

On emission certification of heavy-duty hybrid electric vehicles using hardware-in-the-loop simulation

Jonas Fredriksson and Esteban Gelso
Chalmers University of Technology

Mattias Åsbogård and Michael Hygrel
Volvo Powertrain

Ove Sponton and Nils-Gunnar Vågstedt
Scania

ABSTRACT. The objective of this report is to present an investigation on a test method for heavy-duty hybrid electric vehicles using hardware-in-the-loop simulations for type-approval-tests. Traditionally, certification of heavy-duty vehicle emissions are done in an engine test bench using an engine brake and one or several predefined, standardized engine cycles. The introduction of hybrid technology leads to new demands for type-approval. The number of possible vehicle combinations becomes immense, since for a heavy-duty customer it is possible to select engine size, transmission, gear ratios, battery size, electric motor size, number of drive axles etc., this means that almost no truck is the same. In Japan a method for certification of heavy-duty hybrid vehicles using hardware-in-the-loop has been developed. The method uses measured data from some components (battery, electric motors and internal combustion engine), these data are then used in a simulation model, running against the real control system. The method shows good agreement with runs on chassis dynamometer.

The report studies the method for certification issued in Japan. The aim is to generate knowledge in the area for understanding the requirements that need to be placed on a hardware-in-the-loop simulation type-approval method. In addition to studies of the method, simulation models suitable for hardware-in-the-loop simulation are developed and presented. The report also shows a case study on what problems that occur when connecting a "real" hybrid vehicle controller to the hardware-in-the-loop setup.

CONTENTS

1. Introduction	5
1.1. Aim of project	5
1.2. Delimitations	6
1.3. Outline of report	6
2. The Japanese Method	7
2.1. Certification	7
2.2. Test procedure	8
3. Literature survey	11
3.1. Method	11
3.2. Energy efficiency	12
3.3. Modeling	12
3.4. Summary	15
4. Comments on the method	16
4.1. Abstraction level	16
4.2. Auxiliary systems	16
4.3. Centralized controller	16
4.4. Driver model	17
4.5. Driving cycles	17
4.6. Faults and failures	17
4.7. HILS or SILS	18
4.8. Level of verification	18
4.9. Manual tuning	18
4.10. Terminology and definitions	19
4.11. Unmodeled dynamics	19
4.12. Summary	20
5. Terminology and definitions	21
5.1. Abbreviations	21
5.2. Control system	21
5.3. Control architecture	22
5.4. Safety system - Faults and failures	23
6. Modeling	25
6.1. Modeling philosophy	25
6.2. Model interfaces	25
6.3. Implementation	26
6.4. Component models	27
6.5. Control system interface	34
7. Simulation	35
7.1. Case study: Hybrid distribution truck	35
7.2. Concluding remarks	39
8. System level verification	44
8.1. Example	44
8.2. Discussion	46
9. Future Work	47
10. Conclusions	48
References	49
Appendix A. Simulink models	51
A.1. Chassis	52

A.2. Clutch	54
A.3. Driver	55
A.4. Electric machine	56
A.5. Energy storage	58
A.6. Gearbox	59
A.7. Internal Combustion Engine	61
A.8. Mechanical connection	63
Appendix B. Control system interface	65
Appendix C. Powertrain Data	66
C.1. Conventional bus	66
C.2. Hybrid distribution truck	68

1. Introduction

When you buy a truck today, you are introduced to a greater number of choices compared when buying a passenger car. As a buyer, you can specify the vehicle's powertrain according to your needs. Like the car buyer can choose from a number of different engines and manual or automatic transmission, but otherwise the similarities are few. As a buyer of a heavy-duty vehicle, you can not only choose engine and transmission, but also select different gear ratios in the transmission and final drive, the number of driving axles, suspension systems, etc. In the near future, the number of choices will increase further, due to the introduction of hybrid vehicle technology. The choice extends not only of some new components, such as electric motors/generators, batteries, etc, but also how the powertrain is configured (series or parallel). This may mean that virtually no powertrain are the same.

Unlike passenger cars, traditional certification of heavy-duty vehicle emissions is done in an engine test bench using an engine brake and one or several predefined, standardized engine cycles. For hybrid vehicles the traditional certification method is not that suitable, since the extra degrees of freedom that the electric machine(s) give cannot be used in a representative way. An alternative to the current engine certification could be a chassis dynamometer based test to defend the hybrid powertrain's advantages. This however, is assessed as very expensive and time consuming. Alternative methods for certifications are preferred.

To help engineers working with development of complex systems, modeling and simulation are important tools. Modeling and simulation can be used for both synthesis and verification of functionality. It can also be used for complete system performance analysis and support decisions concerning strategic choices during the development process, such as the choice of powertrain configuration for a Hybrid Electric Vehicle (HEV), sizing of powertrain components, generation of test procedures for component testing or even verification. The use of modeling and simulation for improving the certification of heavy-duty hybrid powertrains is an interesting approach.

In Japan a method for certification of heavy-duty HEVs using Hardware-In-the-Loop Simulation (HILS) has been developed, see [15], from now on referred to as the Japanese method. The method uses a simulation model of the powertrain, this simulation model is simulated in combination with the vehicle's real control system. The result from the simulation is the utilization of the combustion engine in terms of engine torque and speed profiles as function of time. The real combustion engine is then tested in an engine test cell using this torque and speed profile, in this way not all vehicles need to be tested on a chassis dynamometer.

1.1. Aim of project.

In the project *Hardware-in-the-loop simulation for type-approval of hybrid electric vehicles*, the aim is to gain knowledge in the area for understanding the requirements that need to be placed on a HILS type-approval

method. In addition to studies of the method, the project should generate simulation models suitable for HILS applications.

1.2. Delimitations.

Within the project no hardware will be developed, with this delimitation the project becomes a Software-in-the-loop (SIL) simulation project. The project will not cover:

- Development of a new driving cycle for certification
- Hardware issues such as distributed functionality among the vehicles controller units
- Free vehicle topology, (only a parallel hybrid vehicle will be studied)

1.3. Outline of report.

The outline of the report is as follows, in the first section the background of the project and the report is presented. The second section presents the Japanese method is outlined and described. A short literature survey on HILS certification is presented in the third section. In the fourth section the Japanese method is analyzed in more detail, discussing advantages and disadvantages and also outlining problem areas and proposing improvements. The fifth section introduces a set of definitions and terminology associated to HILS testing. Developed simulation models are presented in the sixth section and some simulations are presented in the following section. A discussion on the need of verification is presented in Section 8. Future work and conclusions are presented in the ninth and tenth section respectively.

2. The Japanese Method

In this section, certification of heavy-duty vehicles will be described in general and the Japanese method will be outlined and presented.

2.1. Certification.

Certification for emissions from conventional heavy-duty vehicles is normally done in an engine test bench, where the engine is running under predefined load/speed conditions. In 2001 a new certification procedure was designed and developed. The idea was to use a vehicle test cycle instead of an engine test cycle. The main reason was that “real” vehicle operation was supposed to be used for certification. Furthermore, a vehicle test cycle is more stable over time compared to an engine test cycle, a vehicle test cycle only changes with changes in traffic conditions, while an engine test cycle changes with engine and powertrain technology (introduction of hybrid technology represents one such technology change). For the procedure or method the vehicle test cycle (WTVC) was developed, see [29]. The WTVC cycle is derived from real vehicle use in Europe, Japan and in the US. It covers several different powertrains from 3.5 tonnes up to 40 tonnes. In order to use the WTVC cycle for certification it is necessary to transform the cycle into an engine test cycle (WHTC). The WHTC cycle is defined in terms of normalized engine speed and load, and is created using a generic powertrain model, see Figure 1. The normalized engine speed and load points are then “scaled and stretched” according to the characteristics of the engine that is to be certified.

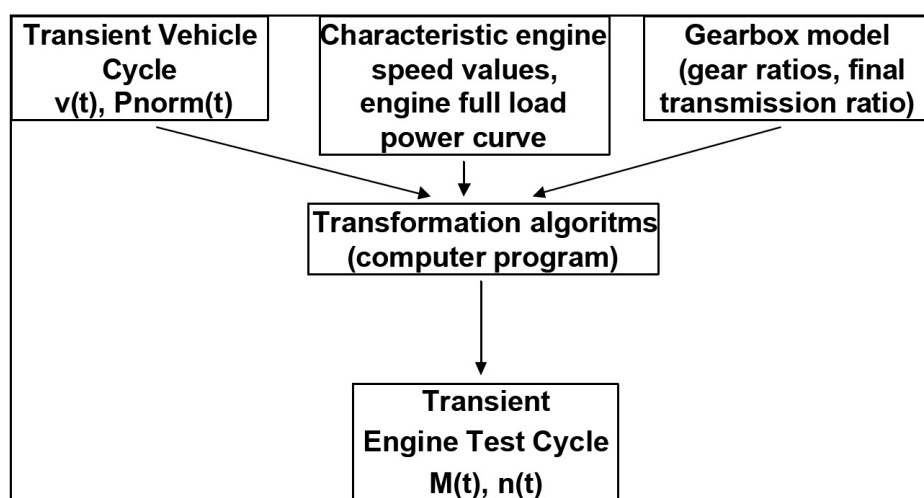


FIGURE 1. Transformation procedure for the WTVC to WHTC cycle, from [29].

The same procedure could be used for hybrid vehicles, but there are some challenges/problems. The main problem is the extra degree of freedom that hybrid's offer, the usage of the engine and the powertrain is to a much larger extent than conventional vehicles dependent on the control system (the energy management strategy). This means that the energy management

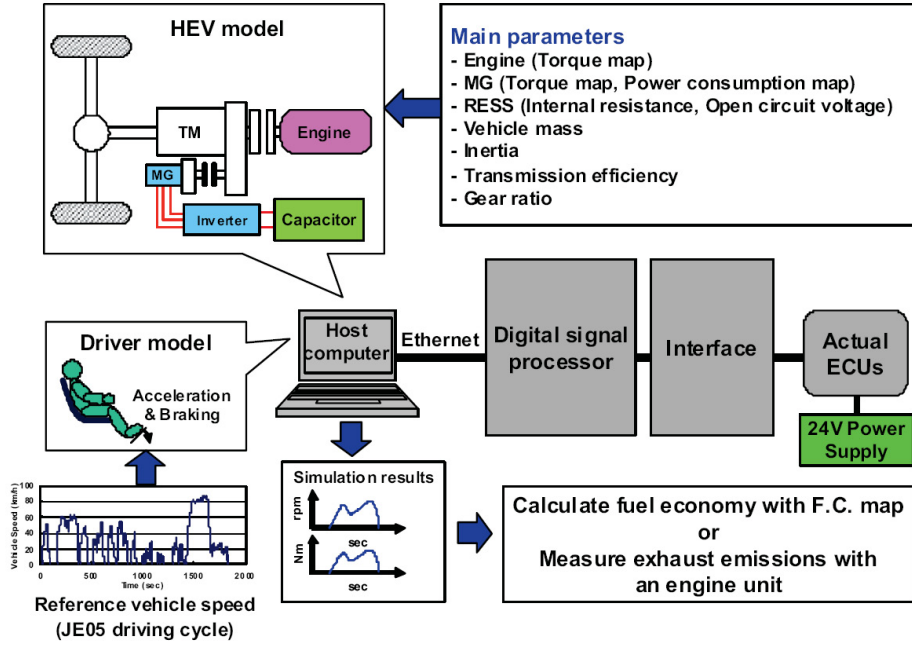


FIGURE 2. Hardware-in-the-loop simulation method, from [15].

strategy needs to be included in some way. In [15] a test method for heavy-duty hybrid electric vehicles (HEVs) using a hardware-in-the-loop simulator (HILS) is presented. This method uses the real hybrid controller unit in combination with a generic powertrain model, see Figure 2. The method proposed in [15] is to a large extent similar to the method shown in Figure 1.

The basic idea shown in Figure 2 is to simulate the powertrain in combination with real controller in a hardware-in-the-loop setup. The setup consists of a powertrain model, representing the powertrain that is to be certified, and the control system (hardware). The inputs to the powertrain model are the specific powertrain parameters. Since the energy management strategies usually are dependent on driving conditions, a driver model (representing the real driver) is needed. The purpose of the driver is to command the vehicle according to the vehicle test cycle. The setup, the HIL system, is simulated and the result, the engine speed and loading conditions are then used in an engine test bench for certification.

2.2. Test procedure.

The complete test procedure for finding the engine speed and loading conditions can be visualized using the following flow chart:

Step 1: Start of procedure and approval of test object.

Step 2: Generate data for engine, electric machines and energy storage according to the test procedures stated in [9]. Data for vehicle mass, inertias, transmissions and gear ratios should also be provided or determined.

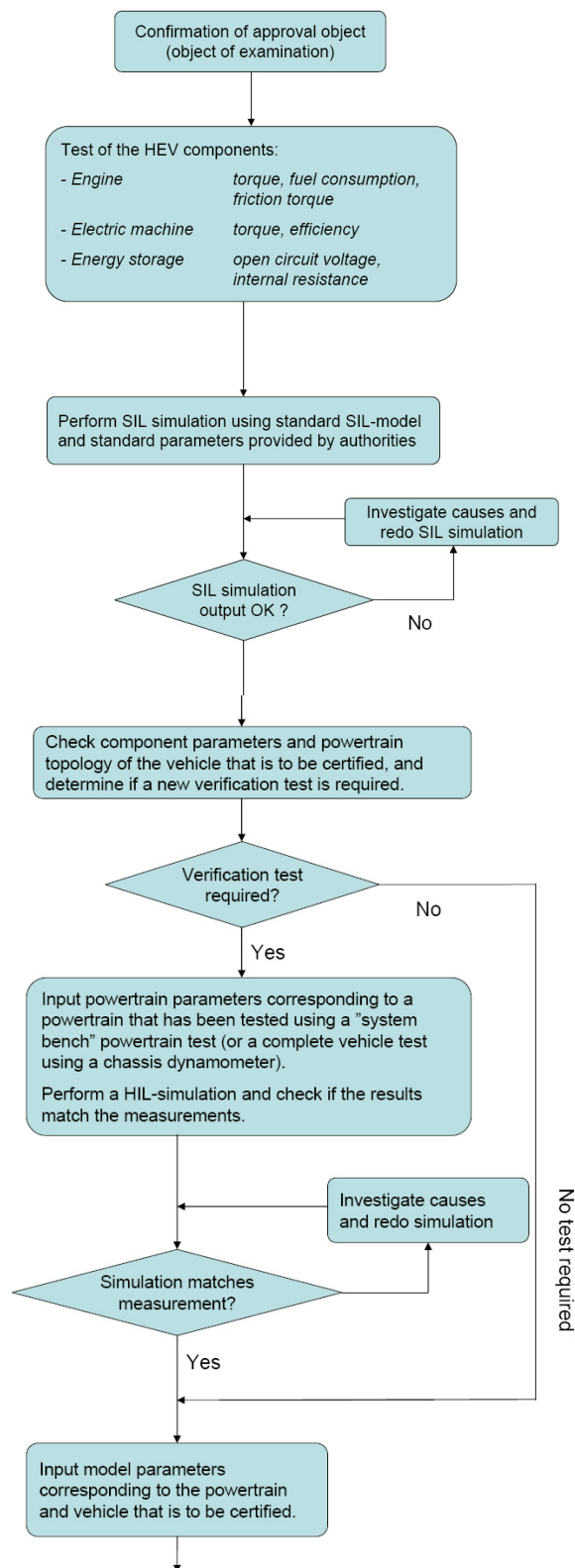
Step 3: Perform a SIL simulation using a basic predetermined control algorithm in order to ensure that the system and component models are working.

Step 4: Check that the simulation work and cause no problems (stops). If ok proceed otherwise investigate what caused the problem.

Step 5: Check if a similar powertrain topology has been certified before, if so check if the powertrain parameters are close to the previously certified powertrain. Analyze if additional verification is needed, if yes do a system verification otherwise go to step 7.

Step 6: If no similar powertrain has been certified before, it is necessary to verify that the model that is to be used represents the vehicle that is to be certified. This is done either using a complete "system test bench" or on a chassis dynamometer. Analyze if the model represents the vehicle, if yes continue to step 7, else investigate causes.

Step 7: Input the model parameters into the HIL simulation model.



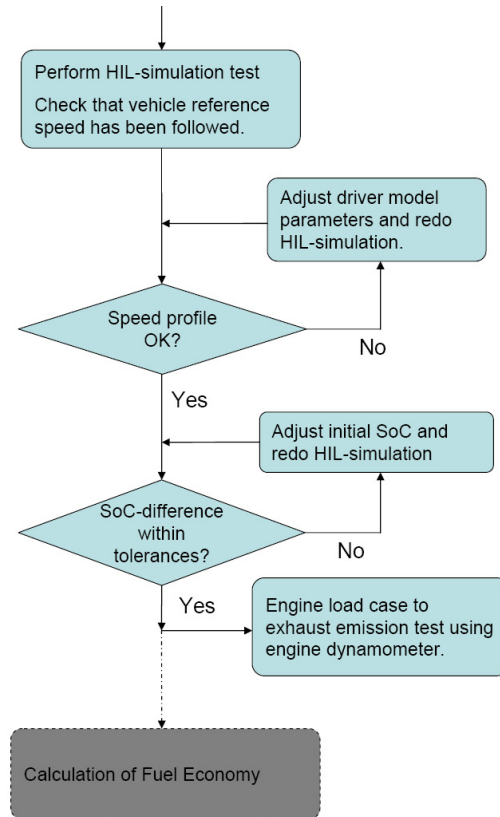
Step 8: Do a HIL simulation.

Step 9: Check if the vehicle follows the reference speed (the predefined driving profile). If yes go to step 10, otherwise adjust driver parameters and redo the HIL simulation.

Step 10: Check if energy level in the energy buffer is within the tolerances for deviation. If yes go to step 11, otherwise adjust initial value of energy level and redo the HIL simulation.

Step 11: Use the engine speed and load profile from the simulation in an engine test bench for certification.

(*Step 12:* Calculate fuel consumption from fuel consumption map.)



The steps can be summarized as component modeling and validation against measurement data, powertrain system development, including system modeling, benchmark energy management control system design and validation against old measurement data. The idea to validate the system performance against old measurement data is the key in the method. If the performance is close enough to a previously validated system, the powertrain system is assumed to be valid and type-approval of the vehicle can be performed. If the powertrain performance differs from a previously validated system, the complete system needs to be validated against chassis dynamometer tests. A number of test for validation on system level is proposed. The outcome of the paper is a verification procedure for HEVs using HILS.

A brief presentation of the method is available in [15], while a more thorough presentation can be found in [9].

3. Literature survey

In this section a brief summary of interesting papers that are related to HILS for heavy-duty HEVs are given. The aim is to outline the papers and its relevance to the area. The papers, reports and other documents reviewed are categorized into different categories. Several paper fit into several different categories.

3.1. Method.

In [15] a test method for heavy-duty hybrid electric vehicles (HEVs) using a hardware-in-the-loop simulator (HILS) is explained. This method is used for exhaust emission (and fuel economy) type-approval-tests, as an alternative to the “system bench method”. It reproduces the operating conditions of the whole hybrid system by simulation and the result, the engine speed and load points can be run on engine dynamometer. The paper also presents component models and standardized testing of HEV models for parameterization.

