Proceedings
of the 4th International Modelica Conference,
Hamburg, March 7-8, 2005,
Gerhard Schmitz (editor)

M. Hommel
Volkswagen AG, Germany

First Results in Cluster Simulation of Alternative Automotive Drive Trains

pp. 201-209

First Results in Cluster Simulation of Alternative Automotive Drive Trains

Mathias Hommel
Brieffach 1778
Volkswagen AG
D-38436 Wolfsburg, Germany

Abstract

Control software plays an important role in the development of alternative drive trains. Energy management intervenes with the control of the combustion engine, the transmission or an additional electrical machine in different ways. In order to develop the energy management before or parallel to the vehicle construction phase, a complex software development process is required that equally supports modeling, simulation and implementation.

In the R&D of Volkswagen cluster simulation was established to simulate the drive train of a vehicle as well as to develop algorithms for the relevant electronic control units (ECU).

The methodology of cluster simulation will be represented in the following article.

1 Introduction

For more than 20 years now Volkswagens deals with alternative concepts for automotive drive trains. First there were the electric vehicles with a comparatively simple control that converted the driver's wish into a driving torque. Today, however, as combustion engine and electrical machine can be connected with each other in multiple ways, one needs a complex control in order to influence the torque distribution depending on the driving conditions. The intention in this case is to positively influence comfort and driving capability.

The control of the different components of the drive train plays a central role in this context. The so-called energy management coordinates the torques of the drive train. The electrical energy storage capacitor is controlled by the so-called battery management and so forth. For example, energy management and battery management influence each other in a complex way.

The simulation of the drive train plays an essential role during the specification of the components as well as during modeling the control algorithms. The objective of this drive train simulation is a fast and manageable process for developing controller algorithms resulting in an automatic code generation within a software-in-the-loop (SIL) process. Manageable in this context means that the developer can react in a quick way to altering structures of the drive train. The simulation time should be faster than real time in order to be able to carry out parameter variations and code development fast. Therefore a cluster computer was built for the drive train simulation consisting of ten dual processor computers. The individual computers themselves are connected with each other with a Myrinet[1] network, which is an optical network.

Within the project SUVA (Surplus Value Hybrid Vehicle) that was supported through the European Community [1] Volkswagen built up a Volkswagen Bora Hybrid with a hybrid drive train (Figure 1).

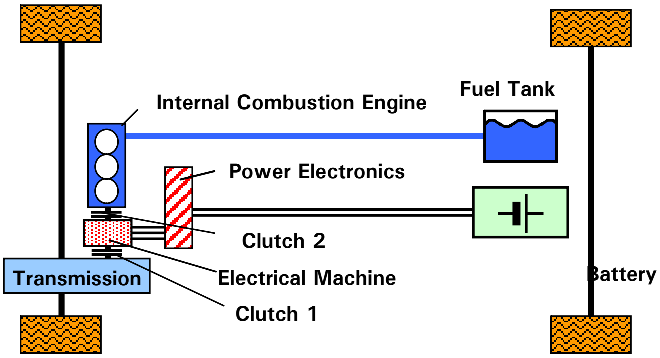


Figure 1: Structure of the Volkswagen Bora Hybrid

The drive train of the Volkswagen Bora Hybrid consists of a combustion engine (1.4 liters, 55 kW, 3 cylinders, turbo diesel), an electrical machine

---

[1] Myrinet is a registered trademark of Myricom, Inc., USA, http://www.myricom.com

(asynchronous machine, 25 kW peak), a clutch between the combustion engine and the electrical motor (clutch No. 2 in Figure 1), the automatic transmission (a dual clutch transmission, named by Volkswagen DSG®[2], [2]) and an energy storage (6 Ah NiMH-battery, 288V).

The transmission concept is such that the transmission is provided with a dual clutch on the gearbox input side (see Fig. 6 in chapter 2.1 as well). This clutch is represented in Figure 1 as being outside of the transmission for the sake of simplification (designated as clutch 1 in Fig. 1).

