CHALMERS

ON VEHICLE SYSTEM CONTROL ARCHITECTURE FOR FUEL CELL- AND HYBRID ELECTRIC VEHICLES

LEO LAINE

Department of Applied Mechanics
CHALMERS UNIVERSITY OF TECHNOLOGY
Göteborg, Sweden, 2005

THESIS FOR THE DEGREE OF LICENTIATE OF ENGINEERING

in

Machine and Vehicle Systems

ON VEHICLE SYSTEM CONTROL ARCHITECTURE FOR FUEL CELL- AND HYBRID ELECTRIC VEHICLES

by

LEO LAINE

Department of Applied Mechanics
Division of Vehicle Safety
CHALMERS UNIVERSITY OF TECHNOLOGY
Göteborg, Sweden 2005

ON VEHICLE SYSTEM CONTROL ARCHITECTURE FOR FUEL CELL- AND HYBRID ELECTRIC VEHICLES
LEO LAINE

THESIS FOR LICENTIATE OF ENGINEERING no 2005:8

ISSN 1652-8565

Department of Applied Mechanics
Division of Vehicle Safety
Chalmers University of Technology
SE–412 96 Göteborg
Sweden
Telephone: +46 (0)31 772 1000

Tryck: Chalmers Reproservice, Göteborg

To Meggie

ON VEHICLE SYSTEM CONTROL ARCHITECTURE FOR FUEL CELL- AND HYBRID ELECTRIC VEHICLES

LEO LAINE
Department of Applied Mechanics
Division of Vehicle Safety
Chalmers University of Technology

Abstract

The work presented in this thesis concerns how vehicle system controllers can be made reusable for different Hybrid Electric Vehicle configurations. The vehicle system controller determines the driver's intentions in order to generate the desired vehicle motion above ground in the most energy efficient way.

Functional decomposition was used to divide the vehicle controller into different functional levels. Generic interface signals were applied between the levels so that the functions could be made compatible for multiple hybrid electric hardware configurations.

The suggested hierarchical control structure divides the hardware and software in different functions and levels, including the three main functional levels. The first level includes the following main functions: Driver Interpreter, which decides the desired longitudinal speed and path of motion, Vehicle Motion Control, which verifies that the path is within the dynamical limits of the vehicle, and Energy Management, which decides how the energy sources should be used in the most efficient way. The second level includes the following basic functions of a ground vehicle: Driver Interface, which reads the driver sensors and gives feedback to the driver, Chassis, which includes steering, tractive force, and braking, Power Supply, which includes the energy sources, converters and transformers that are located before differentials, Auxiliary, which includes all systems that are not necessary for generating vehicle motion, and finally External Information which contains functions for communication with other systems. The third level is the actuator and sensor level.

These ideas for the reusable vehicle system controller were applied and tested both by computer models and by implementation in a Scale Model Car of size 1:5. The suggested functional units were shown to be reusable for different HEV configurations. It was also demonstrated that the top level functions can be made hardware independent by using generic interface signals between higher and lower level functions.

Keywords: Fuel Cell, Hybrid Electric Vehicles, Control Architecture, Functional Units, Interfaces, Generic.

Acknowledgements

The author wishes to acknowledge the financial support from the Gröna Bilen National Research Programme within the FCHEV framework.

The author would like to acknowledge my supervisors Jonas Sjöberg and Jonas Fredriksson, who have provided good guidance during the work, Bengt Jacobson at Volvo Cars for his never-ending interest, and the rest of the steering group for the project.

Special thanks is directed to Johan Andreasson and Jonas Hellgren who have been highly involved in the work and made it really fun to try out ideas together.

Thanks also to Kanehira Maruo who always keeps us updated about what is actually going on in the industry when it comes to HEVs. Additionally the author would like to acknowledge all of my colleagues, especially Mattias Åsbogård and Lars Johannesson, for inspiring conversation about the research, Hans Sandholt for support with generating the first version of the Scale Model Car, Jan Möller, who always has a solution to practical problems, Jimmy Pettersson for his street-wise knowledge about computer issues, and Eirik Langjord for the discussions about how ground vehicles actually work.

Finally the author would like to thank family and friends for all their help and support, and especially to my dear Megan.

Table of Contents

List of Appended Papers

This thesis contains the following papers which will be referenced in text using their associated Roman numerals:

I. Leo Laine and Johan Andreasson, Generic Control Architecture applied to a Hybrid Electric Sports Utility Vehicle, in *Proceedings of the Electric Vehicle Symposium, (EVS 20)*, Los Angeles, USA, November 2003.

II. Johan Andreasson, Leo Laine, and Jonas Fredriksson, Evaluation of a Generic Vehicle Control Architecture, in *Proceedings of the* 30^{th} *Congress of the International Federation of Automotive Engineering Societies, (FISITA'30)*, Barcelona, Spain, May 2004.

