

Hardware-in-the-Loop simulation of a hybrid electric vehicle using Modelica[®]/Dymola[®]

Dietmar Winkler, Clemens Gühmann
Technical University of Berlin

Abstract

In this paper the modelling of a hybrid electric vehicle using the object-oriented modelling language Modelica[®] will be presented. The main focus shall be the real-time simulation in connection with a Hardware-in-the-Loop (HiL) test-bench system. The paper presents the main components of a hybrid electric vehicle (HEV) using the Micro Hybrid architecture. Micro Hybrids allow the use of Start/Stop operation as well as brake energy regeneration. As one example out of the complex tasks of the hybrid controller a control strategy for the battery's state of charge using a sensitivity function will be explained. To show the impact of using such a sensitivity function the results of an simulation are presented in the end.

Keywords: *Hybrid Electric Vehicles, power train, modelling & simulation, real-time, test-bench*

1 Introduction

The current development process of automobiles is shaped by the different demands of the legislator, the manufacturer and the customer.

The main interest of the legislator lays hereby in lower exhaust-gas emissions whereas the customer wants a car with a low fuel consumption. Unfortunately a low fuel consumption does not automatically yield to low exhaust-gas emissions and vice versa. In addition to demands getting ever harder to meet, the development times decrease more and more. But decreasing development times ask for earlier tests of vehicle components especially of components of the power train. In the past therefore a new trend of developing philosophy was being followed by the car manufactures. This new trend is often reverred to as: "From Road to Roller to Rig". Which means that tests which were carried out on expensive test drives with real vehicles on the road are shifted to a roller test-bench. But because you still need a car (which is most likely not available in an early phase of development) these tests are shifted to an engine or transmission test-bench system (*rig*). On such an engine test-bench system a calibration of the engine's electrical control unit (*ECU*) can be carried out. This is normally done for stationary operation points only. If one would like to perform such a ECU calibration also for dynamic engine operations certain conditions have to be met. One is that engine test-bench system must have a highly dynamic operation range. Another being that a load which is normally applied to the engine during a test cycle has to be supplied as a control signal in real-time to the test-bench system. This means that it is necessary to set up a test-bench system together with a real-time computer. Such a highly dynamic engine test-bench system can be used for HiL simulations of any drive train model which provides suitable interface signals. Thus the use of engine test-bench systems for the development of new HEV drive trains and architectures is an interesting and challenging opportunity.

2 Use of simulation in the HEV development process

In the traditional development of HEVs offline simulations are used within the system specification phase to estimate the potentials of possible parameter sets. This results in a tuning of components (e.g., in terms of power and torque rating) and a rough guideline for the various control loops. The most relevant items for this are the engine and transmission control and the higher-level operating strategy. The operating strategy controls the interaction of the electric motor(s) and the combustion engine. Depending on the driving state and the driving task a specific gear is selected and the torque distribution is determined.

In the next step of function specification it follows a more detailed description of the control loops and operating strategy. These offline simulations used within the early phases of the development process are often called Model-in-the-Loop simulations (MiL). It exist a model of the function of the to be used electronic control unit (ECU). For the modelling of the vehicle and functions standard tools such as *MATLAB*[®]/*Simulink*[®]¹ and Modelica[®]² in connection with Dymola[®]³ are used. Offline simulations of HEVs can be done for example by using the *MATLAB*[®]/*Simulink*[®] based tool ADVISOR (e.g.,[4]).

In the following software development the phases module test and integration or system test are accompanied by by MiL or Software-in-the-Loop (SiL) simulations. These established simulation methods increase the overall test coverage. In the SiL simulation the *functional* model of a electronic control unit is replaced by C-code. With this coding errors can be found independent of the future ECU hardware. The next two steps consist of a verification of the interaction of the different modules (integration test) and the system test. For this tests the actual hardware of the ECU is available and the tests can be supported by HiL simulation. But in order to use the HiL simulation we need real-time capable simulation models. Developing these is a real challenge since the models have to be accurate and fast enough at the same time.