Due to this arrangement this specific drive train is called a parallel hybrid drive train since both combustion engine and electrical machine simultaneously or separately supply torque to the entire driving torque of the vehicle - acceleration to the strategy that is worked out in the above mentioned energy management ECU. For the classification of the different hybrid vehicles please refer to the relevant literature [3].

2 Simulation Model

The differences of the block-oriented or causal modeling using for example Matlab/Simulink[3] and the acausal modeling using for example Dymola[4] were described sufficiently [4].

While in the development of ECU algorithms the causal, graphical, signal-based modeling become more and more accepted in prototyping, acausal modeling has its advantages in the description of physical systems. The physical structure is maintained and the description corresponds to the local physical equations of the components that are independent of their environment, as well as their coupling to the entire system of equations.

For this reason the cluster simulation is realized by a simulator link-up: On the one hand Matlab/Simulink is used for modeling the ECU algorithms, the driver model (which generates the accelerator and brake pedal) and the driving cycle (which generates the reference vehicle speed value, height, air pressure and so on).

On the other hand Modelica[5]/Dymola is used for modeling the plant that is the closed loop controlled system vehicle. Furthermore, executable files of ECU algorithms in the form of a DLL (dynamic link library) are incorporated into the simulation (see Fig. 2). As it is shown in Figure 2 in the cluster simulation the essential algorithms of the relevant ECUs are simulated in Matlab/Simulink such as:

- the internal combustion engine ECU (ICE Controller[6]),
- the ABS/ASC-system (Brake Controller),
- ECU of the gearbox (Gearbox Controller) as well as
- the ECU of the electric machine (E-Machine Controller).

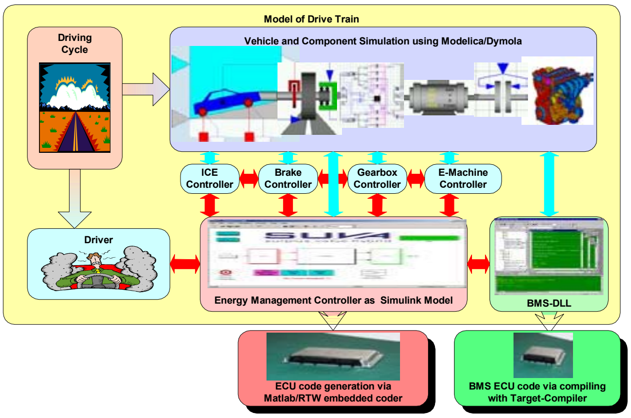


Figure 2: Structure of the drive train model of the cluster simulation

Furthermore, the controller of the energy management ECU which is modeled in Matlab/Simulink is integrated as well as the DLL of the controller of the battery management ECU (BMS, battery management system).

The plant was modeled in Modelica/Dymola as already described above and can be linked to the cluster simulation either as a Dymosim.exe or as a Dymola model.

Executable files (so-called executables or exe) were generated from all models since it is to be expected that through the detailed modeling the performance of the cluster simulation is lowered due to simulation of uncompiled models.

---

[2] DSG is a registered trademark of Volkswagen AG, Germany

[3] Matlab und Simulink are registered trademarks of The Mathworks, Inc., USA

[4] Dymola is a registered trademark of Dynasim AB, Sweden

[5] Modelica is a registered trademark of the Modelica Association

[6] The term controller synonymously stands for a closed loop control algorithm that is the functional software of an ECU (electronic control unit).

The cluster simulation is a so-called forward simulation in contrast to the so-called backward simulation. Starting from a driving cycle the comparison of the actual value of the vehicle velocity with the reference value is done by the driver model which then generates the accelerator or brake pedal command. The strategy of the energy management then controls the components of the drive train to generate the necessary driving torques in order to follow the reference value of the vehicle velocity of the driving cycle. The actual value of the vehicle velocity is traced back to the driver model.

In the following, the individual, modeled systems are shortly described as well as the structured components library in Dymola and the simulator coupling.

2.1 Model of the Plant

The vehicle is structured on the highest modeling level into the three large blocks: chassis (CHS), power train (PTR) and auxiliaries (AUX) (Figure 3).