Skjetne and Egeland, [28], propose a methodology for verification and certification of marine control systems that distinguishes between functional and performance HIL test. On the one side, *functional testing* verifies that the functions of a target system fulfill the requirements, such as failure testing. *Failure testing* is concerned with robustness of safe operation and availability of the control system in the event of failures. On the other side, *performance testing* quantifies the level of performance when executing them in different functional modes when exposed to different exogenous conditions.

In, [27], the same authors as [28] proposed a framework for constructing HILS test applications. The framework includes formalizing test necessity and development of a terminology and a set of definitions related to HILS certification. The terminology and the definitions are based on IEEE and IEC standards, and it should not be any major problems to adapt these to the automotive application. Their work has led to a standard for HIL certification of marine control systems, see [1].

Filipi et. al., [5], presents a setup for running a real engine and simulating a vehicle, so called engine-in-the-loop. The main idea is to evaluate how a real engine perform when integrated in a hybrid powertrain. The setup is quite similar to the Japanese method presented in [15]. The main difference is that in the work by Filipi et. al. the real engine is running during simulation, while in the Japanese method, the engine is run afterwards based on the torque and speed profile generated during simulation. The advantage with the method presented in this paper is that the real engine dynamics is incorporated into the simulation. The powertrain simulation is controlled by a driver model using preview.

In the paper by Nakai et. al., [16], the design of a HIL system is presented. The intended use of the system is to evaluate the performance of ECUs. The system has models for engine, battery, brakes, chassis, electric motor and inverter. The system can handle distributed ECUs. It addresses and highlights important topics in the design of a HILS system, but no

deeper analysis or descriptions are presented. It could be useful inputs if distributed control systems should be addressed.

3.2. Energy efficiency.

Several papers discuss the problem of how to measure the energy efficiency of hybrid powertrains, since it is difficult to evaluate the fuel economy of an HEV due to the transfer of energy to and from the on-board energy storage. The same test performed several times gives for a conventional vehicle the same result (the same fuel consumption), while for a hybrid powertrain the result may differ from test to test, due to transient energy storage and the initial energy level in the energy storage, State-Of-Charge (SOC).

The paper, [2], outlines a linear regression model for compensating of change in SOC. The paper also outline three ways to characterize hybrid vehicle performance, electric-only capability, charge sustaining and charge depleting. In [26] the way of using a linear regression model for compensating of change in SOC is discussed and the problems associated with this method is highlighted. An extended version of the linear regression model is proposed and the pros with this extended version is discussed in detail.

Another interesting problem associated to energy efficiency is highlighted in [3], it discusses the measurement challenge with HEVs. The authors address the problems of measuring SOC and emissions. It has a special focus on the problem with SOC measurements (or the lack of measurements) and discusses how the estimation of SOC effects the results and the control strategy. The paper relates more to testing of HEVs than simulation, but outlines well that results might be very sensitive to errors in SOC estimation. It is important to relate to these problems when designing a type-approval system using HILS.

3.3. Modeling.

For HIL applications, modeling is a very important part. In this subsection modeling on system level as well as on component level is discussed. Papers referred to are either related to mathematical models which can easily be implemented in a HILS system or related to modeling issues, such as validation of models.

3.3.1. Powertrain modeling and validation.

There exist a number of complete powertrain models that are suitable for use in HIL systems, see e.g. [23] or [25]. The models used are simple dynamic or quasi-static models. The models can rather easy be compared against measurement data of a real vehicle, i.e. validated. The powertrain components included are engine, transmissions (different kinds), differential and final drive. The models presented are not that different from the models used in [15], so the main outcome from these papers are the powertrain parameters, which might be of interest for some powertrain models, in a HIL simulation setup.

A powertrain model used by the Japanese government to investigate fuel consumption for conventional powertrains is presented in [18]. The model is implemented in FORTRAN-code and represents a complete conventional powertrain. Unfortunately, it is not clear how the model is to be used. The idea could be to find the lower bound (optimal) fuel consumption for a conventional powertrain using a standardized powertrain model. The model is referenced several times in the paper by [15], but only for verification of simulation parameter values. Some powertrain parameters, such as gearbox efficiency, can be found in this document.

In [21], the importance of validating both the component models and the system model is highlighted. If the simulation models are not adequately validated, the results lack credibility and the authors claim that certification can not be performed. The authors present ideas on how to validate forward simulating models. The simulation models are implemented in PSAT, a simulation tool developed by Argonne National Laboratory. The conclusions made in this paper are the same as in [15]. Since the paper by [21] is much earlier than [15], it is probable that Morita et. al. used this as an input when developing their method.

3.3.2. *Energy storages.*

For energy storages or buffers, the paper by [17] can be considered as a state-of-the-art paper for better understanding how batteries work. This paper reviews the specifications and operational requirements imposed on batteries for HEVs. The operation of batteries in HEVs involves unusual constraints not seen in other applications, such as high currents, fast transients from charging to discharging.

Another source of information is the website by ThermoAnalytics. Interesting papers related to battery modeling are presented. It also presents different battery models, such as electrochemical battery models, equivalent circuit battery models and lumped parameter battery models. Furthermore, it discusses how to model batteries in general, and outlines what is specific with different battery types.

In the master's thesis by Kärverud, [10], degradation of Li-ion batteries are treated. The main factors causing degradation are found to be high temperatures and number of capacity turnarounds. These factors cause among other things rise in the internal resistance and loss in maximum capacity of the battery. A mathematical model is developed and compared with measurement data.

3.3.3. *Electric machines.*

The models of the electric machines used in the Japanese method are simple static maps with a first order filter to represent the dynamics. The models are more or less standard models for representing electric machines.

In [30] a brief introduction to how the dynamics of permanent-magnet synchronous machines (PMSMs) can be modeled. The document includes

enough information for building a dynamic PMSM model, if the simple model used in the Japanese method turns out to be too simple.

The paper by Okuda et. al., [19], presents how the system presented in [16] can be used. Its main focus is on how to model the electric machines. The electric motor model presented and used for HILS is a 3-phase motor model of a PMSM. It is a dynamic model and it is more complex compared to the simple electric equivalent model used in [15]. The model is verified against measurement data with very good results. The model is similar to Oskar Wallmark's model presented earlier in this section.

In the report by Laine and Fredriksson, [13], modeling and control of different hybrid powertrains (the focus is on passenger cars) is presented. In the modeling section, model for different powertrain components are presented. The models are mostly based on standard models found in the literature, but for the electric machine a thermodynamic model has been developed. The thermodynamic model is more or less a stand alone model, using the power as input signal.

3.3.4. *Transmissions, clutches and other machine elements.*

Looking in the literature, the models of transmissions used for HIL are simple static gear ratios (modeled as gains), which are changed during gearshifts. Other machine elements, such as final gears, differentials etc., are also simple to model. The most difficult part to model is the clutch. The master's thesis by Kusters, [12], dealing with frictional components in automotive powertrains, gives a good introduction to the problems. It is a comprehensive document about modeling of stick-slip phenomena in frictional components, such as clutches. In the thesis different methods for modeling and simulating discontinuous systems, clutches, are presented and analyzed. It also includes numerical solvers for simulating the models presented.

3.3.5. *Chassis, brakes and tires.*

Chassis, brakes and tires are not usually categorized as part of the powertrain, but in order to develop a complete vehicle, these components are necessary to be modeled. For powertrain simulations the chassis is usually modeled a simple mass (or large inertia). This works fine, but there are things that one should be aware of when hybridizing a vehicle. For example, in the PhD thesis by Lingman, [14], it is concluded that the foundation brakes must be used frequently in order not to degrade, i.e. extensive use of regenerative braking might not be appropriate.

Furthermore, as stated in [11], since regenerative braking cannot be performed on all wheels, it might be of interest to study, not only longitudinal dynamics, but also lateral dynamics (vehicle stability). This requires a chassis model including longitudinal, lateral, yaw, roll motion. The SAE paper by Salaani, [24], presents a vehicle dynamics model for a driving simulator. The model is to be used in real-time applications. The vehicle model is validated against measurement data. The vehicle dynamics model and the

validation might be of interest from a HEV perspective. Another outcome from this paper is the tire model, longitudinal and lateral slip are solved using first-order dynamics. Using this model, the problem of zero speed can be avoided, meaning that the model seems to be valid in the vehicles complete operating region.

3.4. Summary.

Hardware-in-the-loop simulation has been used in the automotive industry for quite a long time, since it is an efficient way to test part or complete systems. For certification purposes of complete automotive systems it can be concluded that HILS has not been used, except for the Japanese method. The method proposed in Morita et. al., [15], is the first method published using HILS for certification of vehicles. In the marine industry HILS technology has been used for certification of certain control systems. There are things to learn from these activities, such defining terminology and definitions.

From a model point of view it can be concluded that models similar to the models used in [15] are representative for using in HILS applications. Looking in the literature simple modifications to the models used in [15] can be found. These modifications do not mean a higher computational burden or an extensive calibration process.

Hybridizing the powertrain makes it difficult to evaluate the energy efficiency of the vehicle, since the results very much depends on the control system. Furthermore, the results might differ from evaluation to evaluation depending initial conditions on the vehicle.

4. Comments on the method

This section discusses the Japanese method for HIL simulation for certification of heavy-duty hybrid vehicles, in relation to what has been found in the literature and questions that arises when studying the method by itself. The following points will be treated in this section (in alphabetic order):

- Abstraction level
- Auxiliary systems
- Centralized controller
- Driver model
- Driving cycles
- Faults and failures
- HILS or SILS
- Level of verification
- Manual tuning
- Terminology and definitions
- Unmodeled dynamics

4.1. Abstraction level.

For hybrid powertrains the variety of models available is huge, ranging from simple maps via models of how the cogs in a transmission interact with each other to the chemical reactions in a battery on molecule level. Therefore, it is important to select a suitable level for modeling all involved components. The components should have the same level of abstraction. In the Japanese method, the model level is similar for all components. Furthermore, looking in the literature, the selected level seems to be correct. The models are simple and easy to parameterize.

4.2. Auxiliary systems.

One feature with hybridizing a vehicle, is the fact that some auxiliary systems can be electrified and used/controlled more efficiently. This feature is often neglected or ignored in many fuel consumption or emission studies. The Japanese method does not cover the use of electrified auxiliary systems or auxiliary systems in general. Incorporating models of auxiliary systems is difficult for several reasons, it is very time consuming, not only is it time consuming to build the actual models, it is also time consuming to validate all the models and it is also necessary to select which systems to include or model. The most time efficient way to include auxiliary system dynamics is to characterize all systems lumped together in the frequency domain. But on the other hand, including auxiliary systems might not be necessary, since most certification test are performed with minimum use of auxiliary loads. A comment on this issue is also made under the section called Future work.

4.3. Centralized controller.

One difference between Japanese and European automotive manufacturer seem to be the partitioning of the controller units. The Japanese

manufacturers seem to use a centralized controller with all control functionality in one controller, while the European manufacturers are using several different controller units. The controller units are not dedicated to one task, instead the functionality is divided onto several units. This means that connecting the controller unit to the HIL system is not that easy for the European manufacture compared to the Japanese. The Japanese method is built upon the use of a centralized controller.

To adapt the Japanese method for European needs, it is necessary to include the possibility to use decentralized controller units. As a starting point, the work by Nakai et. al., [16] can give useful inputs. Another solution could be to use pure SIL, this will be discussed later on in this chapter.

In the section named Simulation, a test is performed on how to use the Japanese method with a decentralized control system developed by Volvo Powertrain.

4.4. Driver model.

The driver model used in the Japanese method is, as commonly used, represented by a PI-controller. The main problems with using a PI-controller is that it requires some manual tuning of the controller parameters and that the same parameters are used for all gears (gear change the dynamics). This can be seen in the Japanese method in *Step 9*, where the controller parameters need to be tuned until the vehicle speed follows the reference speed within some tolerances. One way to handle the problems with using a PI-controller could be to use an adaptive controller, which adapts to change in dynamics, (a Model Reference Adaptive System could be suitable). Another alternative to investigate is to incorporate a driver model using preview as presented in [5].

It should be noted that from a governmental point of view simplicity is preferred, and a driver with only two parameters to tune (PI-controller) is rather simple.

4.5. Driving cycles.

One important part of a certification method is the choice of input or in this case driving cycle. In the current project this is left out, but it is an important thing to think about. The development of the WTVC test cycle is based on analysis of vehicle use, so this is probably still valid for hybrid vehicles. One thing to consider is the fact that future vehicles will be more tailored for specific tasks, city driving, emission free zones etc. These facts should be incorporated into the certification procedure, maybe resulting in a driving cycle with different modes (zero-emission etc.) or maybe more tailored.

4.6. Faults and failures.

One outcome from the work of [28], is to extend the method of [15] with failure testing. It can be included to analyze the robustness of the control system under fault conditions of work. Fault conditions could include

incipient or degradation faults in e.g. batteries, and due to aging and wear. Then HIL testing should test fuel consumption and exhaust emissions after a predefined time of vehicle use. This is similar to including engine warm-up in the test procedure.

Another interesting problem related to faults is as mentioned earlier in the report associated to energy efficiency, see [3]. It discusses the measurement challenge with HEVs. The authors address the problem of measuring SOC and discusses how the estimation of SOC affects the results and the control strategy. The paper outlines well that results might be very sensitive to errors in SOC estimation. It is important to relate to these problems when designing a type-approval system using HILS.

4.7. HILS or SILS.

As mentioned earlier in this chapter, the Japanese method is designed for a centralized controller unit. To connect several controller units (a decentralized control system) to the HIL system might be difficult. Ideas on how to handle this is presented in [16]. One other way to handle this, might be to use pure SILS. Meaning that the controller is implemented in a simulation software as well and then simulated together with the powertrain model. The advantages with this is that distributed control systems can be handled and the simulation time can be reduced (not necessary to simulate the system in real time) and the need to bring additional equipment, ECUs and cables is reduced, the disadvantage is that the controller needs to be translated into a simulation model. The controller can on the hand be a compiled version of the true control system. This leads to another consequence, who is responsible to verify that the controller is implemented correctly. On the other hand, the procedure for checking in-use emission fulfillment that comes with EuVI might ensure this.

4.8. Level of verification.

The Japanese method is well proposed and very interesting. The limitations lie in the need of validation on the system level (chassis dynamometer required). Intuitively there is a great need for dynamometer testing, since the choices of parameters is huge for heavy-duty HEVs. If the need of system validation can be reduced in some way, the method becomes even more interesting for HEV certification.

One idea could be to investigate how large parameter variations the system can tolerate and still be considered as validated. These parameter variations can be transferred back to the component level, so on beforehand, just by analyzing the components, it can be determined if a chassis dynamometer test is needed or not.

Furthermore, the validation tests need to be investigated, are all of the proposed tests necessary or are more tests needed? In the case of more tests, which should be performed?

4.9. Manual tuning.

The Japanese method includes two steps that requires manual tuning, *Step 9* (driver model) and *Step 10* (SOC difference). The manual tuning in *Step 9* can be reduced by changing to a more advanced driver model, but the manual tuning in *Step 10* needs to be handled in some way. The SOC difference is related to the problem of how to test fuel efficiency for HEVs. In the literature a way to compensate for change in SOC is to use a linear regression model or something similar. [2] outlines a linear regression model for compensating of change in SOC. An extended version of the linear regression model method is presented in [26]. The idea is to normalize the change in energy with respect to system voltage and covered distance during test period (driving cycle). The idea to compensate for change in SOC using some model is nice, but is not the complete solution to the problem. Though, it should be kept in mind when evaluating results. In conclusion the need for manual tuning can be reduced, but it might not be of that importance to reduce it.

4.10. Terminology and definitions.

One main point from the work of [28], in order to be certified for making certification of marine control system, is the work with terminology and definitions. For automotive HIL certification testing, something similar to the terminology and the definitions should be developed. The terminology and the definitions are based on IEEE and IEC standards. It is necessary to incorporate something similar, so that all involved parts are using the same language and understand each other. Furthermore, in order to be able to perform certification of HEVs using HILS it needs to be standardized, and work regarding terminology and definition are part of this standardization. Marine Cybernetics AS, the company that the authors of [28] are associated to, is certified by Det Norske Veritas (DNV) for performing HIL certification of marine systems.

4.11. Unmodeled dynamics.

The component models used in the Japanese method are, as stated before, simple and easy to parameterize but captures the main characteristics well. Unfortunately, some of the models lack some important dynamics, such as models for temperature variations (thermodynamics) or the ability to handle topographical changes. The engine torque build-up is not accurately modeled and needs some modification. Furthermore, the gearshift dynamics is neglected and should be improved.

Of special importance are thermodynamical models for the electric machines and the combustion engine. For electric machines it is important not to overuse or overheat the machines. An extensive use of the electric machines might lead to reduced emission levels compared to the real figures. Furthermore, heavy use of the electric machines lead to battery system temperature increase or in worst case battery degradation.

Thermodynamic models for the combustion engine can also be important to include, if cold starts are to be included in the test procedure in the future. It is also common to have control strategies that use the electric machines to

support the combustion engine during for example cold starts. The problem with modeling the thermodynamic properties of the combustion engine can be avoided if engine-in-the-loop is used.

4.12. Summary.

From the discussion above and from the delimitations of the project, the focus of the work is on terminology, developing models suitable for HILS and extending the simulation models with needed dynamics and look into the problem with level of verification.

It is also interesting to notice that India proposes chassis dynamometer tests for certification, according to Michael Hygrell (Volvo Powertrain). The driving force for this might be that India has no major heavy-duty manufacturers and this allows for small manufacturers or vehicle owners to rebuild their vehicles into hybrids and get them certified.

5. Terminology and definitions

A set of definitions and terminology related to the powertrain of HEV, HIL simulators and faults/failures is described in the following section. It should be noted that the work in this section is not complete, but it should be seen as a starting point for continuation.

5.1. Abbreviations.

ECU	Electronic Control Unit
HEV	Hybrid Electric Vehicle
HIL	Hardware-In-the-Loop
HILS	Hardware-In-the-Loop Simulator
HW	Hardware
ICE	Internal Combustion Engine
SIL	Software-In-the-Loop
SILS	Software-In-the-Loop Simulator
SW	Software

5.2. Control system.

The term *control system* refers to an interconnection of components that act together in a particular configuration to perform a certain objective. Typically main subsystems of a control system include [28]:

- Power system. All systems and equipment necessary to supply electric power to all the essential consumer units in the control system.
- Actuator system. All components and subsystems necessary to supply the control system with necessary effort (action) to make the plant behave in the desired manner.
- Sensor system. All measurement equipment, with hardware and software, including signal communication equipment, that supply the control system with information and corrections necessary to perform the required control and monitoring function of the application.
- Control computer system. A system consisting of at least one computer or processor with CPU processing and input/output (I/O) capacity, and operator stations.

In hybrid electric vehicles, many advanced control systems are required. It must control the power units, choosing between the internal combustion engine and the electric motor, and manage the energy storage devices, for instance when charging the battery from regenerative braking. Moreover, the control systems must also regulate, among other things, the performance of engine and transmission.

The powertrain system of a hybrid electric vehicle includes

- Internal combustion engine
- Energy storage device (e.g. battery pack, capacitor)
- Electric machines
- Transmission
- Driveshafts

- Differential
- Wheels

A combined electronic control unit (ECU) consisting of the engine ECU and the transmission ECU is involved in the powertrain system. In HEVs, in addition to the previous ones, a hybrid ECU is necessary to decide how to use the engine and the electric motor.