After the software tests are successfully passed the calibration of the ECUs can be done on the test-bench or in the vehicle. Only now reliable statements to about the exhaust-gas emissions and the expected fuel consumption can be obtained. At this point changes to the functions and system specifications are time consuming, expensive, and in most cases not possible.

The use of HiL simulations in connection with engine test-benches allows us to obtain very accurate predictions if the exhaust-gas emissions and the fuel consumption in an early phase of the development process. Only the size and type of the combustion engine must have been decided on.

3 Test-bench system

Because of a cooperation of our Department of Electronic Measurement and Diagnostic Technology of the Technical University Berlin with the IAV GmbH Berlin, our department has access to a highly dynamic engine test-bench system. This test-bench system is used in connection with the model based calibration of electronic control units (*ECU*) of engines and transmissions [9].

The setup of the project called “Test-bench of the Future” is depicted in Figure 1. The highly dynamic engine test-bench system consists of a combustion engine which is directly coupled with an electric *Dynamometer* which in turn is controlled by a power electronic converter (*Drive*). In addition to that a *Control System* is needed to supply the needed control signals (e.g. Pedal Value Source α for the engine, *Dynamometer Control* for the *Drive*) and to acquire the measurement data (e.g. speed and torque signals of the shaft). All this is done by the physical input/output cards (i.e. analogue/digital and digital/analogue cards). Connected with the *Control System* is the *HiL Simulator* (a standard PC running the real-time operating system *QNX*⁴).

¹MATLAB and Simulink are registered trademarks of The MathWorks, Inc. (→www.mathworks.com)

²Modelica is a registered trademark of the Modelica Association (→www.modelica.org)

³Dymola is a registered trademark of Dynasim AB (→www.dynasim.se)

⁴QNX is a registered trademark of QNX Software Systems (→www.qnx.com)

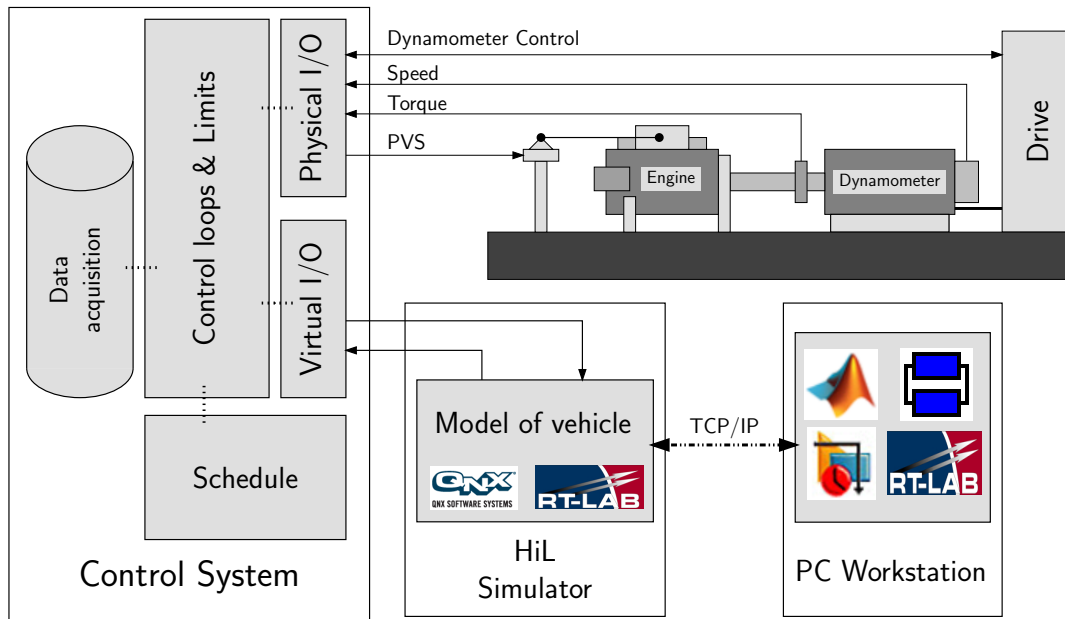


Figure 1: Diagram of the “Test-bench of the Future”