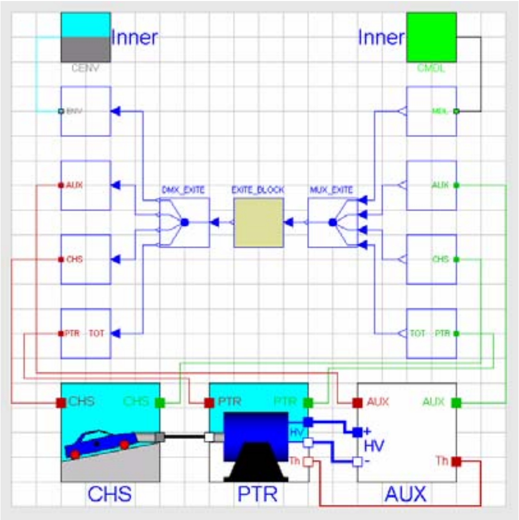


Figure 3: Structure of the vehicle model

The block EXITE_BLOCK represents the simulator coupling described in the chapter 2.4.

The model of the chassis (CHS) incorporates a vehicle model (BOD i.e. body) without lateral dynamics considering all relevant driving resistances as well as a model of the contact of the tire with the street (Figure 4).

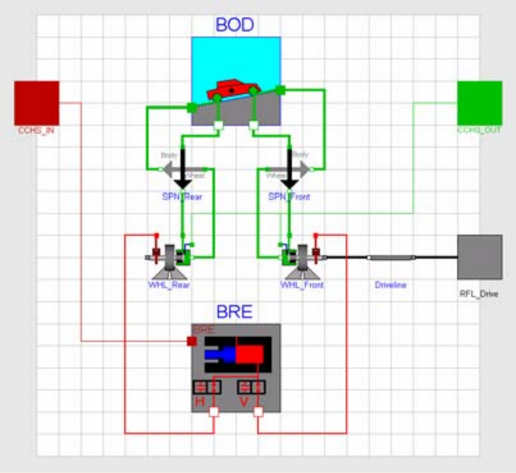


Figure 4: Model of the chassis (CHS)

Furthermore it incorporates a simple hydraulic model for the excitation of the brakes (BRE, which stands for brake model).

The model of the auxiliaries (AUX) consists of the modeled electrical consumers of the vehicle electrics (14V).

The model power train (PTR) represents the relevant components such as the internal combustion, ICE, engine (VKM, German: Verbrennungskraftmaschine), the fuel reservoir (TNK, German: Tank), the clutch between ICE and electrical machine, called separating-clutch (TRK, German: Trennkupplung; clutch 2 in Figure 1), the gearbox (GTR, German: Getriebe), the high voltage battery (BAT) and the electrical machine (Figure 5).

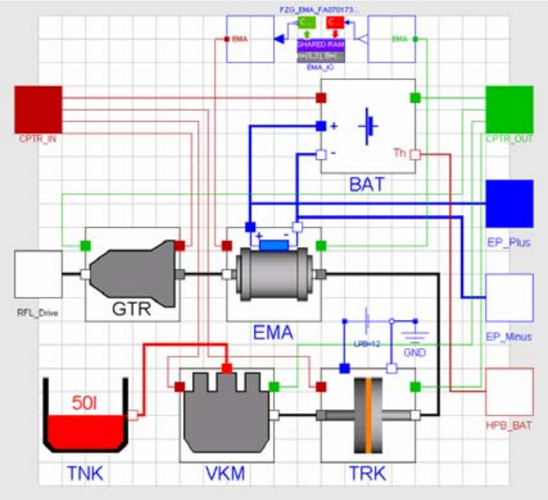


Figure 5: Model of the power train (PTR)

In the upper part of Figure 5 the shared-memory data-link is displayed which is described in chapter 3.1.

Only the internal combustion engine was modeled based on maps (efficiency maps). All other components are equation-based models and are described shortly.

The model of the fuel tank is a simple model of the flow of the fuel.

The model of the separating-clutch is a complex mechatronic model of the clutch and the flywheel. Even the hydraulic actuator and the mechanics of the separating-clutch were modeled.

The induction motor was modeled as an electromechanical drive in α–β stator coordinates based on the well known equations [5].