5.3. Control architecture.

The architecture of the powertrain control system is shown in Figure 3. It consists of four main parts:

- *Powertrain system*: components from the power generation to the transmission, and to the driving wheels.
- *Driver*: determine the level and rate of change of the steering wheel and pedals (accelerator and brake), shift operations, and mode settings to achieve the reference vehicle speed and position.
- *Control computer system*: distributed or centralized system of ECUs necessary to carry out the required control functions.
- *External information*: if it is activated, it uses available external information, such as traffic information, road topography maps, global positioning system (GPS) devices, information of current faults used to make autonomous decisions in the fault-tolerant control systems.

5.3.1. Hardware-in-the-loop simulation.

A HIL simulator can be defined as “a setup that emulates a system by immersing faithful physical replicas of some of its subsystems within a closed-loop virtual simulation of the remaining subsystems”, [4]. Closed-loop simulation refers to a bidirectional interaction between the virtual and physical subsystems.

The software-to-hardware (SW-to-HW) ratio can characterize a HIL simulator. It relates the number of components or subsystems of the real control system that are coded and simulated in the HIL SW versus the amount of HW kept in-the-loop [28]. On the one side, if all subsystems are implemented in SW, it becomes a pure simulation study, as e.g. in a software-in-the-loop simulator (SILS). The real control system is on the other side, where no SW is implemented in the simulator.

For example, a HILS for type-approval-test method of heavy-duty HEVs is used in [15]. HILS SW reproduces by means of a mathematical model the vehicle, motor/generator, transmission, battery, and the driver. Input and output signals of the model are connected to the electronic computer systems (ECUs).

Also in [15], a SILS is performed at an early stage of the method to verify a standard HEV model and its model operation environment (software and hardware). In the SILS, the standard HEV model incorporates a reference ECU model.

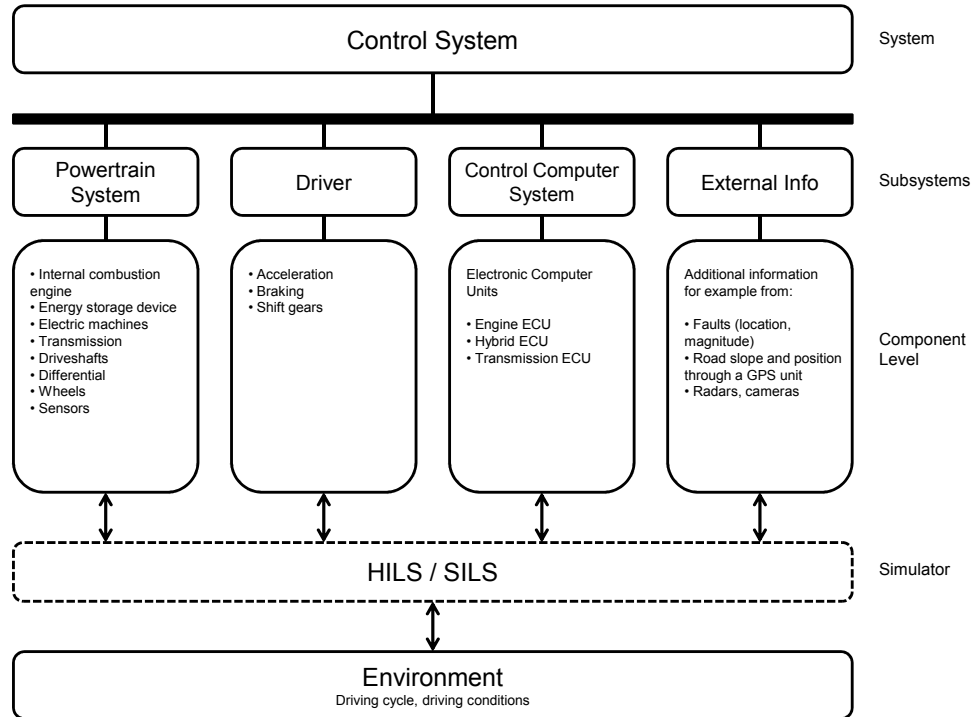


FIGURE 3. Control system architecture

5.3.2. Interface.

Interfaces are the signals that are communicated between the different part of the powertrain control system. The interface represents the I/O to the different parts and could be normal hardware I/O interfaces (analog, digital, serial protocol), network protocol (CAN, ethernet), dedicated HIL test I/O interfaces or simulated versions of the previously mentioned interfaces. The I/O interfaces should allow simulated sensor, actuator feedback and power signals to the parts to be transmitted.

5.4. Safety system - Faults and failures.

The IFAC, the International Federation of Automatic Control, Technical Committee on Fault Detection, Supervision and Safety of Technical Processes (SAFEPROCESS) has made an effort to come to accepted definitions, and some definitions can be found, for example in [8, 7]. In those works, a fault is defined as:

DEFINITION 5.1. A fault is an unpermitted deviation of at least one characteristic property or parameter of the system from the acceptable, usual, standard condition.

In general, the term fault differs from the terms failure and malfunction. In contrast to the term fault, the notion of a failure suggests a complete breakdown of a system or a component, and the inability of it to accomplish its function.

DEFINITION 5.2. A failure is a permanent interruption of a system's ability to perform a required function under specified operating conditions.

DEFINITION 5.3. A malfunction is an intermittent irregularity in the fulfillment of a system's desired function.

5.4.1. *Faults for HEVs.*

HIL testing should test a safety-critical control system with respect to a manageable set of faults.

	Faults	Components
List of typical faults in HEV:	Wear	Energy Storage
	Bias	Energy Storage (SOC estimator)
	Wear	Transmission

6. Modeling

In this section a set of models that are suitable for use in automotive HIL applications are presented. The models are similar to the models used in [15], the main difference is that some of the models are extended with some dynamics in order to better capture the reality or include some important dynamics that are either neglected or not defined.

6.1. Modeling philosophy.

The modeling philosophy that is suitable for HIL applications is called forwarding, which means that the powertrain is described by models described by differential equations. This makes it possible to take into account dynamic effects such as engine speed-up and vehicle inertia. The other alternative, called backwarding, is usually based on quasi-static models. Such descriptions can be simulated much faster, but the result does not describe transient effect in a good way and feedback control loops can not be used with this type of description.

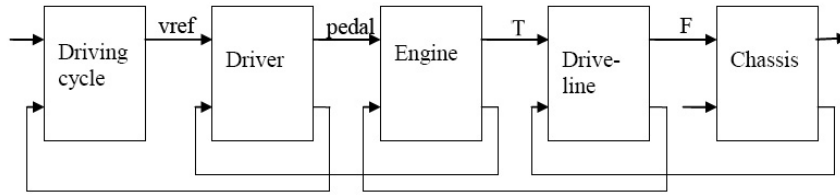


FIGURE 4. Model structure for a powertrain model using forwarding.

Dynamic simulation or forwarding is outlined in Figure 4. The name forwarding comes from the fact that the current subsystem is using information determined (feedback signals) in subsystems *in front of the current subsystem*. The idea is to use a driving cycle, to set the desired vehicle velocity for the driver. The driver utilizes the desired velocity and the current velocity in order to command the vehicle by using the pedals. The driver is usually a PID controller. In turn, the engine uses command signals from the driver and a control system (not shown in Figure 4) and feedback signals from the driveline in order to calculate the current engine states, and so on.

6.2. Model interfaces.

For individual component studies the modeling language (MATLAB, FORTRAN, GTPower etc) is not critical as long as it is well documented, but for complete vehicle studies it is preferable if the component models can be connected together in a straightforward manner to form a complete vehicle model. In order to achieve this, the model interfaces between the powertrain components needs to be determined. In a previous project, CAPSim, see e.g. [6], a set of conventions for dealing with this problem was designed. The ideas from the previous project has been used as an inspiration in this project.

It is important to define good interfaces capturing all necessary information shared by the different objects. The idea is to use kind of a port-based modeling paradigm, see [20]. The communication signals between the different objects are physical signals (or ports), like electric wires, mechanical joints etc. The interfaces or connectors are based on energy flow to and from the component or through a port. A port is characterized by an *across* and a *through* variable, also known as *flow* and *effort* variables in Bond Graph modeling, see [22]. The interfaces are a key to exchangeability of component models. By using a port-based approach the same physical components can be simulated at different levels of complexity without modifying the system level model, i.e. models are exchangeable.

For automotive powertrains, four (five) different categories of models are necessary, the categories are: electrical, mechanical (rotational and translational), chemical and (fluid). The table below shows the port signals.

	Electrical	Mechanical (rotational, translational)	Chemical
Flow	Voltage [V]	Torque [Nm], Force [N]	Specific energy [J/kg]
Effort	Current [A]	Speed [rad/s], Velocity [m/s]	Mass flow [kg/s]

The port-based modeling paradigm is complemented with control signals, for making it possible to control each component.

6.3. Implementation.

MATLAB and Simulink offer extensible support for real-time simulation of models and since MATLAB and Simulink is used as the main simulation tool at Volvo Powertrain and Scania (industrial partners), the port-based conventions has been implemented in MATLAB/Simulink as generic component structures. The component structures or models have connectors for mechanical, electrical, chemical or fluid power. These model structures include electrical-to-electrical components, such as power electronics, electrical energy buffers, and mechanical-to-mechanical components, such as transmissions, clutches, planetary gears etc. Also models for energy conversion from one domain to another have been implemented, in this way models for electric motors and combustion engines can be modeled.

The outline of the structure for three different components is presented in Figure 5a-c. Some components have control interfaces and some do not. What control signals that should be sent to each component needs to be specified. There also exist components which converts energy from one domain to another, e.g. electric motors and combustion engines, see Figure 5d.

In the component structure the (flow,effort)-pair is cross-wise coupled in the Simulink implementation, meaning that the input energy flow to a component consists of one input signal and one output signal, and the output energy flow also consists of one output and one input signal. For the electric converter model in Figure 5b, the input energy flow is (`e1 power in [V] × e1 fb out [A]`) and the output energy flow is (`e1 power out [V] × e1`

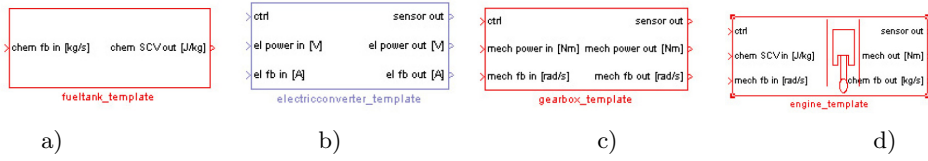


FIGURE 5. Component structure for some of the components.

$fb\ in\ [A]$). Note that the unit for energy flow is Watts [W] (power), this means that the efficiency of each component can easily be determined.

Interfaces for the following components have been defined:

- Chassis
- Clutch
- Driver
- Electric machine (incl. power electronics)
- Energy storage (Battery and Super capacitor)
- Gearbox (Automatic and Manual)
- Internal combustion engine
- Mechanical connection (Planetary gears or Other types)

The models can be connected together to form a powertrain. Depending on how the models are connected different powertrain configurations can be formed. Figure 6 shows a parallel powertrain, a so-called mild hybrid parallel or pre-transmission parallel hybrid powertrain.

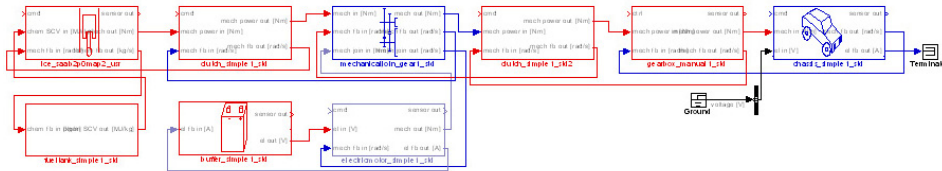


FIGURE 6. Parallel hybrid powertrain using the proposed model structure.

6.4. Component models.

In this subsection the component models that can be connected together to form a number of hybrid electric vehicles will be introduced and described.

The following naming convention for internal variables and variables shared has been used:

Component_Description_Unit

where **Component** is the component category, e.g. Clutch, Engine etc, **Description** is a description of the signal, e.g. Actual torque **ActTorque**, Measured voltage **SenVoltage** and **Unit** is the unit of the signal in SI-units, e.g. Nm, V, A etc.

6.4.1. Chassis.

A basic model of the vehicle, where the vehicle is represented as an inertia. The model computes the vehicle speed given propeller shaft torque

and brake torque. The model considers rolling- and air drag resistance and takes the road slope into account.

The basic principle is that the input torque T_{in} goes through a gear reduction (final gear) with ratio r ,

$$(6.1) \quad T_{drive} = T_{in}r$$

The drive torque T_{drive} is counteracted by the brake torque T_{brake} and the resulting torque turns into a drive force through the wheels with radius r_{wheel} ,

$$(6.2) \quad F_{drive} = \frac{T_{drive} - T_{brake}}{r_{wheel}}$$

and acts on the road to drive the vehicle forward. The force acts towards forces which models the aerodynamical drag, rolling resistance and gravitational force

$$(6.3) \quad m_{vehicle}\dot{v}_{vehicle} = F_{drive} - F_{aero} - F_{roll} - F_{grav}$$

where $m_{vehicle}$ is the total mass of the vehicle and $v_{vehicle}$ is the vehicle speed. The aerodynamical drag force can be calculated as

$$(6.4) \quad F_{aero} = \frac{1}{2}\rho C_d A_{front} v_{vehicle}^2$$

where ρ is the air density, C_d is the drag coefficient and A_{front} is the frontal area of the vehicle. The rolling resistance is computed from the normal load as

$$(6.5) \quad F_{roll} = f m_{vehicle} g \text{sign}(v_{vehicle})$$

where f is the fraction of the normal load that corresponds to rolling resistance. The *sign*-function is included in order to handle the case of zero speed. If gravitational forces are considered then the rolling resistance becomes

$$(6.6) \quad F_{roll} = f m_{vehicle} g \text{sign}(v_{vehicle}) \cos(\alpha)$$

where α is the road slope. The gravitational force is

$$(6.7) \quad F_{grav} = m_{vehicle} g \sin(\alpha)$$

Furthermore, the total mass of the vehicle, $m_{vehicle}$, includes the inertial loads from the powertrain components.

Appendix A.1 shows the Simulink model implementation and data for the component.

6.4.2. Clutch.

A simple model of a clutch. The working principle behind the clutch is that if the clutch is closed then the input torque T_{in} is transferred to the output torque T_{out} . If the clutch is open, the input shaft spins freely and no torque is transferred.

The equations of motion for the clutch, with notation according to Figure 7:

$$(6.8) \quad \begin{aligned} J_1 \dot{\omega}_1 &= T_1 - T_c \\ J_2 \dot{\omega}_2 &= T_c - T_2 \end{aligned}$$

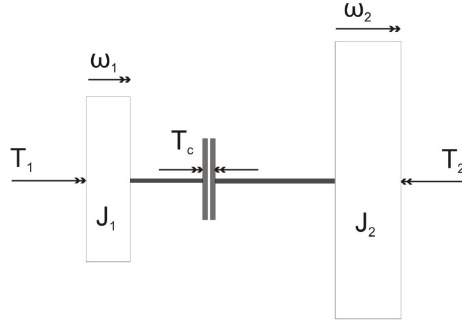


FIGURE 7. Simple clutch model, from [12].

The clutch is working in three different phases; closed, open or in between closed and open, slipping. When the clutch is open, $T_c = 0$, and during slipping

$$(6.9) \quad \begin{aligned} T_c &= T_{ref\ torque} \text{sign}(\omega_1 - \omega_2) \\ \omega_1 &= \omega_2|_{t=0} + \int_0^t (T_1(\tau) - T_c(\tau)) d\tau \end{aligned}$$

where $T_{ref\ torque}$ is the desired torque that is to be transferred through the clutch. When the clutch is closed the $T_1 \equiv T_2$.

Appendix A.2 shows the Simulink model implementation and data for the clutch.

6.4.3. Driver.

The driver is modeled using a simple PI-controller. It takes the reference speed from the driving cycle and compares it to the vehicles actual speed. If the vehicle's speed is too low it uses the accelerator pedal to demand an acceleration, and vice versa if the vehicle's speed is too high, the driver uses the brake pedal to demand a deceleration of the vehicle. The driver interface is made for also controlling the gear shifting and clutch pedal if a manual transmission is used.

Appendix A.3 shows the Simulink model implementation and data for the component.

6.4.4. Electric machine.

An electric machine can generally be divided into two parts, the stator and the rotor. The rotor is the rotating part of the machine. The electric machine is modeled using maps. The main reason is that these maps are rather easy to obtain, the model representation becomes accurate, and several different types of machines can be characterized, such as DC-motors, PMSMs and induction machines.

The electric machine dynamics is modeled as a first order system

$$(6.10) \quad \dot{T}_{em} = -\frac{1}{\tau_1} (T_{em} - T_{em,des})$$

where T_{em} is the machine's torque, $T_{em,des}$ is the desired torque and τ_1 is the electric machine's time constant. The losses for the electric machine and the power electronics are mapped as function of torque and speed

$$(6.11) \quad P_{loss} = f(T_{em}, \omega_{em})$$

The efficiency of the electric machine can be calculated as

$$(6.12) \quad \eta_{em} = \frac{T_{em}\omega_{em}}{T_{em}\omega_{em} + P_{loss}}$$

and the current needed, can be calculated as

$$(6.13) \quad i = \frac{T_{em}\omega_{em}}{\eta_{em}u}$$

where i is the current and u is the battery voltage.

The model is complemented with a simple thermodynamical model. The losses in the electric machine is assumed to be resistive losses.

$$(6.14) \quad P_{loss} = Ri^2$$

where R is the machine's internal resistance and i is current through the windings of the stator. The losses are transformed into heat, heating the machine. The temperature change from the machine, $\Delta\vartheta$, can be determined as

$$(6.15) \quad \Delta\tau = P_{loss}(R_{th,1} + R_{th,2})$$

where $R_{th,1}$ is the machine's thermic resistance from rotor to stator and $R_{th,2}$ is the thermic resistance from stator to the machine. These thermic resistance are sometimes given in data sheets for the electric machine or can rather easily be determined from experiments.

The thermodynamical model is complemented with a first order filter to incorporate the dynamics.

$$(6.16) \quad \begin{aligned} \vartheta_{em} &= \frac{1}{1+s\tau_{em,heat}}(\vartheta_{em,surr} + \Delta\vartheta) \\ &= \frac{1}{1+s\tau_{heat}}(\vartheta_{em,surr} + P_{loss}(R_{th,1} + R_{th,2})) \end{aligned}$$

where $\tau_{em,heat}$ is the time constant for the thermal mass of the machine, $\vartheta_{em,surr}$ is the machine's surrounding temperature (cooling media temperature) and ϑ_{em} is the machine's temperature.

The electric machine can be torque or speed controlled. The physical model is complemented with a local controller. The speed controller is a PI-controller, while the torque controller is an open-loop feedforward controller.

Appendix A.4 shows the Simulink model implementation and data for the component.

6.4.5. Energy storage.

The energy storage or buffer is modeled as a simple battery using a resistor model, see Figure 8. The battery voltage can be determined from Kirchhoff's law as

$$(6.17) \quad u = e - Ri$$

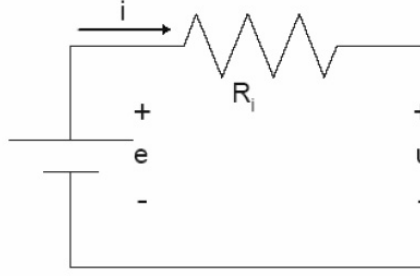


FIGURE 8. Simple battery model.