Due to the *direct* coupling of the dynamometer with the combustion engine, the clutch and the transmission have to be modelled in addition to the remainder of the vehicle’s drive train (e.g. train resistance and so forth). This also means that the simulation model of the power train is not restricted to a classic car concept but can also represent the power train of a hybrid electric vehicle. The type and size of the combustion engine is fixed by the test-bench system. However the rest of the power train consisting of clutches, transmissions and electric motors/generators is completely exchangeable within the simulation. So it is possible to investigate different concepts (e.g., parallel, series, and split power train) and component sizes of the hybrid electric vehicle power train in respect to fuel consumption and exhaust-gas emissions.

4 Simulation model

4.1 Simulation environment

Because a power train model contains different signals from different physical domains (e.g. electrical, mechanical, thermal, etc.) a simulation tool is needed which supports such models. The modelling language Modelica[®] was especially developed to meet the needs of different types of physical domains. Modelica[®] is object oriented which gives the developer the option of reuse of model parts which in return shortens the model development process (for more on Modelica[®] see [2], [7], and [3]). With the help of Dymola[®] the Modelica[®] simulation models can be edited, compiled and simulated. These Dymola[®] models are then integrated in a MATLAB[®]/Simulink[®] model and processed with RT-LAB^{®5} for real-time computation. For more details on the application of Modelica[®] real-time simulation models see [1] and [8].

4.2 Example: Starter/Generator topology

There are different kinds of HEVs architectures. The main categories are Micro, Mild and Full Hybrid. One kind of the *Micro Hybrids* is a HEV with a so-called Starter/Generator unit. This unit can be

⁵RT-LAB is a registered trademark of Opal-RT (→www.opal-rt.com)

placed in various places (e.g., transmission mounted generator, belt-coupled generator, etc.). For this paper we choose an example that uses a Starter/Generator unit which is placed between the engine and the transmission (with one clutch on each side) so the conventional claw-pole generator is not required anymore.

The setup provides the following advantages:

- Impulse-start ability with a high start torque with low power requirements
- Support for synchronisation of gears during gear shifts
- Brake energy regeneration

There are of course also drawbacks like for example the high costs for the two clutches. A principle setup is depicted in Fig. 2

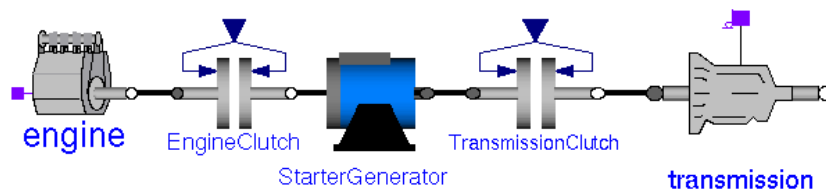


Figure 2: Principle setup of the used Starter/Generator topology

Even more other possibilities of a Starter/Generator topology exist (e.g., transmission mounted generator, belt-coupled generator, etc.) but shall not be explained in this paper.

4.3 The complete drive line model

The developed Modelica[®] model of the drive line uses the free *Modelica Standard Library* (MSL version 2.2.1 [5]) and the commercial available *Power Train Library* [6].