The NiMH battery was modeled with electrical and thermal characteristics including ventilation.

The model of the automatic transmission is a complex mechanical and hydraulic representation of the DSG® (Figure 6).

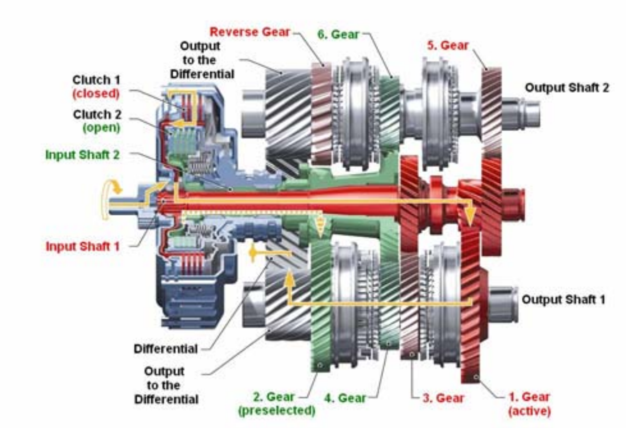
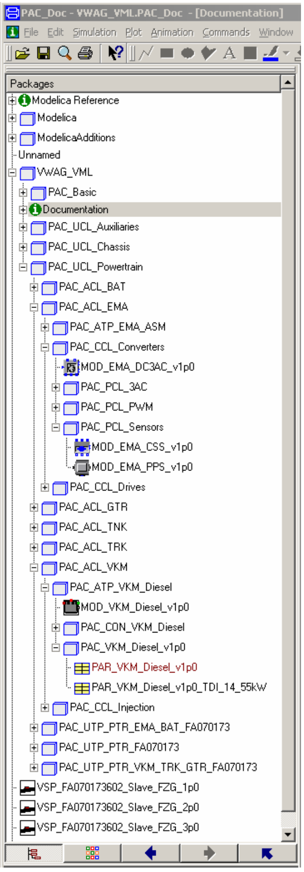


Figure 6: Section of the DSG

2.2 The Vehicle Modeling Library – VML

Dymola supports object-oriented modeling. Class libraries can be created in so-called packages. Since with the aid of the cluster simulation different drive train configurations can be examined, from the start the emphasis was put on a hierarchically structured component library (Figure 7).



- Modelica Reference
- Modelica
- ModelicaAdditions
- Unnamed
- VWAG_VML
  - PAC_Basic
  - Documentation
  - PAC_UCL_Auxiliaries
  - PAC_UCL_Chassis
  - PAC_UCL_Powertrain
    - PAC_ACL_BAT
    - PAC_ACL_EMA
      - PAC_ATP_EMA_ASM
      - PAC_CCL_Converters
        - MOD_EMA_DC3AC_v1p0
        - PAC_PCL_3AC
        - PAC_PCL_PWM
        - PAC_PCL_Sensors
          - MOD_EMA_CSS_v1p0
          - MOD_EMA_PPS_v1p0
      - PAC_CCL_Drives
    - PAC_ACL_GTR
    - PAC_ACL_TNK
    - PAC_ACL_TRK
    - PAC_ACL_VKM
      - PAC_ATP_VKM_Diesel
        - MOD_VKM_Diesel_v1p0
        - PAC_CON_VKM_Diesel
        - PAC_VKM_Diesel_v1p0
          - PAR_VKM_Diesel_v1p0
          - PAR_VKM_Diesel_v1p0_TDI_14_55kW
      - PAC_CCL_Injection
    - PAC_UTP_PTR_EMA_BAT_FA070173
    - PAC_UTP_PTR_FA070173
    - PAC_UTP_PTR_VKM_TRK_GTR_FA070173
- VSP_FA070173602_Slave_FZG_1p0
- VSP_FA070173602_Slave_FZG_2p0
- VSP_FA070173602_Slave_FZG_3p0

Figure 7: Structure of the Dymola library VML

This library handles the components and the component structure as well as the variants of components. In the following, attention is shortly being paid to the construction of the component-library VML.