The open circuit voltage e and the internal resistance R_i are depending of energy level in the battery, state-of-charge SOC , or actually depth-of-discharge, $DOD = 1 - SOC$. The dependency is modeled using tabulated values in maps. The battery is scalable via the number of cells used.

Figure 9 shows the comparison of the battery model and a more detailed battery model provided by Volvo Powertrain, denoted the reference battery model. From the simulation result it can be seen that the state-of-charge follows good, while there is a discrepancy in voltage, some voltage dynamic is neglected.

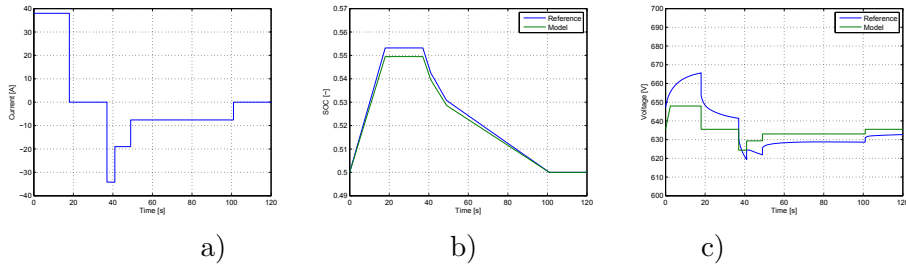


FIGURE 9. Battery test. a) Input signal to battery (follows a standard test cycle to battery testing. b) State-of charge for the model and the reference battery model. c) Terminal voltage for the model and the reference battery model.

Appendix A.5 shows the Simulink model implementation and data for the component.

6.4.6. Gearbox.

The gearbox is modeled as two gears in contact, with a ratio of r_{gear}

$$(6.18) \quad \begin{aligned} T_{out} &= T_{in} r_{gear} \\ \omega_{out} &= \omega_{in} / r_{gear} \end{aligned}$$

Losses for the gearbox is considered to be torque losses, meaning that T_{out} is actually calculated as

$$(6.19) \quad T_{out} = \eta_{gear} T_{in} r_{gear}$$

Losses are given for each gear.

The total gearbox inertia depends on the active gear:

$$(6.20) \quad J_{out} = J_{in} r_{gear}^2 + J_{gearbox}$$

The model also includes a clutch in order to get a torque interrupt. The transmission block can handle up to 12-speed gearboxes. The number of gears is set by a parameter.

Appendix A.6 shows the Simulink model implementation and data for the component.

6.4.7. Internal combustion engine.

The internal combustion engine is modeled in a similar way as the electric machine. Unfortunately, the torque dynamics cannot be represented by one linear first order system, because of the turbo dynamics. The engine torque build-up is represented as

$$(6.21) \quad \dot{T}_{ice} = -\frac{1}{\tau_2(\omega_{ice})}(T_{ice} - T_{ice,des1}(\omega_{ice})) + T_{ice,des2}(\omega_{ice})$$

where T_{ice} is the engine torque, $T_{ice,des}$ is the desired torque and τ_2 is the engine's time constant. The time constant and the desired torque is dependent on engine speed, ω_{ice} . It should be noted that the desired torque is divided into two parts, one dynamic term, $T_{ice,des1}(\omega_{ice})$, and one direct feed through term, $T_{ice,des2}(\omega_{ice})$. Furthermore, if the desired torque is less than the direct feed through term, no dynamic term is needed to capture the engine torque response, i.e. the torque is instantaneous. The time constant and the division of the two parts of the desired torque is mapped as function of speed. Figure 10 shows the comparison of the engine torque model, the "true" engine response (simulated engine response provided by Volvo Powertrain) and the engine response from the model proposed in the Japanese method.

The model is also complemented with a simple thermodynamical model. As the engine is equipped with its own cooling system, the thermodynamical model for the engine is only covering the heating of the engine, i.e. when the engine is turned on, the engine body is heated to its normal operating temperature. The dynamics is modeled as a first order system, but do not consider the use of the engine.

$$(6.22) \quad \vartheta_{ice} = \frac{1}{1 + s\tau_{ice,heat}}(e_{on}\vartheta_{ice,nom} + \vartheta_{ice,surr})$$

where $\tau_{ice,heat}$ is the time constant for the thermal mass of the engine, $\vartheta_{ice,surr}$ is the engine's surrounding temperature, e_{on} is a boolean value stating whether the engine is on or off, $\vartheta_{ice,nom}$ is the engine's normal operating temperature and ϑ_{ice} is the engine's temperature. With this simple model, cold starts (from +25°C) can be handled.

The internal combustion engine can be torque or speed controlled. The physical model is complemented with a local controller. The speed controller is a PI-controller, while the torque controller is an open-loop feedforward controller.

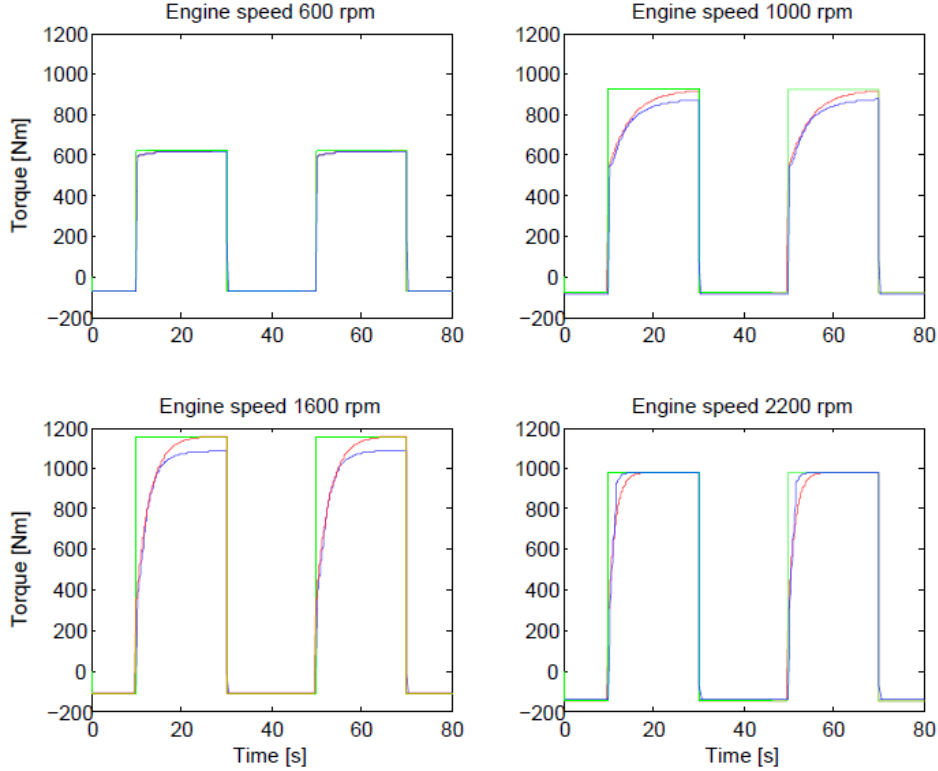


FIGURE 10. Torque responses for different speeds. The blue line corresponds to the true engine response, the red line corresponds to the modeled engine response and the green line corresponds to the engine model using the proposed model in the Japanese method.

Appendix A.7 shows the Simulink model implementation and data for the component.

6.4.8. Mechanical connection.

This component is used to connect two input shafts. One shaft goes through and is connected to a joining input shaft through gears. The output torque is calculated as

$$(6.23) \quad T_{out} = T_{in,1} + T_{in,2}r$$

where $T_{in,1}$ and $T_{in,2}$ are the torques on the input shafts respectively and r is the gear ratio. The speed on the output shaft is equal to the speed on the first input shaft, while speed on the second input shaft is reduced via the gear ratio. The component's inertia J_{join} is added to the total inertia as:

$$(6.24) \quad J_{out} = J_{in,1} + J_{in,2}r^2 + J_{join}$$

Appendix A.8 shows the Simulink model implementation and data for the component.

6.5. Control system interface.

For controlling the models a number of control signals need to be sent to the components, these signals are usually determined from measurements of important variables in the components in some sense. A general interface of control signals and sensor signals that are necessary in order to be able to simulate the models has been determined. The control signal interface proposed is presented in Appendix B.

7. Simulation

In this section the set of models that were presented in the previous section are connected together with a control system. The idea is to do a SIL simulation test, in order to get more familiar with the Japanese method and the problem that arises when connecting the simulation model with a “real” controller. Another reason for doing this, is to verify that the models are representative and work properly.

7.1. Case study: Hybrid distribution truck.

This case study illustrates how the component models can be used in simulation. The idea is to do a SIL simulation of a hybrid distribution truck, modeled using the models presented in the previous section and controlled by a controller designed by Volvo Powertrain. It should be noted that this case study is pure simulation and does not include any attempts for HILS.

The hybrid distribution truck has a pre-transmission parallel hybrid configuration, meaning that the electric machine is mounted between the combustion engine and the transmission. The vehicle is equipped with a 7 liter heavy diesel engine, a 45 kW synchronous permanent magnet electric machine, a Li-ion battery system with a capacity of 2.5 kWh and a 12-speed automated manual transmission.

7.1.1. Modeling.

The modeling of the truck is made by connecting the different models presented in the previous section. The Simulink model of the truck, called the certification model, is shown in Figure 11. All parameters for the truck is presented in Appendix C.2. The input and output signals from the model is presented in Appendix B.

7.1.2. Software-in-the-loop simulation.

The model is complemented with a controller, provided by Volvo Powertrain, to make the system complete. The controller is a software version of a controller developed in a concept study. The controller contains all functionality to control the hybrid vehicle, including a decentralized control architecture. Due to intellectual property rights it is not possible to present the controller.

In order to make the controller compatible with the specified model interface (Appendix B), two interface blocks are needed to be developed. One for the input signals to the controller and one for the output signals from the controller. The idea of the interface blocks are to convert the signals between the controller and the truck model. A conversion can be to assure correct units of the signals, for example rad/s to rpm etc. The interface blocks can also generate additional signals needed for making the control system to work. The complete system is presented in Figure 12.

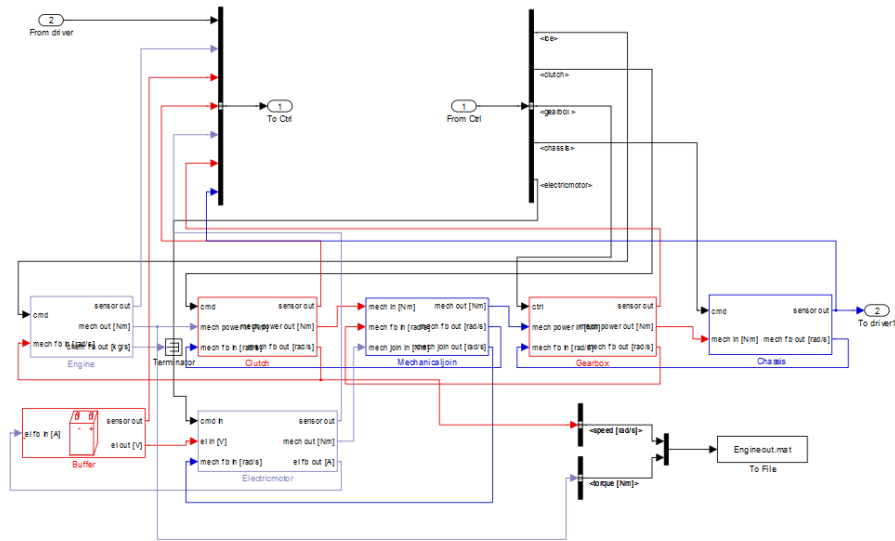


FIGURE 11. SHC model in Simulink of hybrid distribution truck.

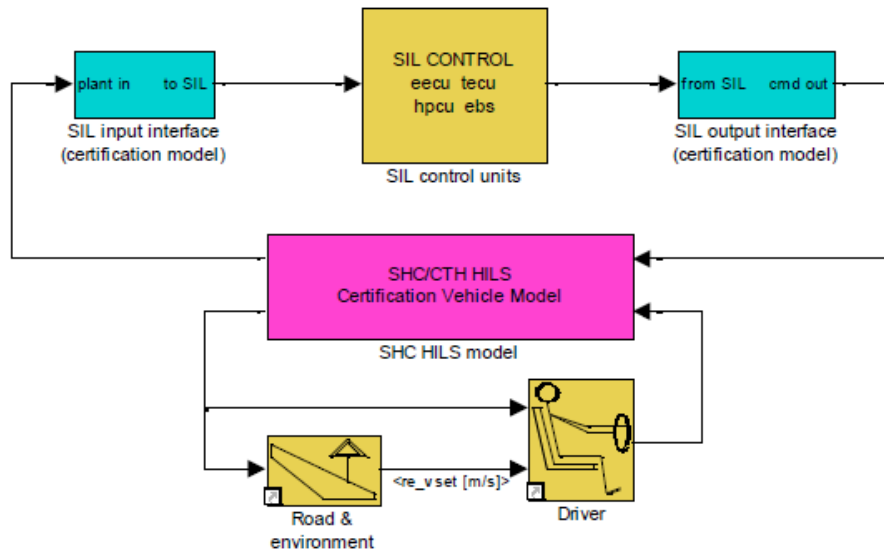


FIGURE 12. SILS with SHC model.

Another use of this case study is, as mentioned earlier, to investigate if the proposed models are representative and work properly. By connecting the same controller to a, by Volvo Powertrain, verified truck model, the qualitative behavior of the truck model can be investigated. The top view of the reference system model is presented in Figure 13.

7.1.3. Results.

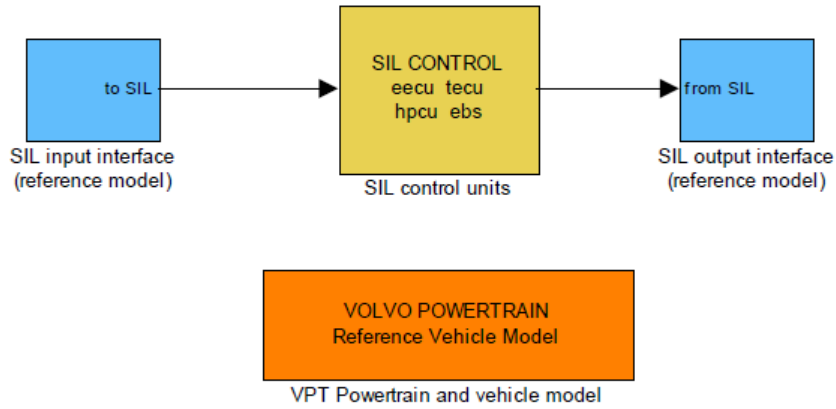


FIGURE 13. SILS with Volvo Powertrain model.

The models are simulated in Simulink using different tests and driving cycles. It is difficult to make a direct comparison of the different simulations, because the input signals are not identical. This results from the use of a driver model and the fact that small model differences will cause the controller to make actions at different times and with different level of amplitude. As a result from this only a qualitative analysis of the models can be done.

The first test is a full throttle test. The simulation result is presented in Figure 14. From the simulation result it can be seen that the response of the certification model is similar to the reference model. This indicates that the

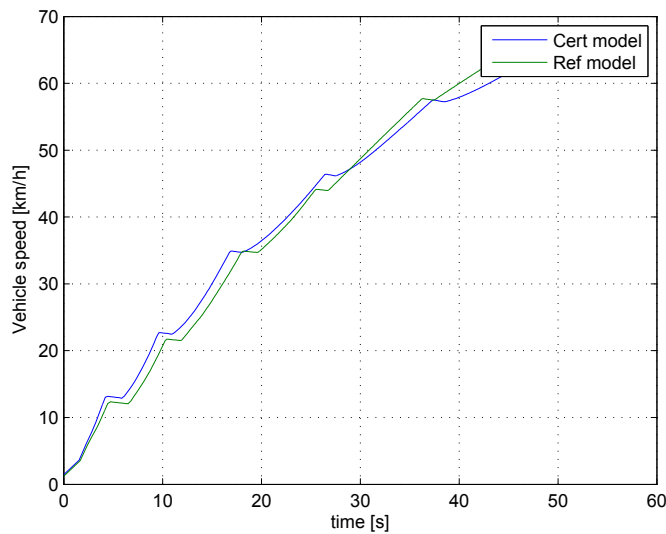


FIGURE 14. SILS with Volvo Powertrain model.

Colchester-Clophill	Unit	Ref model	Cert model
General:			
Fuel consumption	l/100 km	18.8	19.1
Travel time	min	135	134
Traveled distance	km	112.9	112.9
Average speed	km/h	50.0	50.5
Transmission:			
Number of gearshifts		1438	1441
Gearshifts per 100 km		1273	1276
Energy storage:			
Initial SoC	%	50	50
Final SoC	%	48.5	41.6
Delta SOC	%-units	-1.5	-8.4
Max SoC	%	53.4	51.8
Min SoC	%	33.8	34.1
Energy throughput	kWh/h	5.6	3.3

TABLE 1. Statistics from the Colchester-Clophill driving cycle

The second test is a driving cycle test. The used driving cycles are Volvo Powertrain specific driving cycles, one corresponding to real driving between Colchester and Clophill and one corresponding to driving in Gothenburg, the bus route 85 in Gothenburg. The two driving cycles correspond to urban and non-urban driving (Colchester-Clophill) and urban driving (bus route 85). The simulation results for the Colchester-Clophill driving cycle are presented in Figure 15. Figure 15a shows the simulation results from the reference model and Figure 15b shows the simulation results from the certification model. The simulation results for the bus route 85 are presented in Figure 16. Figure 16a shows the simulation results from the reference model and Figure 16b shows the simulation results from the certification model.

The overall picture of the two driving cycle simulations is good. The simulations run smoothly and the time to do a simulation for the reference and the certification model are equivalent. Furthermore, looking at the engine's operating points, Figure 17, it can be concluded that there are similarities between the usage of the engine.

In order to compare the result a bit more in detail, a part of the Colchester-Clophill driving cycle is zoomed in, the first kilometer of driving, see Figure 18. From the simulation result it can be seen that the qualitative behavior of the certification model is close to the behavior of the reference model, this indicates that the certification model has no major errors and represents the hybrid truck (the reference model) in a reasonable manner.

Some statistical figures from the two driving cycle simulations are presented in Table 1 and Table 2. From the statistical figures it can be concluded that the certification model does not use the electrical capability as much as the reference model, this might be due to the unmodeled dynamics in the battery.

