performance of the models are not optimal since the scope of work have not been on sizing on components nor to find the optimal strategies.

5 Conclusions and discussion

Modelica has been a useful way to describe, model and test the architecture. It is a good platform because it allows easy interaction of different domains such as

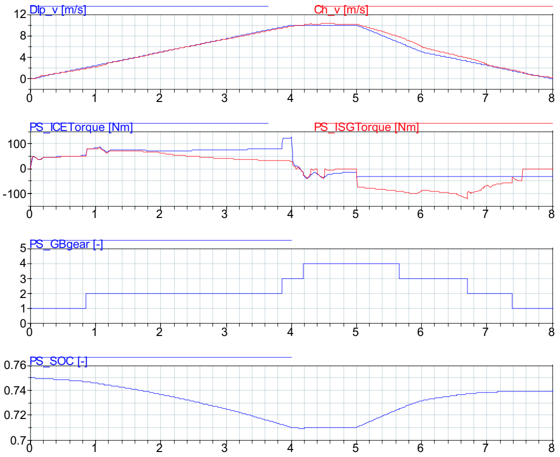


Figure 12: Ramp simulation for the parallel HEV configuration.

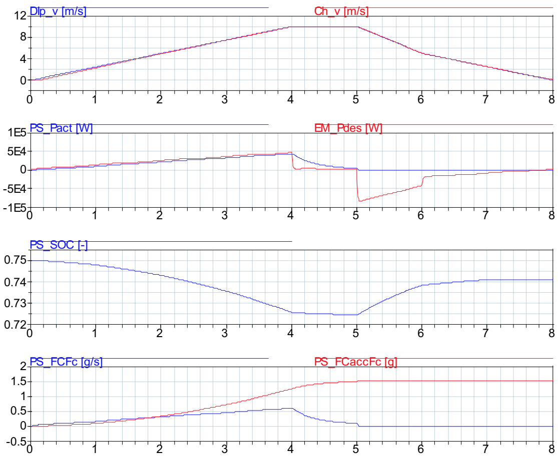


Figure 13: Ramp simulation for the serial HEV configuration.

multibody, electrical, mechanical, and control.

The sample cases demonstrates the architecture. The results show that the architecture manage different hardware configurations and that exchanging hardware does not affect the highest functional level, i.e. Main Control.

Even though the over-all impression is positive, some limitations have been found. The size of the bus is dependent on the number of wheel units, bodies and restrictors and should thus be defined by the chassis itself. Since the size of the bus must be fixed, this is currently not possible. It would also be desirable to be able to send the equations defining the restrictors directly through the bus.

6 Future work

An extension of this work will mainly involve a) Development of a method to evaluate the reusability and constraints applied by using the suggested architecture. b) Verification of the reusability of the suggested architecture for different configurations of HEVs. Especially different configurations of PS. c) Studies on what control strategies within Main Control would apply for the foreseen HEV configurations. d) Studies on how critical states could be handled so that they are recognised by EM and/or VMC. e) More flexible description of restrictor information from functional level 2 to 1. f) Compensation for non-ideal sensors.

7 Acknowledgements

This work is financed by the Driving Dynamics and Main Control projects within the Swedish National Research Programme ”The Green Vehicle/FCHEV”.

References

[1] Modelica Association. http://www.modelica.org.

[2] L. Laine and J. Andreasson. Generic control architecture applied to a hybrid electric sports utility vehicle. *To be presented at the 20th International Electric Vehicle Symposium, Long Beach, CA, November 15-19*, 2003.

[3] J. Andreasson and L. Laine. Driving dynamics for hybrid electric vehicles considering handling and control architecture. *18th Int. Symp. IAVSD, Dynamics of Vehicles on Roads and Tracks, Japan*, August 2003.

[4] J. Andreasson. VehicleDynamics library. In Peter Fritzson, editor, *Proceedings of the 3rd International Modelica Conference*, Linköping, November 2003. The Modelica Association and Linköping University.

[5] J. Andreasson and J. Jarlmark. Modularised tyre modelling in Modelica. In Peter Fritzson, editor, *Proceedings of the 2nd International Modelica Conference*, Oberpfaffenhofen, March 2002. The Modelica Association and Deutches Zentrum für Luft- und Raumfahrt.

[6] S. Zetterstrom. Electromechanical steering, suspension, drive and brake modules. *VTC 2002-Fall,Vancouver, Canada, September 24-28*, 2002.