There are basically four levels of hierarchy: the UCL - classes of the models of complete units, the ACL - classes of the aggregate models, the CCL - the classes of the components and their parts – the PCL. In order to structure the system vehicle into manageable subsystems the vehicle is structured into the abovementioned subsystems chassis, powertrain and auxiliaries which were named units. The main parts of a unit are referred to as aggregates, such as the ICE or the gearbox of the unit powertrain. Parts of aggregates are referred to as components such as the converter or the electrical drive itself of the aggregate electrical machine. So-called parts are elements of components such as different sensors or the voltage conversion of the component converter.

This structure results from the principle of decomposition assuming the following: units consist of aggregates, aggregates consist of components and components consist of parts. The structure is displayed by the structure of packages in Dymola. The packages by themselves are subdirectories on the hard disk.

Classes of units (UCL) and aggregates (ACL) furthermore can contain packages of connectors (Pac_CON_...) and models (Pac_UTP_... and Pac_ATP_... resp.; TP stands for a special type or model). Archived models of particular aggregates or units are stored in the type package. Parameterizing is supported by using records.

2.3 Modeling of the Controller

The difficulty of the simulation of the drive train is not mainly modeling the physical system; it can be modeled with sufficient accuracy with more or less effort.

The problem rather is the modeling of the control algorithm of the components that represents the control characteristic. So the functionality of the ICE control, the transmission control, the electrical machine control and the ABS/ASC control was modeled with relatively large effort. All models in Matlabt/Simulink were modeled discrete (for instance fixed step size 10 ms).

The algorithm of the battery management system (BMS, that is the ECU that controls the battery with respect to its boundaries) could be directly inserted into the cluster simulation as a DLL since the algorithm was developed by the author himself. And the complete algorithm of the energy management ECU (Vehicle Management Unit, VMU), which was developed in Matlab/Simulink, could be inserted too. For these two last-mentioned ECU algorithms there is a software development process with which one can generate the flash code directly out of the simulation by means of automatic code generation or compilation with the target compiler for the ECU (see Figure 2). In case of the BMS this process is a C programming language software development process and for the VMU a Matlab/Simulink/Real-Time-Workshop (RTW) software development process. Both processes were used in this way within the above-mentioned SUVA project.

2.4 Master Model

In the so-called master model which was modeled in Matlab/Simulink the interface data of all submodels are exchanged via a network of the representatives of all submodels which is fed back on itself (Figure 8).

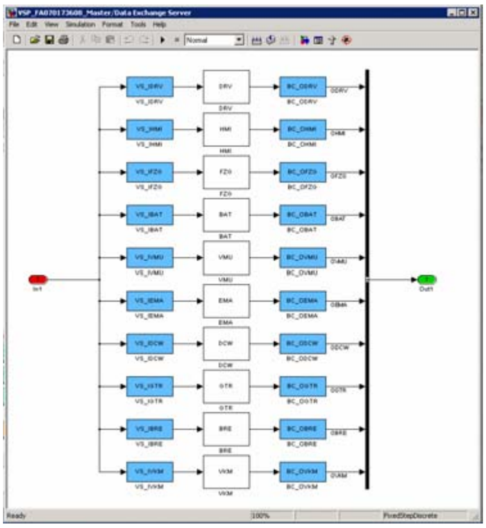


Figure 8: Feedback structure of the representatives of the submodels (the output is fed back without delay onto the input)

As already mentioned in chapter 2.1, the simulator coupling is carried out by the tool EXITE of the company Extessy AG, Germany [6]. EXITE realizes a simulator coupling on the basis of a client-server-linkage. The server is the representative of a submodel which only provides the interfaces according to the regarding submodel. The client is the submodel itself.

The Extessy AG provides a simulator coupling for different simulation tools for example for Matlab/Simulink, ASCET-SD[7], Saber[8] and Dymola.

Several methods for data communication between the client and server are supported such as the simple sequential communication and the full-duplex communication. EXITE relies on the ISO-OSI layer model of communication. So the communication protocols TCP/IP and MPI are supported too. The Master Model handles all interface data of all submodels, therefore it reflects the so-called communication matrix which shows which submodel exchanges which data with which submodel in what sample time. In this cluster simulation over 900 signals are exchanged mainly because of the emulation of the CAN bus (Controller Area Network – a commonly used network in the automotive context). 500 of these signals are relevant stimulation inputs for the simulation.