The complete drive line model consists of the following components (see Fig. 3):

1. Driver (including the driving cycle)
2. HiL Engine (The engine is not simulated since the real engine of the test-bench system is used for the HiL simulation)
3. Starter/Generator (permanent magnet synchronous machine)
4. Gear- and Transmission clutches
5. Hybrid controller (for controlling the clutches, the gear shifting, the battery's SOC, etc.)
6. FOC unit (Field Oriented Control unit provides the correct voltage levels for the Starter/Generator)
7. Transmission (6-gear automated shift transmission)
8. Axle (model of the final drive and the wheels)
9. Car (includes all the driving resistances of the vehicle)

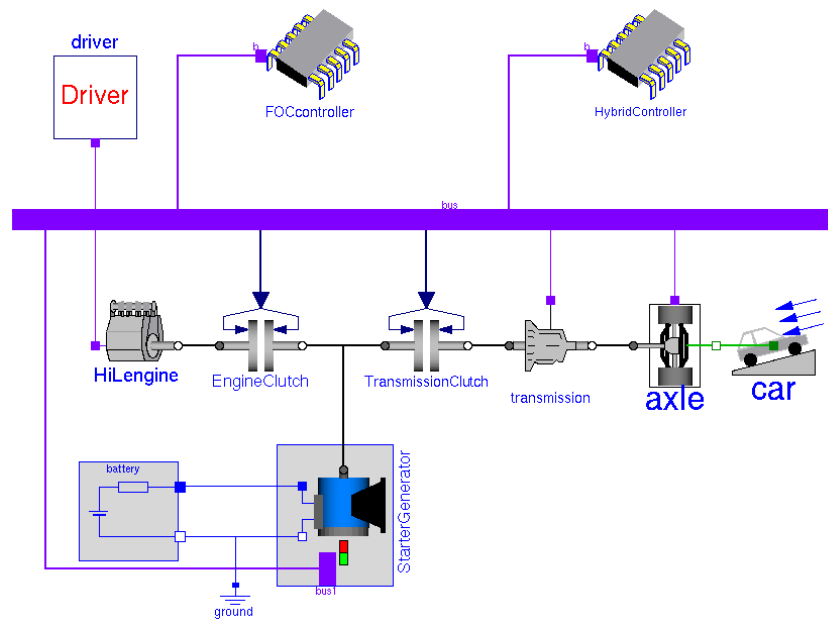


Figure 3: Drive line model of the HEV

5 Operation strategy

Finding an optimal operation strategy for the HEV is a very complicated and complex process. One task of the hybrid controller is the minimisation of the fuel consumption. This means to operate the combustion engine *and* the Starter/Generator in an overall area of maximum efficiency.

We want to explain two main aspects of the operation strategy here. One is the adaptation of the engines load level and the other the brake energy regeneration.

5.1 Adaptation of engine load level

With the help of the Starter/Generator it is possible to either increase the load of the engine by charging the battery (i.e., by increasing the engines load level to one with a higher efficiency in respect to fuel consumption per *kWh*) or to decrease the load on the engine (i.e., electrical boosting). The tricky thing now is to decide when does the battery need recharging and when can the battery be discharged without the risk of loosing too much battery power. The energy efficiency of the engine has to be put in relation to the electrical power that can be drawn out of the increased engine load level. In a nutshell, how many grams of fuel per *kWh* does it cost to charge the battery at certain operation points. Or the other way round, how much fuel can we save when using the stored battery energy at another point of operation (e.g., typically at low speeds).

The hybrid controller needs to account for *all* losses connected with the charging *and* discharging process of the battery. Examples of electrical losses are:

- Inverter losses (power dependent)
- Battery losses (dependent on battery type, SOC (state of charge), temperature)
- Electrical machine losses (copper and iron losses)

One very important factor is also the SOC of the battery. There is no point in trying to charge the battery when its SOC is already at the allowed maximum boundary. Taking into account the electrical and mechanical losses in connection with the SOC and the current fuel efficiency of the engine the hybrid

controller can then decide if the load level shall be increased, decreased, or left as it is. To give real-time compatibility all losses are stored as characteristic maps keeping the computational effort at a minimum.