Bus route 85	Unit	Ref model	Cert model
General:			
Fuel consumption	l/100 km	34.5	39.0
Travel time	min	75	73
Traveled distance	km	22.8	22.8
Average speed	km/h	18.3	18.9
Transmission:			
Number of gearshifts		1868	2060
Gearshifts per 100 km		8165	9004
Energy storage:			
Initial SoC	%	50	50
Final SoC	%	51.7	49.3
Delta SOC	%-units	1.7	-0.7
Max SoC	%	59.8	58.0
Min SoC	%	42.4	42.7
Energy throughput	kWh/h	10.4	6.9

TABLE 2. Statistics from the bus route 85 driving cycle

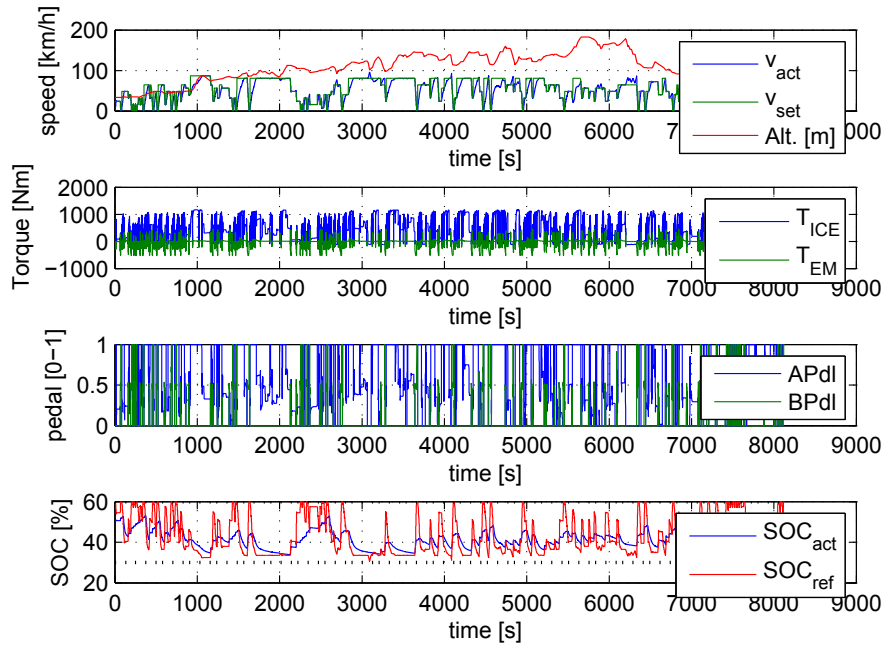
Further studies and analysis are required in order to make a more complete conclusion regarding the accuracy of the model and if further improvements are needed.

7.2. Concluding remarks.

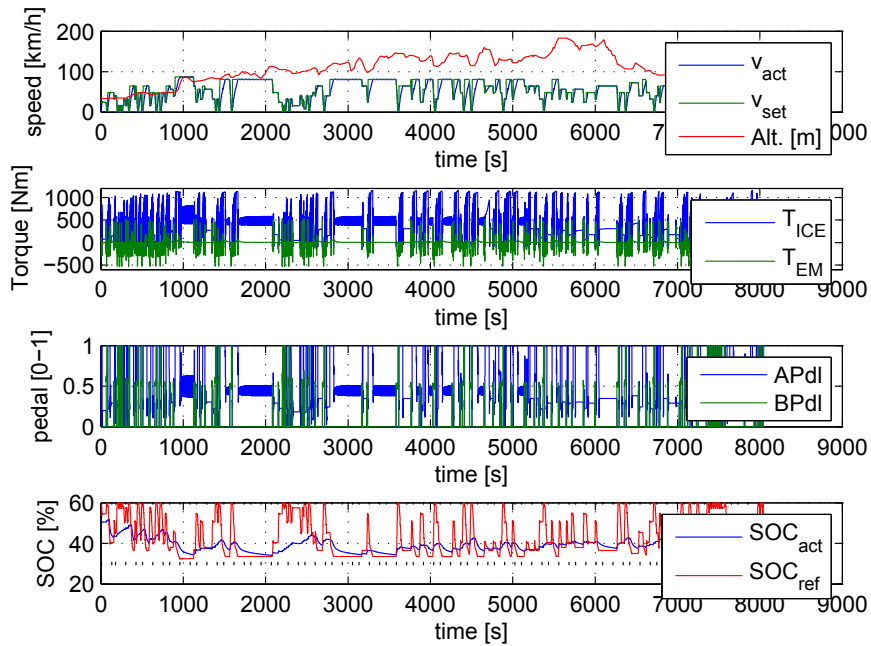
From the case study it can be concluded that it is not that easy to connect a control system to the HILS model. Even though, the interface signals are well defined, the development of the interface blocks takes some time. The main problem during the case study was to develop and debug the interface blocks, making sure that the controller and model communicate with the correct signals. It is more or less a prerequisite to have a reference model in order to debug the interface blocks. It should be noted that no additional dynamics or algorithms were added in the interface blocks, only rewiring, scaling and constants were added, but on the other hand no diagnostic functionality is included in the used controller.

Furthermore, it can be concluded that the proposed models are representative and captures the powertrain characteristics in an acceptable way, but a more thorough analysis is required in order to finalize this statement.

From a more general point of view, if the problem with distributed control systems can be solved by using pure simulation, there are other possibilities that come with this. For example, the simulation does not need to run in real time, this will reduce the simulation time a lot, in the case study the simulation time was around 10-15 times faster than real time. This makes it possible to do other studies during the same time as a HIL simulation, for example study how sensitive the controller is for battery degradation or errors in SOC estimation.

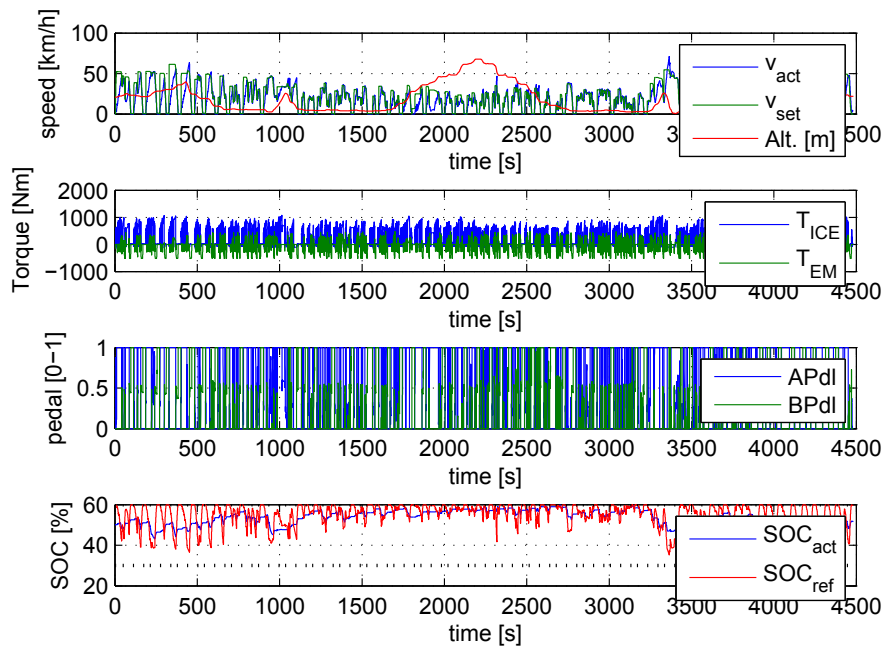


a)

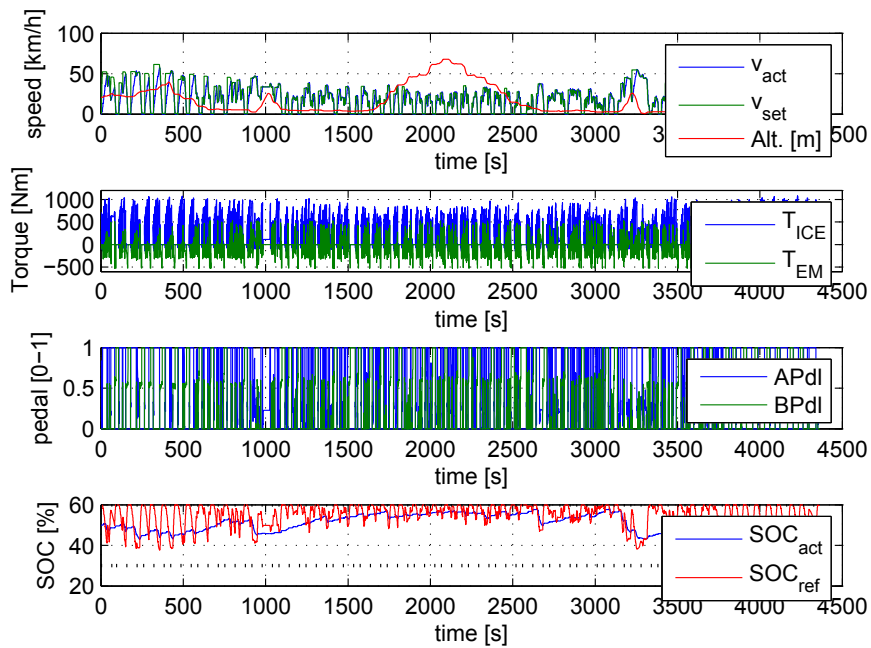


b)

FIGURE 15. Simulation result from the Colchester-Clophill driving cycle. a) corresponds to the reference model and b) corresponds to the certification model.



a)



b)

FIGURE 16. Simulation result from the bus route 85 driving cycle. a) corresponds to the reference model and b) corresponds to the certification model.

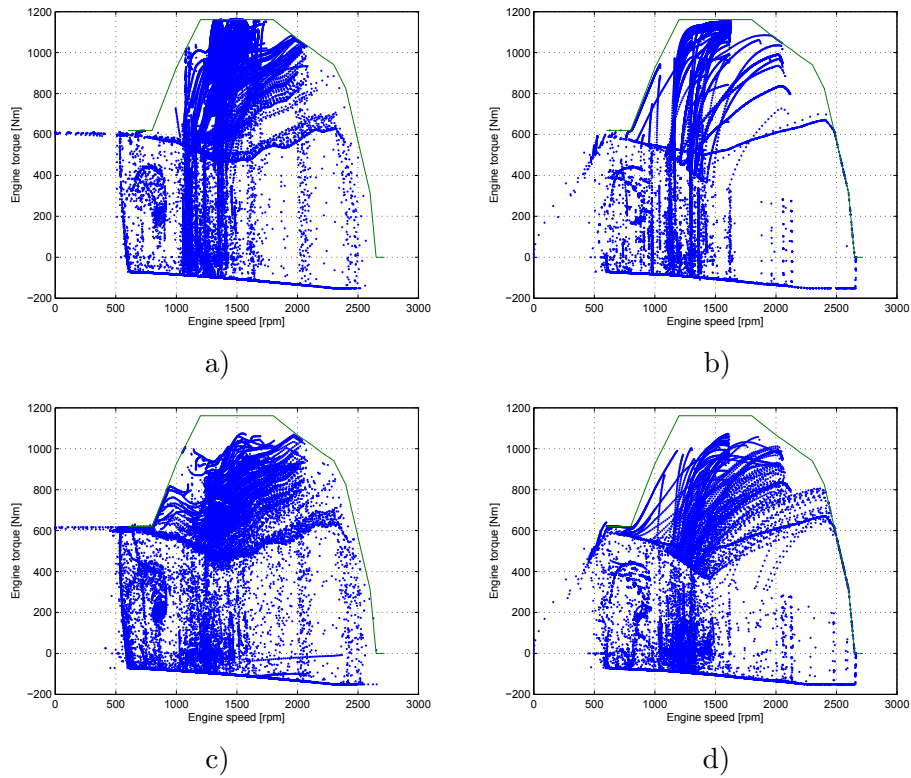


FIGURE 17. Simulation result a) Engine torque-speed diagram for the reference model from the Colchester-Clophill driving cycle. b) Engine torque-speed diagram for the certification model from the Colchester-Clophill driving cycle. c) Engine torque-speed diagram for the reference model from the bus route 85 driving cycle. d) Engine torque-speed diagram for the certification model from the bus route 85 driving cycle.

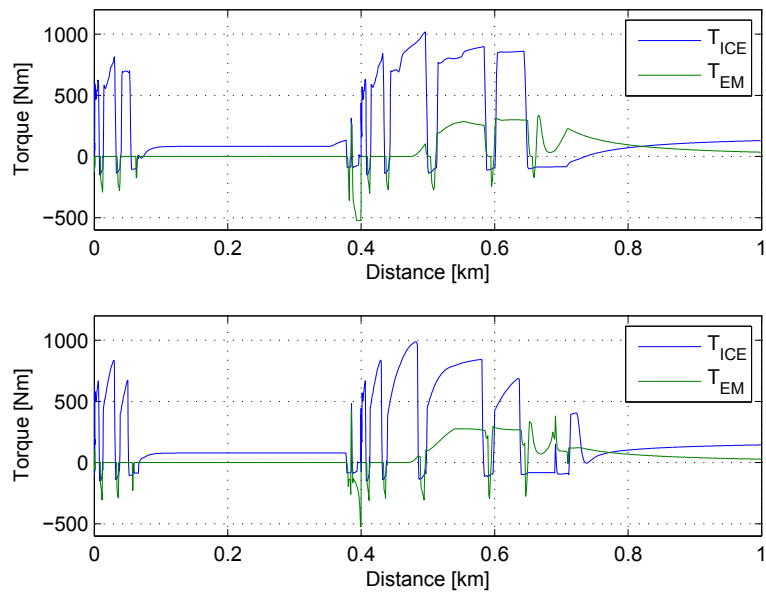


FIGURE 18. Simulation result from the first kilometer of driving of the Colchester-Clophill driving cycle.

8. System level verification

The importance of verification, both the component models and the system model is highlighted in both [21] and [15]. If the simulation models are not adequately verified, the results lack credibility and certification can not be performed. This is of course true, but the accuracy for verification on system level stated in [15], seems a little bit too high. By specifying such high accuracy, the need of verification on system level will be huge. Intuitively there is a great need for dynamometer testing, since the choices of parameters is huge for heavy-duty HEVs. If the need of system verification can be reduced in some way, the method becomes even more interesting for HEV certification. Ideally, only component verification should be enough! This ideal situation might not be fulfilled but it might be possible to reduce the need of verification.

For component level verification the level of accuracy stated in [9] is good. The way to measure and verify the component models are well proposed in [9], some modifications are thus needed in order to incorporate the dynamics that is lacking.

One idea in order to reduce the need of verification on system level is to decrease the accuracy level. This can be motivated from the work done within UNECE with the development of the WTVc driving cycle and the certification method. The method is presented in [29] and outlined in Figure 1. The basic idea is, as presented in Section 2, to transform the WTVc cycle into an engine test cycle (WHTC). The WHTC cycle is defined in terms of normalized engine speed and load, and is developed using a generic powertrain model. The normalized engine speed, ω_{norm} , and load, T_{norm} , points are then “scaled and stretched” for the engine that is to be certified according to:

$$(8.1) \quad \begin{aligned} \omega_{eng} &= \omega_{norm}(0.6n_{lo} + 0.2n_{hi} + 0.2n_{prefs} - n_{idle})/0.5363 + n_{idle} \\ T_{eng} &= T_{norm}T_{max}/100 \end{aligned}$$

where n_{lo} , n_{hi} , n_{prefs} and n_{idle} are speed values for the specific engine and T_{max} is the maximum engine torque, see Figure 19. The certification of emissions from conventional heavy-duty vehicles is then normally done in an engine test bench, where the engine is running under predefined load/speed conditions. The verification on system level is little or none, since a general powertrain model is used for all heavy-duty vehicles, ranging from 3 to 40 tonnes. To illustrate the flexibility of the method a simple example is used.

8.1. Example.

In this example the idea is to show the deviation between the engine speed points determined from the method presented above and engine speed points determine from simulation of a complete powertrain model. The vehicle in question as a conventional city-bus. The bus is modeled using the models presented earlier in this report. The Simulink model is shown in Figure 20.

The powertrain components are complemented with a controller unit in order to translate the driver’s commands into control signals for the powertrain components. The controller is a simple pedal map translating the

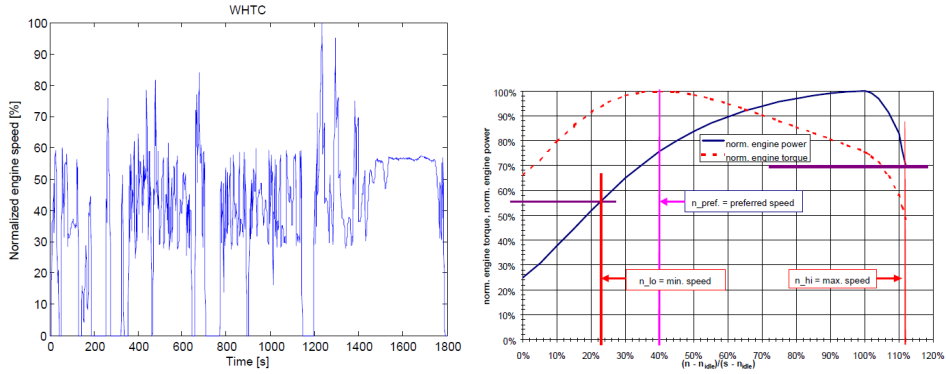


FIGURE 19. Characteristic plots for the certification procedure based on the WHTC driving cycle. The left figure shows the normalized engine speed as function of time. The right figure shows the specific values in the torque-speed diagram, which are important for scaling the normalized engine speed curve.

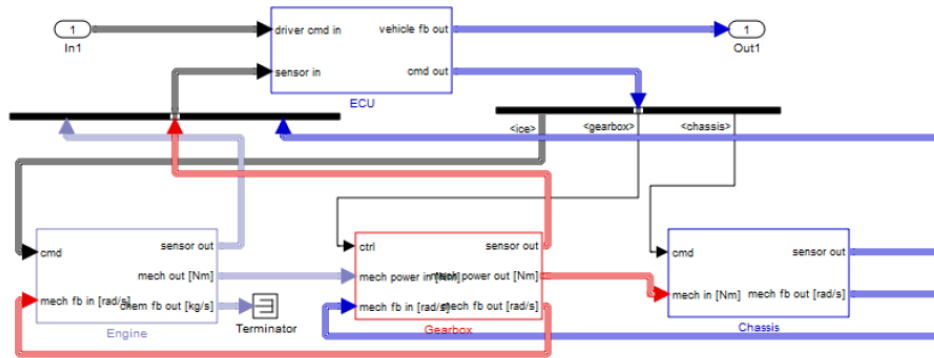


FIGURE 20. Simulink model representing a conventional city-bus.

accelerator pedal position into a desired engine torque. The engine is a 7-liter medium-heavy diesel engine, 300 hp, the gearbox is an automatic 6-speed gearbox and the total weight of the vehicle is 14 tonnes. All data used for simulation is presented in Appendix C.1.

Figure 21a shows the engine speed and load points using the standardized way for certify heavy-duty trucks, while the Figure 21b-d show the engine speed and load points from the simulated powertrain model with different gearshift points.