The VMU and the BMS are stimulated in this way with all signals available in reality. Thus this

---

[7] ASCET-SD is a registered trademark of ETAS GmbH, Germany

[8] Saber is a registered trademark of Synopsys, Inc., USA

simulation represents a real SIL simulation concerning these two ECU algorithms. The relevant signals are considered for the other ECU.

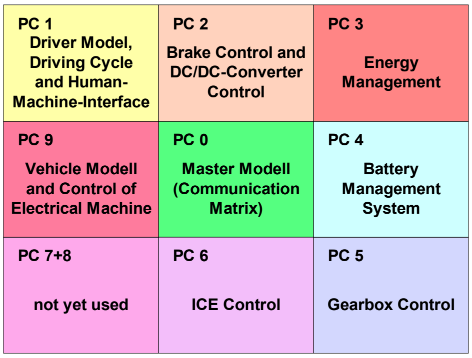
3 First Results

3.1 Shared-Memory-Coupling

The control of the electrical machine was realized with the switching frequency of the converter which is 8kHz. The model coupling of the electrical machine in Dymola and its control in Matlab/Simulink would slow down the simulation extremely. For this reason, a shared-memory data-link was created so that two processes access the same storage area. The processes are controlled so that both are processed on different processors of a dual processor computer. Thus the vehicle model is carried out on one processor as a Dymosim.exe. On the other processor the electrical machine control is processed as a Matlab/Simulink/RTW-executable. The shared memory block is represented in the upper part of the Dymola model in Figure 5.

3.2 Partitioning

The distribution of the submodels on the individual computers was done according to performance aspects, because the slowest simulation determines the overall performance of the cluster. For this reason the partitioning shown in Figure 9 was carried out.



PC 1 **Driver Model, Driving Cycle and Human-Machine-Interface**	**PC 2** **Brake Control and DC/DC-Converter Control**	**PC 3** **Energy Management**
PC 9 **Vehicle Modell and Control of Electrical Machine**	**PC 0** **Master Modell (Communication Matrix)**	**PC 4** **Battery Management System**
PC 7+8 **not yet used**	**PC 6** **ICE Control**	**PC 5** **Gearbox Control**

Figure 9: Distribution of the submodels on the individual simulation PC

It was expected that the vehicle model determines the performance of the overall system due to the detailed modeling of the components. For this reason two more computers were reserved (PC 7 and 8 in Figure 9) in order to split the vehicle model into several parts and to simulate them separately in case of simulation overload.

3.3 Benchmarks

As described in chapter 2.4, different combinations of the communication protocols and of the communication methods are possible.

All benchmarks of cluster simulation were carried out with the full-duplex communication method with the protocol MPI/GM[9] via Myrinet, the optical network.

The Matlab/Simulink models were modeled discrete. For the Dymola drive train model the integration algorithm **lsodar** (a multi-step-solver with a variable step size for continuous and discreet systems) has proved to be very robust.

In Table 1 the results of several benchmarks are listed. The third column contains the ratio of simulation time to simulated time (RT/tsim).

Table 1: Results of the performance measurements

Sim. No.	Configuration	RT/tsim
(1)	all models as dummies – i.e. empty models	67.1
(2)	all ECUs as RTW-exe, except gearbox ECU; vehicle as dummy models	8.9
(3)	all ECUs as RTW-exe, except VMU-ECU; vehicle as dummy models	5.9
(4)	vehicle model as Dymosim.exe, all ECUs as RTW-exe, EMA control with 8 kHz shared-memory data-link	1/390
(5)	as (4), EMA control with 4 kHz shared-memory data-link	1/216
(6)	vehicle model split into the electrical high voltage part, modeled in Matlab/Simulink and compiled to an exectuable and the remainder as Dymosim.exe; all ECUs as RTW-exe	1/15 [10]