5.2 Brake energy regeneration

The other interesting fuel saving potential is the use of brake energy regeneration. Depending on the amount of braking power needed the hybrid controller decides if the Starter/Generator can provide that amount of braking power or if the disk brakes have to be activated in addition. In contrast to the load level adaptation the battery will always be charged if brake energy is available as long as the maximum SOC is not reached.

5.3 Controlling the state of charge (SOC)

In order to minimise the overall losses of the power train the operation strategy only looks for the minimum loss path. For small torque that would mean that the HEV would preferable be driven by the electric motor only. This would lead to a complete discharge of the battery which is hardly desirable. So an optimum hybrid controller must be able to adapt its strategy according to the current state of charge of the battery. This is done with the help of sensitivity curves.

5.3.1 Sensitivity curve

The hybrid controller should keep the SOC of the battery within certain boundaries during operation. The operation strategy should be changed when the SOC is coming close an upper or lower boundary. The goal is to minimise the additional losses caused by the charging or discharging process. To account for these additional losses a sensitivity unit is introduced.

The sensitivity S can be calculated by:

$$S = \frac{\Delta P_{\text{loss}}}{\Delta I_{\text{batt}}} \quad (1)$$

The control of the battery's SOC is now done by using a sensitivity curve S_{curve} which is projected on top of the SOC interval. This curve defines how much losses we are prepared to accept in order to charge/discharge the battery. If for example the battery's SOC is at the optimum operation point (e.g., SOC= 70%) then the value of the sensitivity curve is zero and the loss minimisation strategy is used. However the more the SOC differs from the optimum the more losses will be accepted. So the operation strategy is only modified if the expected losses are lower than the according sensitivity value S_{curve} . Fig. 4 shows an example of a linear sensitivity curve with sensitivity maxima of $-50W/A$ and $+50W/A$ for a SOC of 60% and 80% respectively.

The sensitivity curve can also consist of two branches. One for the account for charging losses and one for the discharging losses. Also the gradient and curvature can vary. A higher curvature forces the hybrid controller to allow fewer losses around the optimal SOC operating point.

Example In Figure 5 an example for several sensitivity curves S_{curve} with different curvatures is given. The optimal SOC operating point is set to $SOC_{\text{opt}} = 70\%$ and the allowed operation interval of the battery's SOC is set between 50% and 85%. The maximum sensitivity for charging is $S_{\text{charge}_{\text{max}}} = 40W/A$ and the maximum sensitivity for discharging is $S_{\text{discharge}_{\text{max}}} = 60W/A$.

For the curvatures different potentials of the function:

$$y = m \cdot x^p \quad (2)$$

are depicted.