From the simulation it can be seen that the speed and load points differ quite a bit. One of the reasons is the gear shifting strategy, in other words the control strategy effects the operating points. In the simulated model a simple speed dependent gear shifting strategy is used, in Figure 21b upshift occur when engine speed reaches 1500 rpm and downshift when engine speed goes below 1200 rpm, in Figure 21c upshift occur when engine speed reaches 1600 rpm and downshift when engine speed goes below 1100 rpm and in

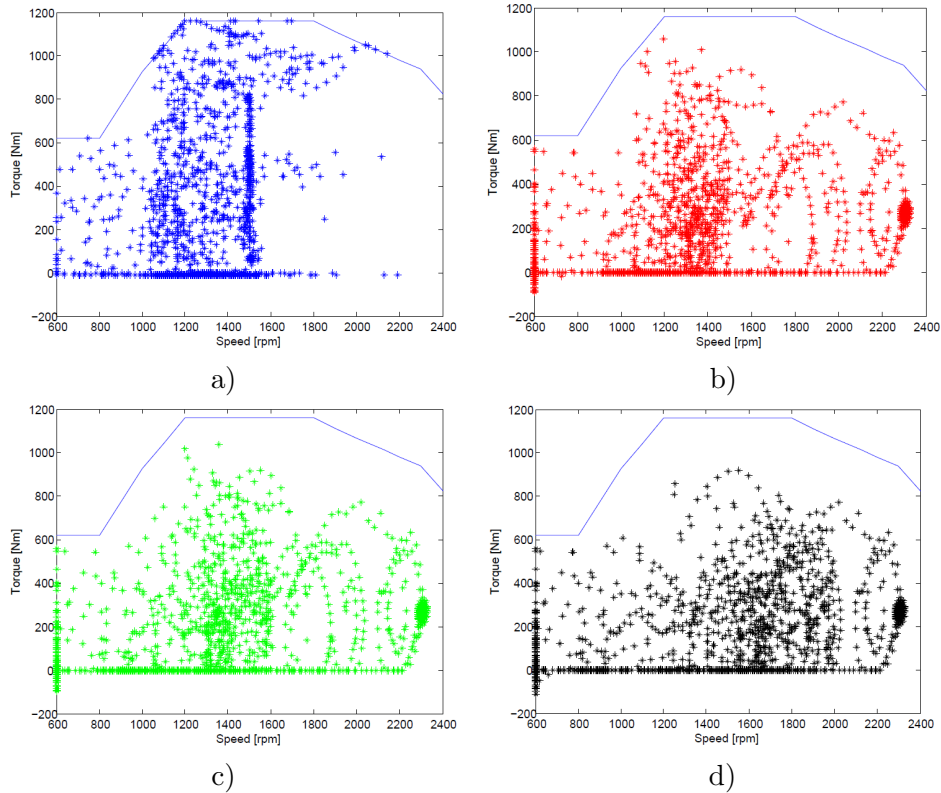


FIGURE 21. Torque-speed diagram. Operating points for emission certification. a) shows operating points from the normalized standard for WHTC, b-d) show operating points from simulation.

Figure 21d upshift occur when engine speed reaches 2000 rpm and downshift when engine speed goes below 1000 rpm.

8.2. Discussion.

In the Japanese method the verification level in *Step 5* is very tough, i.e. the difference between two different simulations should be small. On the other hand, in the certification procedure for a conventional vehicle there is a great freedom, as the simulated example shows. The system can be certified using the reference points in Figure 21a, but can run in reality according to the other figures in Figure 21. *It should be noted that this simplified example do not show the complete picture, but it gives some indication on that there is a possibility to reduce the level of verification.*

9. Future Work

From the work presented in this report, there is a need for further investigation on a number of points.

The first point that needs a more thorough investigation is the level of verification needed on system level. In the Japanese method the requirement on system level verification is tough, 3 times tougher than the level for a conventional vehicle. Furthermore, the organization ACEA has a vision for using HILS for certification of hybrid vehicles that only requires verification on component level. With these two opposite requirements/visions it is interesting to investigate the need of verification. One way to do this investigation could be to inject errors into the component models and see what the consequences are on system level (new engine load and speed point). The new engine load and speed points are then run in an engine test in parallel with the old, "true" engine load and speed points. The results on emissions are then compared and evaluated. It can also be concluded from the findings in the simulation case study, that there is a need to find ways to compare simulations not using exactly the same input trajectories, but similar input trajectories.

The second point to investigate is the need for using HILS compared to SILS. The problem with distributed control systems is that it is not that easy to connect them to the HIL system as it is with a centralized control system. Is it possible to use a SIL code version of the control system instead of using the actual controller hardware? Associated to this, there are a number of questions that need to be answered. Questions like, who's responsible for verifying that the SIL code is working in the same way as the actual controller hardware, should the SIL code be identical to the HIL code or is it possible to use a "light" version not including diagnostic functionality. Intellectual property issues need also to be worked out.

The third point to investigate is how the method should handle the fact that some hybrid vehicles are not charge sustaining, for example plug-in hybrids or special vehicles with auxiliary systems that use the electrical energy stored in the energy storage. An introduction of non-charge sustaining strategies, would probably lead to several modifications of the Japanese method.

A fourth point to continue working on is the definition and terminology standardization.

The last point to investigate is to look at the driving cycles. Is the driving cycle, WTVC etc, still representative when introducing hybrid vehicles or is it necessary to develop new driving cycles. This point is more of an international/governmental concern than in a small research project.

10. Conclusions

In this work, the method is studied in more detail and it can be concluded that the method is well proposed and coincide well with the method used for certification of conventional vehicles. The main difference is that for hybrid vehicles the control system is important to include when determining the operating points of the engine. The Japanese method incorporates the control strategy by using HIL, where the hardware is the electrical controller unit, including the hybrid control strategy.

This work has investigated the problem of using distributed control systems in the Japanese method and it can be concluded that it is not trivial to connect a distributed control system to the HIL system. First of all, if a distributed control system is to be connected and work in an appropriate way, almost the complete electrical system for the vehicle needs to be connected. Even though, this is possible, there is a great need for developing a rather specific interface for each system that is to be tested, generating non-existing powertrain signals, scaling signals, changing units etc. In this work, a light (without diagnostics) software version of the distributed control system is connected to a model of a hybrid vehicle. From the case study, it can be concluded that it is not straight forward to connect the controller to the simulation model, there is a need to design a rather large interface system in order to generate or create necessary signals.

The models used in the Japanese method are on a good and simple level. Unfortunately, some important dynamics are missing or not accurately modeled, such as engine and electric machine temperature and engine torque build-up. The mentioned missing dynamics are important to include in order not to overuse the hybrid system or powertrain and get better results than actually is possible. In this work, a set of models have been developed. The models have the same or similar characteristics as the ones used in the Japanese method, but the models include the above stated missing dynamics.

Finally, it can be concluded that Hardware-in-the-loop simulation for certification is not trivial. The Japanese method is one step on the way to a more efficient type-approval method for hybrid electric vehicles, but it needs some improvements.

References

1. Den Norske Veritas, *Standard for certification of hil testing*, Tech. report, REPORT NO. 2005-1301, 2005.
2. M. Duoba, *Challenges for the vehicle tester in characterizing hybrid electric vehicles*, In Proceedings in the 7th CRC On Road Vehicle Emissions Workshop, 1997.
3. M.m Duoba, J. Anderson, and H. Ng, *Issues in emissions testing of hybrid electric vehicles*, Global Powertrain Congress, 2000.
4. Hosam K. Fathy, Zoran S. Filipi, Jonathan Hagena, and Jeffrey L. Stein, *Review of hardware-in-the-loop simulation and its prospects in the automotive area*, vol. 6228, SPIE, 2006, p. 62280E.
5. Z. Filipi, H. Fathy, J. Hagena, Knaf. A., R. Ahlawat, L. Liu, D. Jung, D. Assanis, P. Peng, and J. Stein, *Engine-in-the-loop testing for evaluating hybrid propulsion concepts and transient emissions - hmwv case study*, SAE Paper No 2006-01-0443 (SAE International, ed.), 2006.
6. J. Fredriksson, J. Larsson, J. Sjberg, and P.Krus, *Evaluating hybrid electric and fuel cell vehicles using the capsim simulation environment*, EVS22, 2006.
7. R. Isermann, *Fault-diagnosis systems: An introduction from fault detection to fault tolerance*, Springer Verlag, 2006.
8. R. Isermann and P. Ballé, *Trends in the application of model-based fault detection and diagnosis of technical processes*, Control Engineering Practice **5** (1997), 707–719.
9. JASIC, *Test procedure for fuel consumption rate and exhaust emissions of heavy-duty hybrid electric vehicles using hardware-in-the-loop simulator system*, Tech. report, Kokujikan No. 281, 2007.
10. M. Kärverud, *Modeling of degradation in a li-ion traction battery for hybrid electrical vehicle applications*, Master’s thesis, Chalmers University of Technology, 2005.
11. D. Kim, C. Kim, S. Hwang, and H. Kim, *Hardware in the loop simulation of vehicle stability control using regenerative braking and electro hydraulic brake for hybrid electric vehicle*, In Proceedings of the 17th World Congress of the International Federation of Automatic Control, 2008.
12. J.R.P. Kusters, *Modelling and coordinated control of advanced automotive transmissions*, Master’s thesis, Technische Universiteit Eindhoven, 2005.
13. L. Laine and J. Fredriksson, *Brake blending for hybrid electric vehicles using control allocation*, Tech. Report R011_2007, Department of Signals and Systems, Chalmers University of Technology, 2007.
14. P. Lingman, *Integrated brake control - downhill driving strategies*, Ph.D. thesis, Department of Applied Mechanics, Chalmers University of Technology, 2005.
15. Kenji Morita, Kazuki Shimamura, Seiichi Yamaguchi, Keiji Furumachi, Nobuya Osaka, Shuichi Nakamura, Kazuyuki Nurusawa, Kwang-Jae Myong, and Terunao Kawai, *Development of a Fuel Economy and Exhaust Emissions Test Method with HILS for Heavy-Duty HEVs*, SAE International Journal of Engines **1** (2009), no. 1, 873–887.
16. T. Nakai, H. Uozumi, S. Hisai, S. Kobayashi, and T. Yamasaki, *Development of integrated hybrid vehicle simulator*, FUJITSU TEN Technical Journal **Vol 30** (2008).
17. R.F. Nelson, *Power requirements for batteries in hybrid electric vehicles*, Journal of Power Sources **Vol. 91** (2000), no. No. 1, 2–26.
18. Ministry of the Environment website, *Transformation algorithm into the japanese new transient engine test cycle - for spark ignition and for diesel engine*, <http://www.env.go.jp/air/car/program/en/index.html>, January 2010.
19. N. Okuda, N. Ishikawa, Z. Kang, T. Katayama, and T. Nakai, *Hils application for hybrid system development*, SAE Paper No 2007-01-3469 (SAE International, ed.), 2007.
20. C. Paredis, A. Diaz-Calderon, R. Sinha, and P. Khosla, *Composable models for simulation based design*, Engineering with Computers **17** (2001), 112–128.
21. M. Pasquier, M. Duoba, and A. Rousseau, *Validating simulations tools for vehicle system studies using advanced control and testing procedure*, In Proceedings of the 18th International Electric Vehicle Symposium (EVS18), 2001.
22. H. Paynter, *Analysis and design of engineering systems*, MIT Pres, 1961.

23. M.K. Salaani and G.J. Heydinger, *Powertrain and brake modeling of the 1994 ford taurus for the national advanced driving simulator*, SAE Paper No 981190 (SAE International, ed.), 1998.
24. M.K. Salaani, C. Schwarz, G.J. Heydinger, and P.A. Grygier, *Parameter determination and vehicle dynamics modeling for the national advanced driving simulator of the 2006 bmw 330i*, SAE Paper No 2007-01-0818 (SAE International, ed.), 2007.
25. A. Serrarens, *Coordinated control of the zero inertia powertrain*, Ph.D. thesis, Technische Universiteit Eindhoven, 2001.
26. K-I. Shimizu, M. Nihei, and T. Okamoto, *Guideline for measurement of quantity-of-electricity in fuel consumption test for hev*s, In Proceedings of the 22nd International Battery, Hybrid and Fuel Cell Electric Vehicle Symposium and Exposition, 2006.
27. R. Skjetne and O. Egeland, *Hardware-in-the-loop testing of marine control systems*, SIMS2005 - Scandinavian Conference on Simulation and Modeling, 2005.
28. Roger Skjetne and Olav Egeland, *Hardware-in-the-loop testing of marine control system*, Modeling, Identification and Control: A Norwegian Research Bulletin **27** (2006), no. 4, 239–258.
29. UNECE, *Development of a worldwide harmonised heavy-duty engine emission test cycle*, Tech. report, ECE-GRPE WHDC Working Group, 2001.
30. O. Wallmark, *Dynamic modelling of permanent-magnet synchronous machines, model documentation in capsim*, <http://www.capsim.se>, 2005.

Appendix A. Simulink models

This appendix shows the Simulink implementations of the models developed for HIL. All models except the driver use the model structure presented in Figure 22. The model structure is divided into two parts, the physical model and the local controller. Every model includes a local controller, which converts the control signals from the control system (if existing) into local control signals, the block also send the sensor signal values to the control system, i.e. it handles the communication with control system and physical model. The physical model block includes the implementation of the model equations.

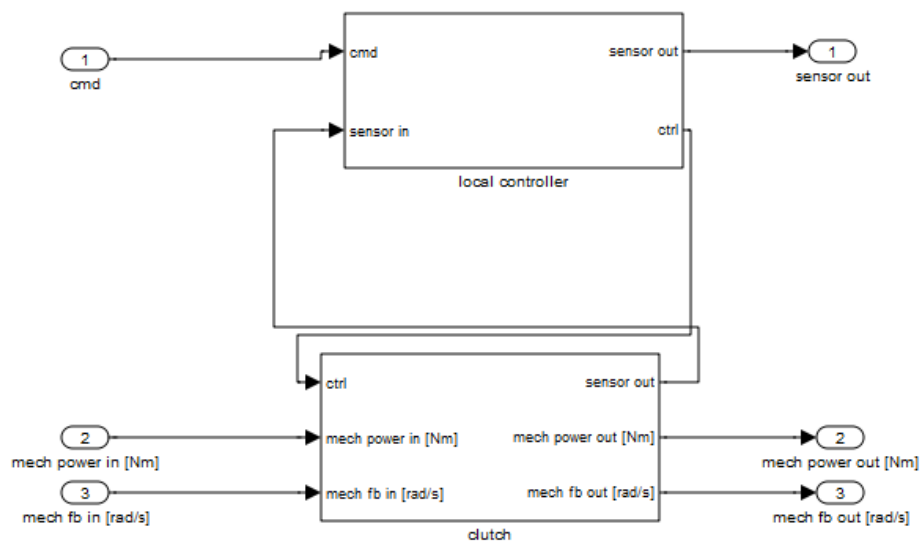


FIGURE 22. Model structure.

A.1. Chassis.

A basic model of the vehicle, where the vehicle is represented as an inertia. The model computes the vehicle speed given propeller shaft torque and brake torque. The model considers rolling- and air drag resistance and takes the road slope into account.

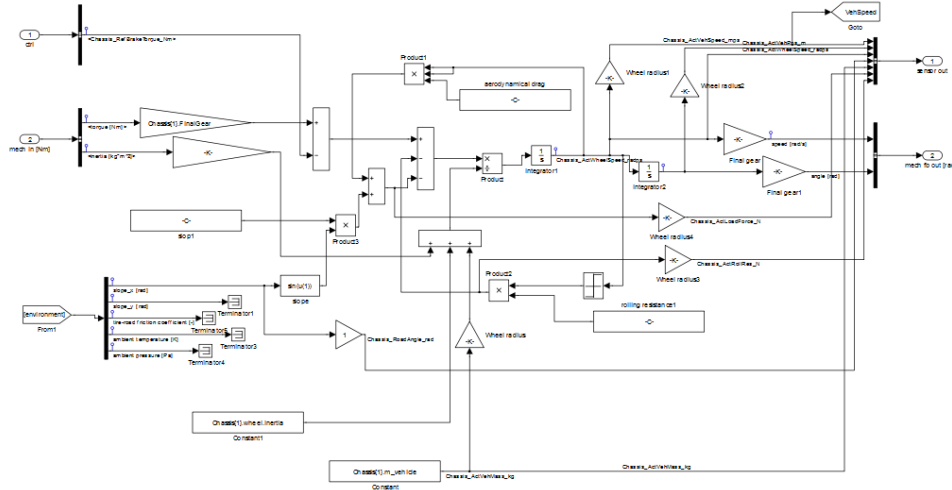


FIGURE 23. Chassis model.

A.1.1. Parameters and constants.

Description	Name in model	Unit
Final gear (differential gear)	Chassis{1}.FinalGear	-
Gravitational acceleration	Chassis{1}.g	m/s ²
Wheel radius	Chassis{1}.wheel.radius	m
Rolling resistance coefficient	Chassis{1}.wheel.rollres	-
Wheel inertia	Chassis{1}.wheel.inertia	kgm ²
Frontal area	Chassis{1}.A.front	m ²
Aerodynamic drag coefficient	Chassis{1}.Cd	-
Aerodynamic drag (Cd*A.front)	Chassis{1}.CdA	m ²
Air density	Chassis{1}.rho_air	kg/m ³
Vehicle mass	Chassis{1}.m_vehicle	kg

A.1.2. Control interfaces.

Node	Name	Description	Unit
cmd	Chassis_RefBrakeTorque_Nm	Reference brake torque	Nm
sensor_out	Chassis_ActVehSpeed_mps	Actual vehicle speed	m/s
	Chassis_ActVehPos_m	Actual vehicle position	m
	Chassis_ActWheelSpeed_rad/s	Actual wheel speed	rad/s

A.1.3. *Physical interfaces.*

Node	Name	Description	Unit
mech in	torque	Input torque	Nm
	inertia	Output inertia	kgm ²
mech fb out	speed	Feedback rotational speed	rad/s
	angle	Feedback rotational angle	rad

A.2. Clutch.

A simple model of a clutch. The working principle behind the clutch is that if the clutch is closed then the input torque is transferred to the output torque. If the clutch is open, the input shaft spins freely and no torque is transferred.

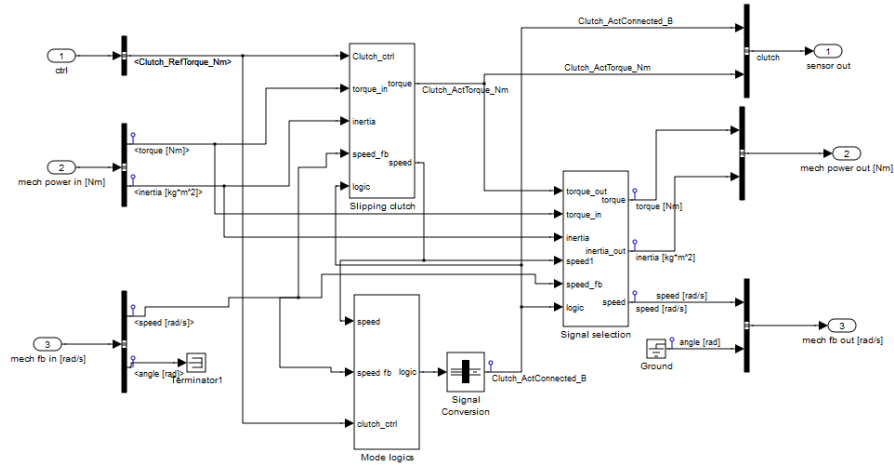


FIGURE 24. Clutch model.

A.2.1. Parameters and constants.

Description	Name in model	Unit
Maximum torque that can transferred	Clutch{1}.max_torque	Nm

A.2.2. Control interfaces.

Node	Name	Description	Unit
cmd	Clutch_RefTorque_Nm	Reference torque to be transmitted	Nm
sensor_out	Clutch_ActConnected_B	Is clutch connected or not	0/1

A.2.3. Physical interfaces.

Node	Name	Description	Unit
mech power in	torque	Input torque	Nm
	inertia	Output inertia	kgm ²
mech fb in	speed	Feedback rotational speed	rad/s
	angle	Feedback rotational angle	rad
mech power out	torque	Output torque	Nm
	inertia	Output inertia	kgm ²
mech fb out	speed	Feedback rotational speed	rad/s
	angle	Feedback rotational angle	rad

A.3. Driver.

The driver is modeled using a simple PI-controller.