To examine the influence of the Matlab/Simulink ECU models on the performance of the cluster simulation in simulation No. (2) and (3) of table 1 the gearbox controller and the VMU controller, respectively, were replaced by empty models (so-called dummies). As mentioned above, all other Matlab/Simulink models were RTW-executables. Even the Dymola plant was simulated by a dummy. The simulations were 8.9 and 5.9 times faster than real time. This means that because of the complexity of the model of the gearbox controller the performance of the cluster simulation will be more influenced by the gearbox controller than by the energy management controller. Furthermore it is obvious that even

---

[9] GM is a registered trademark of Myricom, Inc., USA

[10] Estimated value based on simulation of the vehicle model without shared memory data-link and without control of electrical machine

if the plant could be simulated faster as the gearbox controller, the cluster simulation would be only a maximum of 5.9 times faster than real time when performing SIL-simulation for VMU controller algorithm development.

For the sake of comparison, the simulated time for a complete empty-cluster simulation is given in simulation No. (1). Only dummy models were simulated. It follows that the pure communication of empty models is 67.1 times faster than real time. Per simulation step (10ms fixed step), approximately 150µs is required (operating system and Matlab/Simulink overhead).

In simulation No. (4), all dummy models were replaced by their respective models. The vehicle model was compiled by Dymola into the executable Dymosim.exe. All Matlab/Simulink models were compiled by Matlab/RTW into executable files. The simulation of the control of the electrical machine (fixed step size) and therefore the data exchange via shared-memory data-link between the electrical machine in Dymola (variable step size) and the control of the electrical machine in Matlab/Simulink was carried out with a 8kHz switching frequency of the converter of the electric drive. As a result the simulation was 390 times slower than real time. Of course this result is caused by the communication step size of 125µs. As mentioned above, the integration algorithm used in Dymola was **lsodar** with a tolerance of 1E-5. (The model did not run by a tolerance of 1E-4.) The data exchange between vehicle model and the control of the electrical machine organized by EXITE every 10ms activates an event in Dymola. As a consequence additional CPU time is required through reinitialization during solving the differential equations and thus the performance of the system slows down.

Reducing the switching frequency and in this way the frequency of the data exchange between Dymola and Matlab to 4kHz still leads to a simulation which is 260 times slower than real time. Moreover, with this lower switching frequency at maximum rotational speed of the electrical motor no effective mechanical torques can be generated.

The transition of modeling the electrical machine in $\alpha-\beta$ stator coordinates to d-q field coordinates and, hence, the loss of the universal description of the machine for the benefit of the symmetrical machine would reduce the simulated time and the effort due to data exchange in such a way that the simulation rate would be moved into manageable proximity. The data exchange via shared-memory then could be done in 10ms steps. Only one disadvantage would arise: the harmonic pattern and consequently the torque ripple of the electrical machine would be simulated no more. However, the torque ripple supplies an insignificant contribution with regard to its effects onto the torque characteristics of the drive shaft during the software development of the energy management algorithms and so it could be neglected.

Keeping this in mind, the high voltage electric part of the drive train is presently removed out of the Dymola vehicle model onto a Matlab/Simulink model which also includes the controller of the electrical machine (estimated simulation time see simulation No. (6) in Table 1). Thus the performance of the drive train simulation could be increased at least to 15 times slower than real time.

4 Alternatives for Increasing the Performance

It has been shown that the influence of the modeling of a complex controlled system on the performance of the entire simulation is quite important. To put it precisely: in the present cluster simulation the Dymola model determines the overall performance. For this reason possible alternatives for improving the performance of the cluster simulation will be described in this chapter.

4.1 Simplification of the Modeled Plant

The vehicle model must be redesigned in such a way that models with only small influence on the entire simulation or those with a vague description are reduced to simple constants or low-pass filters of first order.

The auxiliaries model for example: a constant efficiency and a constant load at the 14V power supply can be accepted. The effect on the development of the energy management algorithm is minimal and the error can be accepted.

4.2 Elimination of Stiffness

The stiffness values of the system have an essential influence on the performance of the cluster simulation. These must be identified and eliminated. Since that was not done until now, a further increase in performance can be expected.