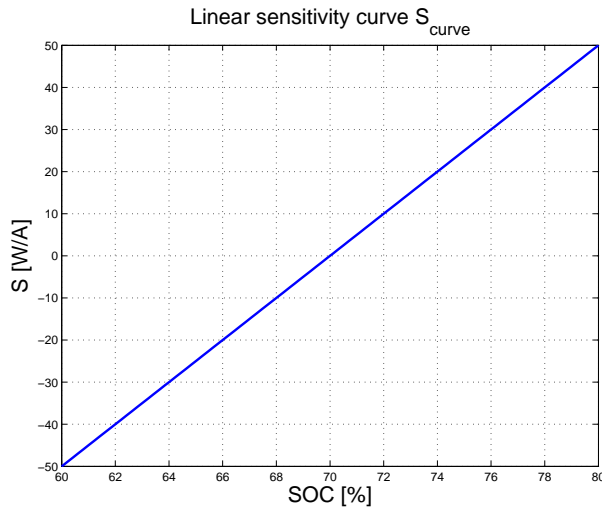


Figure 4: A linear sensitivity curve

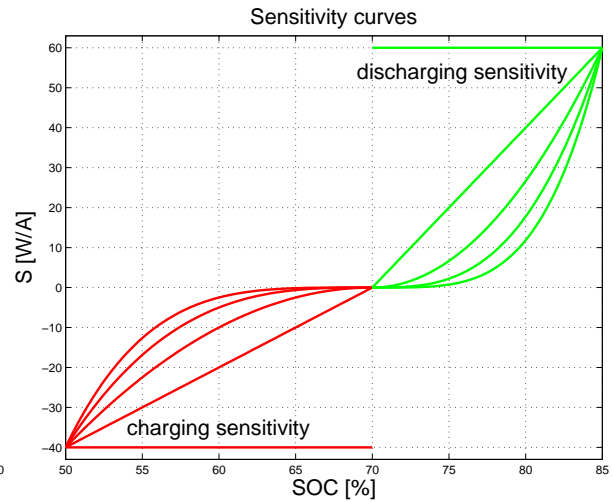


Figure 5: Sensitivity curves for $p = \{0, 1, 2, 3, 4\}$

6 Hardware-in-the-Loop simulation

The finalised HEV drive line model needs to be modified in order to be used with the HiL system (see section 4.1). The main signals with which the engine test-bench is controlled are hereby the pedal value source α , the drive line torque momentum M and the drive line speed n . To understand the way the test-bench system interacts with the real-time simulation model (i.e., the HEV model) we have to explain the used control strategies.

6.1 Control strategies

There are two main control strategies used in connection with the test-bench system.

1. $M - \alpha$ -control (i.e., control by torque momentum and pedal value source)
2. $n - \alpha$ -control (i.e., control by speed and pedal value source)

For both control strategies the engine is controlled with the pedal value source α (see again Fig. 1). In the first case, the $M - \alpha$ -control, the dynamometer is controlled by a torque set point. The resulting engine speed is then fed back to the drive line model as input.

The second case, the $n - \alpha$ -control, the dynamometer is controlled by a speed set point. This time the resulting engine torque is measured and again fed back to the HEV drive line model as input. Depending on the operational mode one control strategy is more applicable than the other. In case of the HEV drive line we need the $M - \alpha$ -control whenever the `EngineClutch` (see again Fig. 3) is opened. This is because at this state the HEV drive line model has no information whatsoever on what the speed of the engine side of the `EngineClutch` is. The one thing we do know is that the torque is near zero during that state.

For all other operational states the $n - \alpha$ -control strategy is preferred since the output torque of the engine is always unique for a certain speed. In contrast to that setting the engine load to a certain torque set-point can produce more than one speed settling point. This can lead to instabilities or safety problems of the test-bench system (i.e., maximum speed is violated or such likes).

6.2 Precautions for impulse start

When we tried to prepare our model to interact with the test-bench system we came across challenge of implementing the impulse start. At engine stand-still the test-bench system does not accept a torque command as set-point. Therefore the engine needs to be started using the $n - \alpha$ -control and then the control strategy needs to be instantly switched to $M - \alpha$ -control so that engine can reach idle-speed. So this has to be taken into account when modelling the impulse start functionality for the test-bench HiL simulation.

7 Simulation results

One interesting aspect is the impact of the sensitivity function on the battery's SOC. To investigate this impact we simulated the HEV model using the New European Driving Cycle. The top part of Fig. 6 shows the course of the vehicle speed over the time. The middle part shows the calculated sensitivity value at the corresponding time instant for a linear ($p = 1$) and a quadratic ($p = 2$) sensitivity curvature. And finally in the lower part the corresponding battery's SOC in percent is shown.