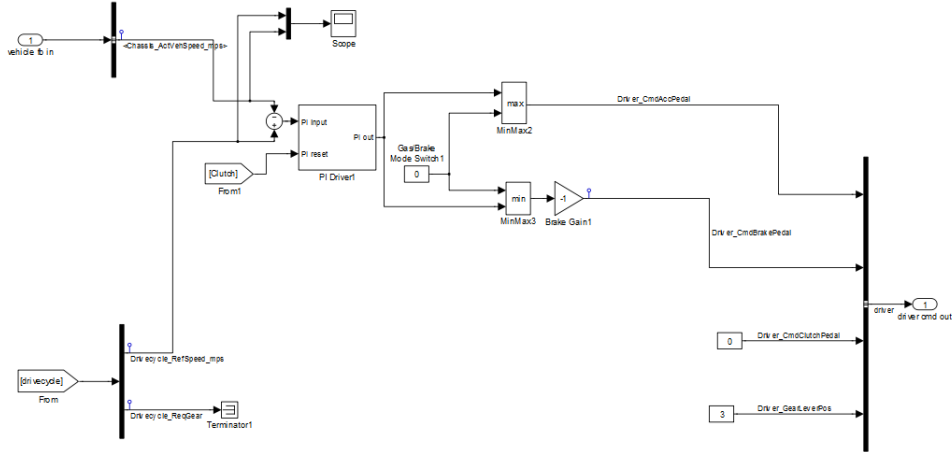


FIGURE 25. Driver model.

A.3.1. Parameters and constants.

Description	Name in model	Unit
Proportional gain for driver	Driver{1}.PID_P	
Integral gain for driver	Driver{1}.PID_I	

A.3.2. Interfaces.

Node	Name	Description	Unit
vehicle fb in	Chassis_ActVehicleSpeed_mps	Actual vehicle speed	m/s
driver cmd out	Driver_CmdAccPedal	Accelerator pedal command	[0-1]
	Driver_CmdBrakePedal	Brake pedal command	[0-1]
	Driver_CmdClutchPedal	Clutch pedal position	[0-1]
	Driver_CmdGearLeverPos	Gear lever position	[0-1]

A.4. Electric machine.

The electric machine is modeled using maps.

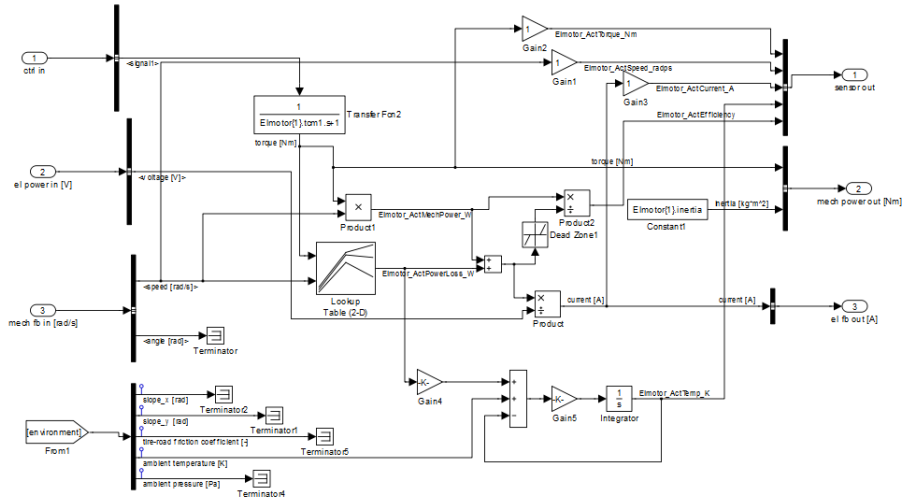


FIGURE 26. Electric machine model.

A.4.1. Parameters and constants.

Description	Name in model	Unit
Electric motor inertia	Elmotor{1}.inertia	kgm ²
Max torque (abs) curve (indexed by w_maxtq)	Elmotor{1}.max_torque	Nm
Index vector for max torque vector	Elmotor{1}.w_maxtq	rad/s
Power loss map (indexed by w_Ploss and T_Ploss)	Elmotor{1}.Ploss	W
Speed index for losses	Elmotor{1}.w_Ploss	rad/s
Torque index for losses	Elmotor{1}.T_Ploss	Nm
Mechanical time constant for electric machine	Elmotor{1}.tcm1	
Thermal resistance from rotor to stator	Elmotor{1}.Rth1	C/W
Thermal resistance from stator to ambient	Elmotor{1}.Rth2	C/W
Heat time constant for electric machine	Elmotor{1}.theat	
Proportional gain for speed controller	Elmotor{1}.PID_P	
Integral gain for speed controller	Elmotor{1}.PID_I	

A.4.2. Control interfaces.

Node	Name	Description	Unit
cmd	Elmotor_RefSpeed_radps	Reference speed	rad/s
	Elmotor_CmdRefSwitch_B	Switch between torque (0) or speed control (1)	0/1
	Elmotor_RefTorque_Nm	Reference torque	Nm
sensor_out	Elmotor_ActTorque_Nm	Actual electric motor torque	Nm
	Elmotor_ActSpeed_radps	Actual electric motor speed	rad/s
	Elmotor_ActCurrent_A	Actual electric motor current	A
	Elmotor_ActTemp_K	Actual electric motor temperature	K

A.4.3. Physical interfaces.

Node	Name	Description	Unit
el power in	voltage	Input voltage	V
mech fb in	speed angle	Feedback rotational speed	rad/s
		Feedback rotational angle	rad
mech power out	torque inertia	Output torque	Nm
		Output inertia	kgm ²
el fb out	current	Output current	A

A.5. Energy storage.

The energy storage or buffer is modeled as a simple battery using a resistor model.

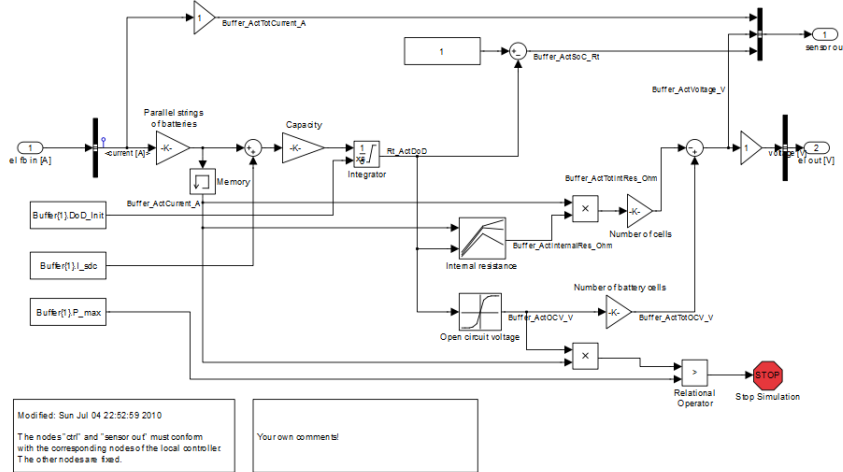


FIGURE 27. Energy storage model.

A.5.1. Parameters and constants.

Description	Name in model	Unit
Open circuit voltage (indexed by DoD_Uocv)	Buffer{1}.Uocv	V
Depth of Discharge index vector (DoD=1-SoC)	Buffer{1}.DoD_Uocv	[0-1]
Capacity of cell	Buffer{1}.Capacity	Ah
Self-discharge current	Buffer{1}.I_sdc	A
Coulombic efficiency	Buffer{1}.Eff_clmb	
Maximum allowed power	Buffer{1}.P_max	W
Number of cells in series	Buffer{1}.NoOfCells_series	
Number of parallel strings of cells	Buffer{1}.NoOfParString	
Initial DoD value	Buffer{1}.DoD_Init	%
Charging or discharging current (index vector for R.internal)	Buffer{1}.Current	A
Internal resistance map	Buffer{1}.R_internal	Ohm

A.5.2. Control interfaces.

Node	Name	Description	Unit
sensor_out	Buffer_ActTotCurrent_A	Total current	A
	Buffer_ActVoltage_V	Battery terminal voltage	V
	Buffer_ActSoC_Rt	State of charge	

A.5.3. Physical interfaces.

Node	Name	Description	Unit
el fb in	current	Input current	A
el power out	voltage	Output voltage	V

A.6. Gearbox.

The gearbox is modeled as two gears in contact, with a ratio of r_{gear} . The gearbox also includes a clutch model.

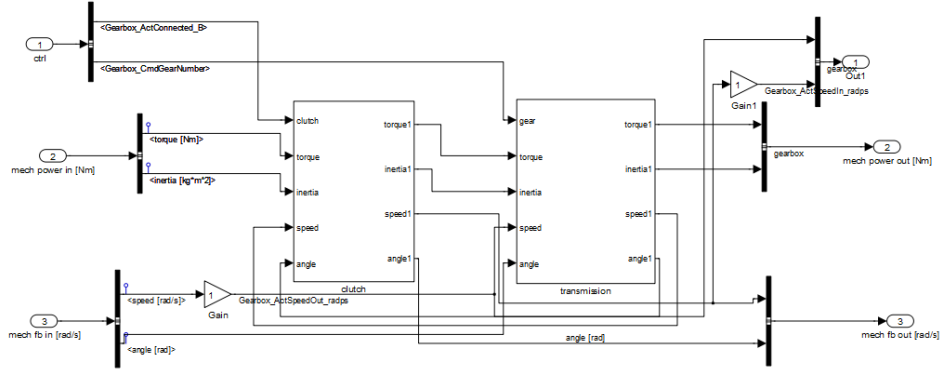


FIGURE 28. Gearbox model.

A.6.1. Parameters and constants.

Description	Name in model	Unit
Number of gears	Gearbox{1}.nog	
Gear numbers	Gearbox{1}.gear_number	
Gear ratios	Gearbox{1}.gear_ratio	
Inertia	Gearbox{1}.gear_inertia	kgm ²
Gear efficiency	Gearbox{1}.gear_efficiency	
Maximum torque transferred by clutch	Gearbox{1}.clutch.max_torque	Nm
If shifting controller is included:		
Shifting time	Gearbox{1}.clutch.time	s
Gearshifting up	Gearbox{1}.gear_up	rad/s
Gearshifting down	Gearbox{1}.gear_down	rad/s

A.6.2. Control interfaces.

Node	Name	Description	Unit
cmd	Gearbox_CmdGearNumber	Commanded gear number	
sensor_out	Gearbox_ActGearNumber	Actual gear number	
	Gearbox_ActConnected_B	Connected	0/1
	Gearbox_ActSpeedOut_radps	Outgoing speed from gearbox	rad/s
	Gearbox_ActSpeedIn_radps	Ingoing speed to gearbox	rad/s

A.6.3. Physical interfaces.

Node	Name	Description	Unit
mech power in	torque	Input torque	Nm
	inertia	Output inertia	kgm ²
mech fb in	speed	Feedback rotational speed	rad/s
	angle	Feedback rotational angle	rad
mech power out	torque	Output torque	Nm
	inertia	Output inertia	kgm ²
mech fb out	speed	Feedback rotational speed	rad/s
	angle	Feedback rotational angle	rad

A.7. Internal Combustion Engine.

The internal combustion engine is modeled in a similar way as the electric machine. Unfortunately, the torque dynamics cannot be represented by one linear first order system, because of the turbo dynamics.

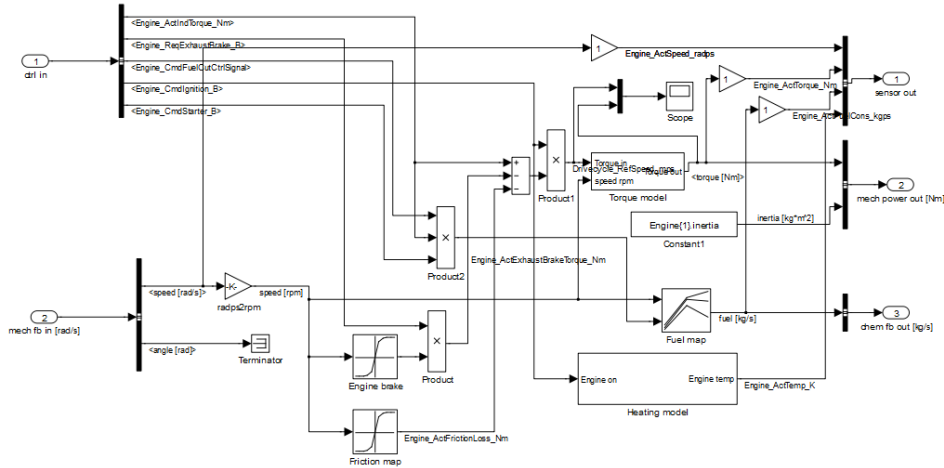


FIGURE 29. Internal combustion engine model.

A.7.1. Parameters and constants.

Description	Name in model	Unit
Engine inertia	Engine{1}.inertia	kgm ²
Max torque curve (indexed by w_maxtq)	Engine{1}.max_torque	Nm
Index vector for max torque vector	Engine{1}.w_maxtq	rad/s
Engine friction curve (index by w_loss)	Engine{1}.friction_loss	Nm
Index vector for friction curve	Engine{1}.w_loss	rad/s
Engine exhaust brake (indexed by w_exhaust)	Engine{1}.exhaust_brake	Nm
Index vector for exhaust brake curve	Engine{1}.w_exhaust	rad/s
Fuel consumption map (indexed by tq_fuel and w_fuel)	Engine1.fuelmap	kg/s
Torque index vector	Engine{1}.tq_fuel	Nm
Speed index vector	Engine{1}.w_fuel	rpm
Proportional gain for speed controller	Engine{1}.PID_P	
Integral gain for speed controller	Engine{1}.PID_I	

A.7.2. Control interfaces.

Node	Name	Description	Unit
cmd	Engine_RefSpeed_radps	Reference speed	rad/s
	Engine_CmdRefSwitch_B	Switch between torque or speed control	0/1
	Engine_RefTorque_Nm	Reference torque	Nm
	Engine_ReqExhaustBrake_B	Exhaust brake	0/1
	Engine_CmdIgnition_B	Engine Ignition	0/1
	Engine_CmdStarter_B	Starter	0/1
	Engine_CmdFuelCut_B	Fuel cut off	0/1
sensor_out	Engine_ActFuelCons_kgps	Engine fuel consumption (instantaneous)	kg/s
	Engine_ActSpeed_radps	Actual engine speed	rad/s
	Engine_ActTorque_Nm	Actual engine torque	Nm

A.7.3. Physical interfaces.

Node	Name	Description	Unit
mech fb in	speed	Feedback rotational speed	rad/s
	angle	Feedback rotational angle	rad
mech power out	torque	Output torque	Nm
	inertia	Output inertia	kgm ²
chem fb out	fuel	Fuel flow	kg/s

A.8. Mechanical connection.

This component is used to connect two input shafts. One shaft goes through and is connected to a joining input shaft through gears.

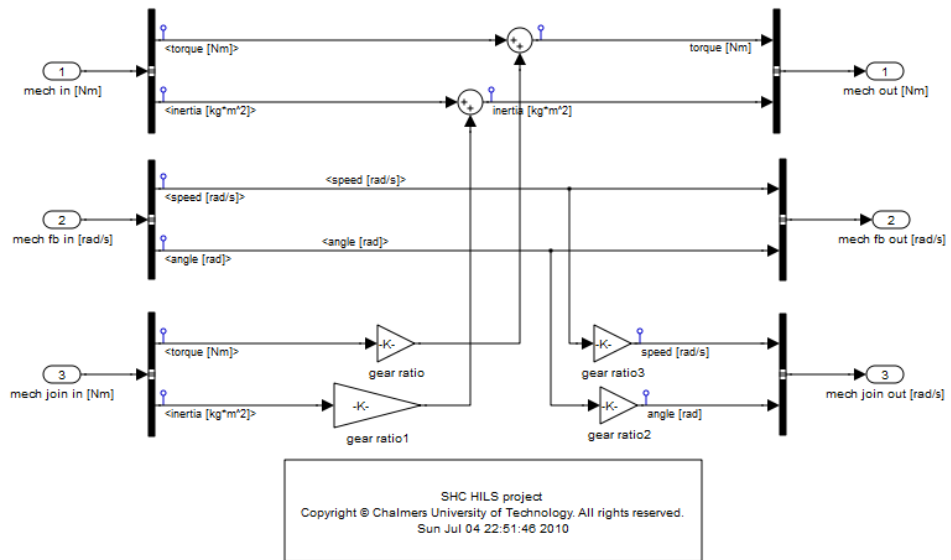


FIGURE 30. Mechanical connection model.

A.8.1. Parameters and constants.

Description	Name in model	Unit
Gear ratio	MechJoin{1}.ratio	

A.8.2. Control interfaces.

No control signals

A.8.3. Physical interfaces.