4.3 Calculating Vehicular Submodels Separately

As it was shown in chapter 3.3 it is possible to shorten the simulated time by splitting the vehicle model into several parts. A further alternative for increasing the performance therefore is the identification of effects with large time constants and submodels which can be calculated separately. So a second or a third Dymola session could be opened and for example the simulation of the auxiliaries could be calculated separately.

4.4 Usage of more Efficient Solvers

The simulation slows down due to events (10ms data exchange via EXITE, discontinuities in modeling) and because of the variable step size and the solver used in Dymola.

The most efficient way for speeding-up the simulation is the use of a single step solver for continuous systems with variable step size and state event handling. The usage of Dynasim's GODESS library (GODESS stands for generic ODE solving system) that incorporates such solvers is presently proved.

4.5 Replacing Modeled Controller by its ECU-DLLs

If the real ECU code of a controller is available, complex modeling can be avoided. Furthermore, the ECU code is often more efficient. Thus, the BMS code could be included into the cluster simulation. In the same manner the DSG®-ECU code could be linked to the cluster simulation. By doing so and together with all other herein mentioned possibilities for increasing the performance of this simulation there could be an increase in simulation time which would result to a 9 times faster performance than real time (see simulation No. (2) in Table 1).

5 Conclusions and Future Work

A complex mechatronic simulation was presented in a heterogeneous cluster of simulators used for hybrid drive train simulation in the automotive industry. The objective was to clarify whether or not it is possible to set up a manageable SIL process with extensive computational aid. As a result it can be said that on principle ECU algorithms can be developed with the aid of the presented method. An advantage of SIL compared to traditional applications in the vehicle is that the control algorithms can be developed robust in respect of fluctuations in components and environmental data and in a reproducible manner. Effects of the communication between the ECUs can also be examined. Decisive for the manageability of such complex simulation is the level of detail of the submodels and the solver used.

With Modelica/Dymola as an object-oriented, multi-domain modelling tool it is possible to alter plant structures in a fast way.

One next step has to be the validation of the simulation. For this purpose, an approximately 300 km long driving cycle has already been measured.

In future more ECU algorithms will be linked as DLL into the cluster simulation which makes the control characteristics more realistic and reduces the amount of work necessary for modeling control algorithms.

For the development of controller algorithms and for the specification of components, an automated simulation will be designed; with it, parameters can be changed within their boundaries by predefined scripts or Monte-Carlo analysis, allowing massive parameter variations to be carried out automatically.

In order to obtain a manageable SIL, the cluster simulation has to be redesigned to be faster than real time.

6 Acknowledgements

This work was done within a project of Volkswagen R&D in cooperation with various partners. For this reason the author would like to thank the respective colleagues of the Audi Electronics Venture GmbH, of the Extessy AG as well as of the Lineas Automotive GmbH and there in particular Wolfgang Borgs and Vadim Bulakhov for their assistance in modeling and puting the cluster simulation into operation.

References

[1] J.-W. Biermann, C. Bunz, M. Crampen, S. Köhle, D. Mesiti: Drei OEM, ein gemeinsames Antriebskonzept – Drei neue Hybridfahrzeuge, entwickelt im EU-Projekt SUVA, 13. Aachener Kolloquium Fahrzeug- und Motorentechnik, Aachen, 2004

[2] Timo Götte, Thomas Pape: DSG – Das Direktschaltgetriebe von Volkswagen, VDI-Tagung "Getriebe in Fahrzeugen 2004", Friedrichtshafen, 22.-23. Juni 2004

[3] Roland Kube, Michael Böckl, Mathias Hommel, Siegfried Köhle: Energy Management Strategies for Hybrid Drive Train Sys-

tems Using Infrastructure Information, EUR-motor New Advances in Electronics Engineering, Energy Management – Today and tomorrow, 23.-24.9.2004, Aachen

[4] M. Tiller, D. Linzen: A Comparison of Different Methods for Battery and Supercapacitor Modelling, SAE 2003-01-2290, 2003

[5] Werner Leonhard: Control if Electric Drives, Springer Verlag, 2. edition, 2000

[6] EXITE Handbuch Version 1.3.0, EXTESSY AG, Wolfsburg, 2004