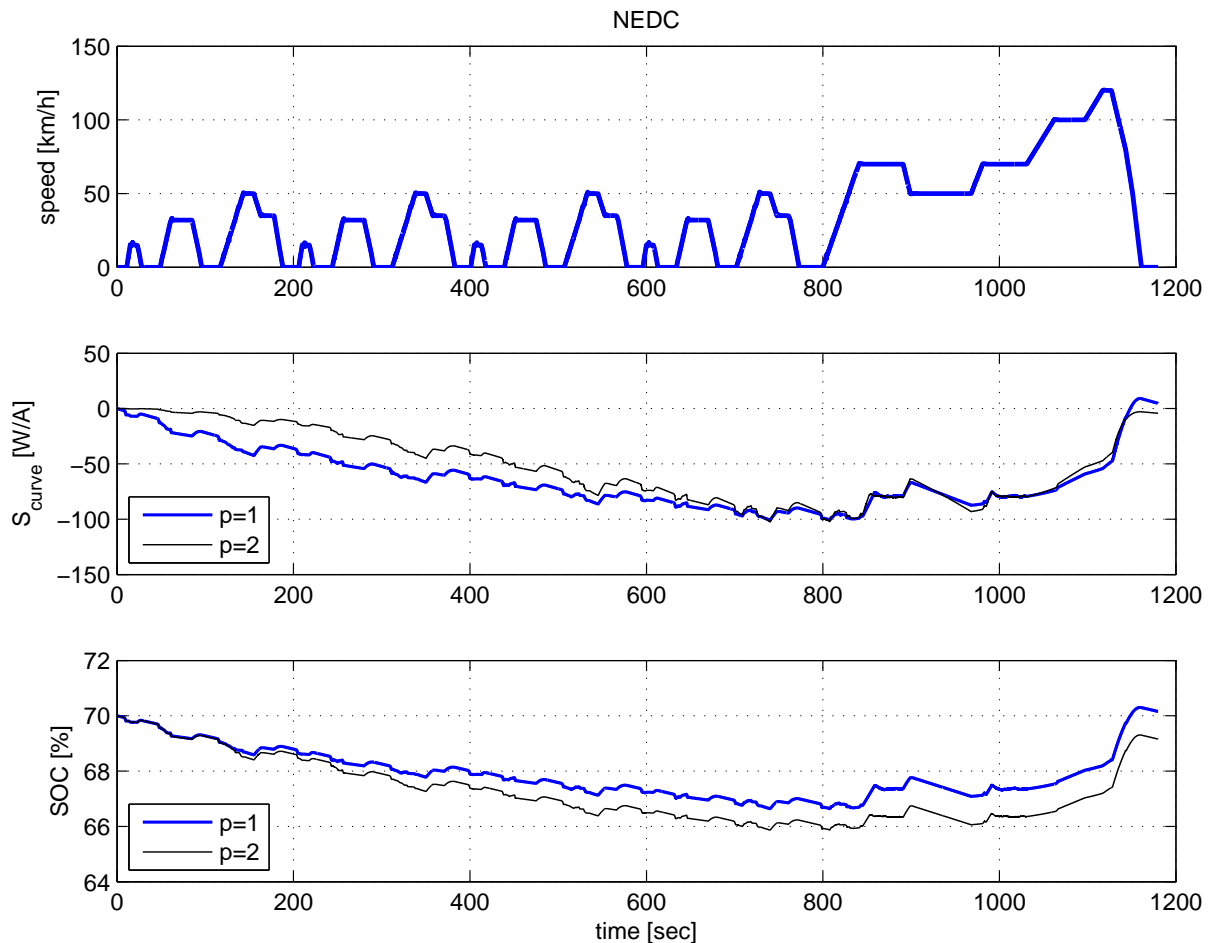


Figure 6: Simulation of NEDC using different sensitivity curves

It is worth noticing that by using the quadratic curvature the battery is not as quickly discharged. This is because of when the hybrid controller has to decide on whether or not to charge the battery the quadratic curvature will not allow as much losses as the linear sensitivity function does. The hybrid controller

will therefore hold on to the loss minimisation strategy even longer as it would be in case of the linear sensitivity curve.