Node	Name	Description	Unit
mech in	torque inertia	Input torque 1 Output inertia	Nm kgm ²
mech fb in	speed angle	Feedback rotational speed Feedback rotational angle	rad/s rad
mech join in	torque inertia	Input torque 2 Output inertia	Nm kgm ²
mech out	torque inertia	Output torque Output inertia	Nm kgm ²
mech fb out	speed angle	Feedback rotational speed Feedback rotational angle	rad/s rad
mech join out	speed angle	Feedback rotational speed Feedback rotational angle	rad/s rad

Appendix B. Control system interface

Model	Input/Output	Symbol	Meaning of signal	Unit	Remarks
Buffer	Input-1	Buffer_ActTotCurrent_A	Total current	A	
	Input-2	Buffer_ActVoltage_V	Battery terminal voltage	V	
	Input-3	Buffer_ActSoC_Rt	State of charge	-	Value between 0 and 1
Chassis	Input-1	Chassis_ActVehSpeed_mps	Actual vehicle speed	m/s	
	Input-2	Chassis_ActVehPos_m	Actual vehicle position	m	
	Input-3	Chassis_ActWheelSpeed_rad/s	Actual wheel speed	rad/s	
Chassis	Output-1	Chassis_RefBrakeTorque_Nm	Reference brake torque	Nm	
	Input-1	Clutch_ActConnected_B	Is clutch connected or not	0/1	
Clutch	Output-1	Clutch_RefTorque_Nm	Reference torque to be transmitted through the clutch	Nm	
	Input-1	Elmotor_ActTorque_Nm	Actual electric motor torque	Nm	
Electric machine	Input-2	Elmotor_ActSpeed_radps	Actual electric motor speed	rad/s	
	Input-3	Elmotor_ActCurrent_A	Actual electric motor current	A	
	Input-4	Elmotor_ActTemp_K	Actual electric motor temperature	K	
	Output-1	Elmotor_RefSpeed_radps	Reference speed	rad/s	
	Output-2	Elmotor_CmdRefSwitch_B	Switch between torque or speed control	0/1	0 = torque, 1 = speed
	Output-3	Elmotor_RefTorque_Nm	Reference torque	Nm	
	Input-1	Engine_ActFuelCons_kgpps	Engine fuel consumption (instantaneous)	kg/s	
Engine	Input-2	Engine_ActSpeed_radps	Actual engine speed	rad/s	
	Input-3	Engine_ActTorque_Nm	Actual engine torque	Nm	
	Output-1	Engine_RefSpeed_radps	Reference speed	rad/s	
	Output-2	Engine_CmdRefSwitch_B	Switch between torque or speed control	0/1	0 = torque, 1 = speed
	Output-3	Engine_RefTorque_Nm	Reference torque	Nm	
	Output-4	Engine_ReqExhaustBrake_B	Exhaust brake	0/1	on or off
	Output-5	Engine_CmdIgnition_B	Engine ignition	0/1	on or off
Chassis	Output-6	Engine_CmdStarter_B	Starter	0/1	on or off, (engine on or off)
	Output-7	Engine_CmdFuelCut_B	Fuel cut off	0/1	on or off
Gearbox	Input-1	Gearbox_ActGearNumber	Actual gear number	-	
	Input-2	Gearbox_ActConnected_B	Connected?	0/1	Is gearbox in neutral or not
	Input-3	Gearbox_ActSpeedOut_radps	Outgoing speed from gearbox	rad/s	
	Input-4	Gearbox_ActSpeedIn_radps	Ingoing speed to gearbox	rad/s	
	Output-1	Gearbox_CmdGearNumber	Commanded gear number	-	

Appendix C. Powertrain Data

C.1. Conventional bus.

```

% HILS parameters for Conventional Bus
%-----
% Jonas Fredriksson
% Chalmers University of Technology
% 2010-06-30
%-----

%-----
% Chassis parameters
%-----
Chassis{1}.FinalGear    = 4.7;    % Final gear (differential gear) [-]
Chassis{1}.g           = 9.82;    % Gravitational acceleration [m/s^2]
Chassis{1}.wheel.radius = 0.47;  % Wheel radius [m]
Chassis{1}.wheel.rollres = 0.005; % Rolling resistance coefficient

Chassis{1}.A_front     = 8;      % Frontal area [m^2]
Chassis{1}.Cd          = 0.7;    % Aerodynamic drag coefficient
Chassis{1}.CdA         = 5.6;    % Aerodynamic drag (Cd*A_front)
Chassis{1}.rho_air     = 1.2;    % Air density [kg/m^3]
Chassis{1}.m_vehicle   = 14000;  % Vehicle mass [kg]

%-----
% Clutch parameters
%-----
Clutch{1}.max_torque   = 2000;   % Max torque transm [Nm]

%-----
% Driver
%-----
Driver{1}.PID_P        = 0.4;    % Proportional gain for driver
Driver{1}.PID_I        = 0.01;   % Integral gain for driver

%-----
% Internal combustion engine parameters
%-----
Engine{1}.inertia      = 2.1;    % Engine inertia [kgm^2]
% Controller
Engine{1}.PID_P        = 2;      % Proportional gain for controller
Engine{1}.PID_I        = 20;    % Integral gain for controller
% Torque build-up
Engine{1}.w_boost      = ...    % Index vec for "boost pressure" [rpm]
    [0 600 800 1000 1200 1400 1600 1800 2000 2200 2400 2600];
Engine{1}.t_boost      = ...    % Time constant for engine torque
    [3 3 3 5 6 7 3 3 2 1.5 1.5 1];
Engine{1}.w_instor     = ...    % Index vec for inst torque [rpm]
    [0 600 800 1000 1200 1400 1600 1800 2000 2200 2400 2600];
Engine{1}.ins_torque   = ...    % Instant torque [Nm]
    [0 590 570 530 460 370 330 320 300 270 260 100];
% Heating model
Engine{1}.theat        = 50;     % Heat time constant for engine
Engine{1}.warm_temp    = 273 + 95; % Engine temperature [K]
% Engine friction
Engine{1}.friction_loss = ...    % Engine friction curve [Nm]
    [0,70,75,80,85,95,105,130,140,155];

```

```

Engine{1}.w_loss          = ...      % Index vector for friction curve [rpm]
    [0,500,700,900,1100,1300,1500,1900,2100,2300];
% Exhaust brake
Engine{1}.exhaust_brake = ...      % Engine exhaust brake [Nm]
    [0 0 90 110 120 150 160 190 200 220];
Engine{1}.w_exhaust      = ...      % Index vector for exhaust brake [rpm]
    [0 600 900 1100 1200 1500 1600 1900 2000 2200];
% Maximum torque
Engine{1}.max_torque     = ...      % Max torque curve [Nm]
    [620,620,926, 1043,1161,1161,1161,1161,1161,1161,1161,1161,...
    1114,1067,1014,982, 941, 824, 314, 0, 0, 0, 0];
Engine{1}.w_maxtq       = ...      % Index vector for max torque [rpm]
    [600,800,1000,1100,1200,1300,1430,1500,1600,1700,1780,1800,...
    1900,2000,2130,2200,2300,2400,2600,2650,2700,2710,2720];
% Fuel consumption
Engine{1}.tq_fuel       = ...      % Torque index vector [Nm]
    [100 700 1100 1400];
Engine{1}.w_fuel        = ...      % Speed index vector [rpm]
    [0 600 900 1100 1200 1500 1600 1900 2000];
Engine{1}.fuelmap       = ...      % Fuel consumption map [kg/s]
    zeros(length(Engine{1}.w_fuel),length(Engine{1}.tq_fuel));

%-----
% Gearbox
%-----
Gearbox{1}.nog          = 6;        % Number of gears
Gearbox{1}.gear_number = ...      % Gear numbers
    [0 1 2 3 4 5 6];
Gearbox{1}.gear_ratio   = ...      % Gear ratios
    [0.0 3.4 2.6 2.1 1.5 1.2 1];
Gearbox{1}.gear_inertia = ...      % Inertia
    [0 0.05 0.05 0.05 0.05 0.05 0.05];
Gearbox{1}.gear_efficiency = ...   % Gear efficiency
    [0 0.97 0.97 0.97 0.97 0.97 0.98 ];
Gearbox{1}.clutch.max_torque = 2000;% Maximum torque
Gearbox{1}.clutch.time  = 0.5;    % Shifting time
% Gearshifting up at approx 2500 rpm
Gearbox{1}.gear_up      = ...
    (2000*2*pi/60)./(Gearbox{1}.gear_ratio(2:6).*Chassis{1}.FinalGear./Chassis{1}.wheel.radius);
% Gearshifting down at approx 1000 rpm
Gearbox{1}.gear_down    = ...
    1000*2*pi/60./Gearbox{1}.gear_ratio(2:6)./Chassis{1}.FinalGear*Chassis{1}.wheel.radius;

%-----
% Controller
%-----
ECU{1}.IdleSpeed       = 600*2*pi/60;
ECU{1}.MaxBrakeTorque  = 40000;
ECU{1}.maxTorque       = 1600;
ECU{1}.pedalmap_pedal  = [0 1];
ECU{1}.pedalmap_speed  = [0 300];
ECU{1}.pedalmap_torque_ref = [0 0;1550 1550];

```

C.2. Hybrid distribution truck.

```

% HILS parameters for Hybrid Distribution Truck
%-----
% Jonas Fredriksson
% Chalmers University of Technology
% 2010-06-30
%-----

%-----
% Chassis parameters
%-----
Chassis{1}.FinalGear    = 2.64;    % Final gear (differential gear) [-]
Chassis{1}.g           = 9.82;    % Gravitational acceleration [m/s^2]
Chassis{1}.wheel.radius = 0.522; % Wheel radius [m]
Chassis{1}.wheel.rollres = 0.0047; % Rolling resistance coefficient
Chassis{1}.wheel.inertia = 77    % Inertia of wheel [kgm^2]

Chassis{1}.A_front     = 9.9;     % Frontal area [m^2]
Chassis{1}.Cd          = 0.6;     % Aerodynamic drag coefficient
Chassis{1}.CdA         = 5.94;    % Aerodynamic drag (Cd*A_front)
Chassis{1}.rho_air     = 1.2;     % Air density [kg/m^3]
Chassis{1}.m_vehicle   = 25000;   % Vehicle mass [kg]

%-----
% Clutch parameters
%-----
Clutch{1}.max_torque   = 5000;    % Max torque transm [Nm]

%-----
% Energy storage parameters (Battery)
%-----
Buffer{1}.Uocv         = ...      % Open circuit voltage [V]
    [3.50 3.46 3.42 3.40 3.37 3.36 3.36 3.35 3.35 3.35 ...
    3.35 3.35 3.35 3.35 3.35 3.34 3.34 3.33 3.32 3.31 ...
    3.31 3.30 3.30 3.30 3.29 3.28 3.26 3.24 3.23 3.23 ...
    3.21 3.20 3.18 3.16 3.13 3.10 3.06 3.01 2.95 2.88 2.80];
Buffer{1}.DoD_Uocv    = ...      % DoD index vector (DoD=1-SoC) [0..1]
    [0.00 0.01 0.02 0.03 0.04 0.05 0.06 0.07 0.08 0.09 ...
    0.10 0.13 0.15 0.18 0.20 0.25 0.30 0.35 0.40 0.45 ...
    0.50 0.55 0.60 0.65 0.70 0.75 0.80 0.83 0.85 0.88 ...
    0.90 0.91 0.92 0.93 0.94 0.95 0.96 0.97 0.98 0.99 1.00];
Buffer{1}.Capacity    = 3.8;     % Capacity of cell [Ah]
Buffer{1}.I_sdc       = 0;       % Self-discharge current [A]
Buffer{1}.Eff_clmb    = 0.99;    % Coulombic efficiency [-]
Buffer{1}.P_max       = 860;     % Maximum allowed power [W]
Buffer{1}.NoOfCells_series = 192; % Number of cells in series
Buffer{1}.NoOfParStrings = 1;    % Number of parallel strings of cells
Buffer{1}.DoD_Init    = 0.5;     % Initial DoD value [%]
% Internal resistance
R_internal            = 0.0017; % Internal res (local variable) [ohm]
Buffer{1}.Current     = ...      % Current (index vector) [A]
    [-200 -150 -100 -50 0 50 100 150 200];
Buffer{1}.R_internal   = ...      % Internal resistance map [ohm]
    R_internal*ones(length(Buffer{1}.Current),length(Buffer{1}.DoD_Uocv));

```

```

%-----
% Internal combustion engine parameters
%-----
Engine{1}.inertia      = 2.1;      % Engine inertia [kgm^2]
% Controller
Engine{1}.PID_P        = 2;        % Proportional gain for controller
Engine{1}.PID_I        = 20;       % Integral gain for controller
% Torque build-up
Engine{1}.w_boost      = ...      % Index vec for "boost pressure" [rpm]
    [0 600 800 1000 1200 1400 1600 1800 2000 2200 2400 2600];
Engine{1}.t_boost      = ...      % Time constant for engine torque
    [3 3 3 5 6 7 3 3 2 1.5 1.5 1];
Engine{1}.w_instor     = ...      % Index vec for inst torque [rpm]
    [0 600 800 1000 1200 1400 1600 1800 2000 2200 2400 2600];
Engine{1}.ins_torque   = ...      % Instant torque [Nm]
    [0 590 570 530 460 370 330 320 300 270 260 100];
% Heating model
Engine{1}.theat        = 500;      %Heat time constant for engine
Engine{1}.warm_temp    = 273 + 95;% Engine temperature [K]
% Engine friction
Engine{1}.friction_loss = ...      % Engine friction curve [Nm]
    [0,70,75,80,85,95,105,130,140,155];
Engine{1}.w_loss       = ...      % Index vector for friction curve [rpm]
    [0,500,700,900,1100,1300,1500,1900,2100,2300];
% Exhaust brake
Engine{1}.exhaust_brake = ...      % Engine exhaust brake [Nm]
    [0 0 90 110 120 150 160 190 200 220];
Engine{1}.w_exhaust    = ...      % Index vector for exhaust brake [rpm]
    [0 600 900 1100 1200 1500 1600 1900 2000 2200];
% Maximum torque
Engine{1}.max_torque    = ...      % Max torque curve [Nm]
    [620,620,926, 1043,1161,1161,1161,1161,1161,1161,1161,1161,...
    1114,1067,1014,982, 941, 824, 314, 0, 0, 0, 0];
Engine{1}.w_maxtq      = ...      % Index vector for max torque [rpm]
    [600,800,1000,1100,1200,1300,1430,1500,1600,1700,1780,1800,...
    1900,2000,2130,2200,2300,2400,2600,2650,2700,2710,2720];
% Fuel consumption
Engine{1}.tq_fuel      = ...      % Torque index vector [Nm]
    [100 700 1100 1400];
Engine{1}.w_fuel       = ...      % Speed index vector [rpm]
    [0 600 900 1100 1200 1500 1600 1900 2000];
Engine{1}.fuelmap      = ...      % Fuel consumption map [kg/s]
    zeros(length(Engine{1}.w_fuel),length(Engine{1}.tq_fuel));

%-----
% Gearbox
%-----
Gearbox{1}.nog         = 12;       % Number of gears
Gearbox{1}.gear_number = ...      % Gear numbers
    [0 1 2 3 4 5 6 7 8 9 10 11 12];
Gearbox{1}.gear_ratio  = ...      % Gear ratios
    [0 15 12 9 7 5.5 4.4 3.4 2.6 2.1 1.5 1.2 1];
Gearbox{1}.gear_inertia = ...     % Inertia
    [0 0.05 0.05 0.05 0.05 0.05 0.05 0.05 0.05 0.05 0.05 0.05 0.05];
Gearbox{1}.gear_efficiency = ...  % Gear efficiency
    [0 0.97 0.97 0.97 0.97 0.97 0.98 0.98 0.98 0.98 0.98 0.98 0.99];
Gearbox{1}.clutch.max_torque = 5000;% Maximum torque

```

```

Gearbox{1}.clutch.time = 0.5;    % Shifting time

%-----
% Mechanical Join (Gear for adding torques)
%-----
MechJoin{1}.ratio      = 1;      % Gear ratio

%-----
% Electric motor
%-----
Elmotor{1}.inertia     = 0.5;    % Electric motor inertia [kg*m^2]
Elmotor{1}.max_torque  = ...     % Max torque (abs) curve [Nm]
    [523 523 523 523 523 523 523 523 523 523 474 426 388 355 328 ...
    305 284 266 251 237 224 213 203 194 185 178 171 164 158 ...
    152 147 142 138 133 129 125 122];
Elmotor{1}.w_maxtq     = ...     % Index vector for max torque vector
    [0 10 21 31 42 52 63 73 84 94 105 115 126 136 ...
    147 157 168 178 188 199 209 220 230 241 251 262 272 283 ...
    293 304 314 325 335 346 356 367];
Elmotor{1}.Ploss      = ...     % Power loss map [W]
    [0 8615 8895 9367 65973 82467 98960 115454 131947 148440 164934
181427 197920 214414 230907
    0 7698 7948 8215 62832 78540 94248 109956 125664 141372 157080
172788 188496 204204 219911
    0 6898 7121 7362 59690 74613 89535 104458 119381 134303 149226
164148 179071 193993 208916
    0 6184 6384 6600 56549 70686 84823 98960 113097 127235 141372
155509 169646 183783 197920
    0 5538 5717 5911 53407 66759 80111 93462 106814 120166 133518
146869 160221 173573 186925
    0 4946 5107 5281 5644 62832 75398 87965 100531 113097 125664
138230 150796 163363 175929
    0 4400 4543 4704 4889 58905 70686 82467 94248 106029 117810
129591 141372 153153 164934
    0 3898 4026 4169 4328 54978 65973 76969 87965 98960 109956
120951 131947 142942 153938
    0 3430 3543 3672 3819 4332 61261 71471 81681 91892 102102 112312
122522 132732 142942
    0 2998 3098 3216 3345 3578 56549 65973 75398 84823 94248 103673
113097 122522 131947
    0 2596 2684 2789 2909 3054 3759 60476 69115 77754 86394 95033
103673 112312 120951
    0 2224 2303 2397 2508 2633 2952 3678 62832 70686 78540 86394
94248 102102 109956
    0 1883 1950 2036 2136 2252 2436 2939 56549 63617 70686 77754
84823 91892 98960
    0 1569 1629 1705 1796 1905 2034 2334 2878 56549 62832 69115
75398 81681 87965
    0 1284 1335 1404 1487 1589 1705 1902 2241 2719 3315 60476 65973
71471 76969
    0 1027 1071 1132 1210 1304 1415 1560 1810 2138 2536 3054 3542
61261 65973
    0 796 835 891 964 1052 1157 1280 1478 1741 2047 2387 2764 3199
3722

```

```

0 593 628 679 747 832 933 1049 1212 1440 1704 1994 2300 2624
2972
0 417 448 496 561 643 740 855 998 1205 1451 1718 1999 2287
2596
0 269 297 343 406 484 579 692 827 1025 1265 1518 1794 2071
2375
0 145 171 216 277 355 451 559 697 904 1148 1410 1681 1959 2254
0 125 159 205 264 336 426 525 641 825 1021 1242 1448 1662 1890
0 232 277 329 393 471 565 673 802 989 1213 1441 1683 1940 2197
0 359 417 476 545 629 727 839 987 1191 1428 1682 1953 2234
2522
0 505 582 649 726 815 919 1037 1211 1443 1710 1999 2311 2644
3009
0 671 770 850 935 1032 1142 1275 1491 1769 2092 2462 2916 3573
54978
0 856 982 1076 1173 1278 1396 1566 1847 2214 2703 3432 56549
61261 65973
0 1058 1219 1329 1439 1555 1684 1930 2332 3058 54978 60476
65973 71471 76969
0 1279 1478 1609 1733 1862 2027 2413 3410 56549 62832 69115
75398 81681 87965
0 1517 1759 1915 2056 2199 2461 3266 56549 63617 70686 77754
84823 91892 98960
0 1771 2065 2247 2409 2569 3080 54978 62832 70686 78540 86394
94248 102102 109956
0 2043 2394 2608 2789 3014 4129 60476 69115 77754 86394 95033
103673 112312 120951
0 2333 2747 2995 3203 3637 56549 65973 75398 84823 94248 103673
113097 122522 131947
0 2639 3125 3410 3647 4772 61261 71471 81681 91892 102102 112312
122522 132732 142942
0 2961 3530 3860 4134 54978 65973 76969 87965 98960 109956
120951 131947 142942 153938
0 3305 3963 4340 4784 58905 70686 82467 94248 106029 117810
129591 141372 153153 164934
0 3671 4430 4863 6001 62832 75398 87965 100531 113097 125664
138230 150796 163363 175929
0 4062 4931 5426 53407 66759 80111 93462 106814 120166 133518
146869 160221 173573 186925
0 4479 5480 6047 56549 70686 84823 98960 113097 127235 141372
155509 169646 183783 197920
0 4934 6084 6734 59690 74613 89535 104458 119381 134303 149226
164148 179071 193993 208916
0 5435 6762 7764 62832 78540 94248 109956 125664 141372 157080
172788 188496 204204 219911];
Elmotor{1}.w_Ploss = ... % Speed index for losses [rad/s]
[0 26 52 79 105 131 157 183 209 236 262 288 314 340 367];
Elmotor{1}.T_Ploss = ... % Torque index for losses [Nm]
[-630 -600 -570 -540 -510 -480 -450 -420 -390 -360 -330 ...
-300 -270 -240 -210 -180 -150 -120 -90 -60 -30 30 60 90 120 ...
150 180 210 240 270 300 330 360 390 420 450 480 510 540 570 600];

Elmotor{1}.PID_P = 20; % Proportional gain for controller
Elmotor{1}.PID_I = 2; % Integral gain for controller

Elmotor{1}.tcm1 = 0.01; % Time constant for electric machine
%Heating of electric machine

```

```
Elmotor{1}.Rth1      = 0.01;    % Thermal resistance2 [C/W]
Elmotor{1}.Rth2      = 0.01;    % Thermal resistance1 [C/W]
Elmotor{1}.theat     = 0.01;    % Heat time constant
```