8 Summary

The paper showed a brief overview of the simulation of a HEV using the object-oriented modelling language Modelica[®]. Some of the many issues to be considered when building up model for a HEV were discussed. The model in this paper is used to communicate with a HiL test-bench system. Thus further modifications had to be applied to the model in order to enable HiL interaction. Especially the impulse start ability is a challenge on both the model control loops and the test-bench control loops. At the end some simulation results using the NEDC cycle were presented to show the impact of the used sensitivity curves for control of the battery's SOC.

The future work of our department will include the modelling of the electric drives in more detail. With such more detailed motor models investigations the fault behaviour (e.g., electric shorts within one phase winding) of the electric drives and their influence on the drive line shall be carried out.

References

- [1] H. Elmqvist, S. Mattsson, H. Olsson, J. Andreasson, M. Otter, C. Schweiger, and D. Brück. Real-time Simulation of Detailed Automotive Models. In *Proceedings of the 3rd International Modelica Conference*, pages 29 – 38, 2003.
- [2] H. Elmqvist, S. Mattsson, and M. Otter. Modelica-a language for physical system modeling, visualization and interaction. In *Computer Aided Control System Design, 1999. Proceedings of the 1999 IEEE International Symposium on*, pages 630–639, 22-27 Aug. 1999.
- [3] P. A. Fritzson. *Principles of Object-Oriented Modeling and Simulation with Modelica 2.1*. John Wiley & Sons, Januar 2004.
- [4] T. Markel and K. Wipke. Modeling grid-connected hybrid electric vehicles using ADVISOR. In *Applications and Advances, 2001. The Sixteenth Annual Battery Conference on*, pages 23–29, 9-12 Jan. 2001.
- [5] Modelica Association. *Modelica[®] - A Unified Object-Oriented Language for Physical Systems Modeling – Language Specification*, version 2.2 edition, 2nd February 2005.
- [6] M. Otter, C. Schweiger, and M. Dempsey. *PowerTrain Library 1.0*. German Aerospace Center (DLR), Oberpfaffenhofen, 1.0 edition, 2002.
- [7] M. Tiller. *Introduction to Physical Modeling with Modelica*. Kluwer International Series in Engineering & Computer Science. Kluwer Academic Publishers, 2001.
- [8] D. Winkler and C. Gühmann. Synchronising a Modelica Real-Time Simulation Model with a Highly Dynamic Engine Test-Bench System. In *5th International Modelica Conference*, Vienna, 4-5 Sep 2006. Modelica Association.
- [9] D. Winkler, C. Gühmann, B. Barzantny, and M. Lindemann. Model Based Calibration of ECUs Using a Highly Dynamic HiL Test Bench System. In *Design of Experiments (DoE) in Engine Development II*. expert verlag, 2005.

Authors



Dipl. Ing. Dietmar Winkler studied Electrical Engineering at the Technical University of Berlin and specialised in the field of control of electric drives. Since January 2004 he works as a research and teaching assistant and doing his PhD at the Department of Electronic Measurement and Diagnostic Technology at the TU Berlin.

address: TU Berlin, Sekr. EN 13, Einsteinufer 17, D-10587 Berlin

phone: +49 30 314-24501, fax: +49 30 314-22120

email: Dietmar.Winkler@tu-berlin.de



Prof. Dr.-Ing. Clemens Gühmann is head of the Department of Electronic Measurement and Diagnostic Technology at the Technical University of Berlin. The departments research focuses on: Modern Data Processing Methods for Automotive Systems; Simulation, Test, and Calibration Methods for Automotive ECUs (HiL, MiL, SiL-Simulation); Modelling and Real-Time Simulation of Automotive Systems; Pattern Recognition and Technical Diagnosis; and Hybrid Vehicle Control Strategies.

address: TU Berlin, Sekr. EN 13, Einsteinufer 17, D-10587 Berlin

phone: +49 30 314-29393, fax: +49 30 314-25717

email: Clemens.Guehmann@tu-berlin.de