III. Leo Laine and Johan Andreasson, Reusable Functional Partitioning of Tractive Force Actuators Applied on a Parallel Hybrid Electric Vehicle, in *Proceedings of the* 7^{th} *International Symposium on Advanced Vehicle Control, (AVEC'04)*, Arnhem, Netherlands, August 2004.

IV. Leo Laine, Jonas Hellgren, Henrik Kinnunen, and Magnus Rönnberg, Reusable Control Architecture Implemented in a Scale Model of a Hybrid Electric Vehicle, Technical Report, Division of Mechatronics, Chalmers University, March 2005. A shorter version is found in *Proceedings of the Electric Vehicle Symposium, (EVS 21)*, Monaco, April 2005.

The author has also contributed to the following papers

V. Johan Andreasson and Leo Laine, Driving Dynamics for Hybrid Electric Vehicles Considering Handling and Control Architecture, *Journal of Vehicle System Dynamics*, Vol. 41, pp 497-506, 2004.

VI. Leo Laine and Johan Andreasson, Modelling of Generic Hybrid Electric Vehicles, in *Proceedings of the* 3^{rd} *International Modelica Conference*, Linköping, Sweden, November 2003.

VII. Jonas Fredriksson, Johan Andreasson and Leo Laine, Wheel Force Distribution for Improved Handling in a Hybrid Electric Vehicle using Nonlinear Control, in *Proceedings of the* 43^{rd} *IEEE Conference on Decision and Control, (CDC'43)*, Paradise Island, Bahamas, December 2004.

VIII. Jonas Hellgren, Leo Laine, Jonas Sjöberg, Magnus Rönnberg, Dennys Gomes, and Aizezi Abudings, Systematic Design and Development of Hybrid Electric Scale Model Car, Submitted to *IEEE Transactions on Vehicular Technology*, October 2004.

Nomenclature

Actuator(A)	Device responsible for activating or putting into action.
Arbitration	Process of evaluating and prioritising request signals, where the number of incoming requests is greater than outgoing requests. The opposite of Coordination.
Architecture	Organisation of system hardware and software.
Auxiliary Systems(Aux)	Vehicle functionality not required for generating vehicle motion.
Buffer(bf)	Energy carrier which stores a limited amount of energy and can contribute both positive and negative power to the system.
Chassis(Ch)	Part of the vehicle responsible for the generation of ground motion including converters located after differentials and the Power Supply is excluded.
Connector(c)	Physical interface between Functional Units, such as mechanical and electrical.
Converter(Conv)	Hardware which converts energy into a different form, for example, a combustion engine converting chemical energy to mechanical or an electric motor converting electrical energy to mechanical.
Coordination	Process of splitting request signals by evaluation, where the number incoming requests is less than outgoing requests. The opposite of arbitration.

Driver Interface(DIf)	Device which receives driver input and provides sensor information in order to change certain sensor values and drive the vehicle.
Driver Interpreter(DIp)	Function that interprets the driver's intentions and sets a desired driving path.
Energy Carrier(EC)	Apparatus which carries energy in the vehicle. Examples of Primary ECs are the gasoline tank and hydrogen tank. Secondary ECs can be batteries or super capacitors.
Energy Management(EM)	Function that controls the power coordination between the available energy carriers within Power Supply.
Fuel Cell Vehicle(FCV)	Vehicle containing a converter where chemical hydrogen energy is converted to electrical which is then used mainly to propel the vehicle.
Function(Fn)	Action or activity that must be accomplished to achieve a desired outcome.
Functional Decomposition	Process of identifying fundamental functions within a system and decomposing the system into Functional Units.
Functional Unit(FU)	Entity of software and/or hardware capable of accomplishing a specific function.
Generic	Hardware independent.
Hybrid Electric Vehicle(HEV)	Vehicle containing two or more energy carriers used for propulsion, where at least one is electrical.
Information signal	Estimates of performed requests or request limits.
Interface	Shared boundary between two Functional Units, such as signals and/or connectors.
Limits(lim)	Upper and lower boundaries of request signals.
Power Supply(PS)	Part of the vehicle responsible for the main energy carriers and also converters located before differentials.

Request	Signal used for controlling a function.
Sensor(S)	Device that responds to a signal or stimulus.
State of Charge(SOC)	Level of energy within a buffer.
Strategic Control(SC)	Function that makes final arbitrations on request signals.
Vehicle Motion Control(VMC)	Function that controls the vehicle's ground motion and coordinates the Wheel Units.
Wheel Unit(WU)	Function located at the contact point between the chassis and the road surface that generates forces on the road.

Chapter 1

Introduction and Motivation

This chapter gives the background and motivation for this thesis. The objective, limitations, and main contributions are also stated.

1.1 Development of Control Functions within Vehicle Systems

Until recently, reusable software was typically of importance only for suppliers of subsystems for automotive manufacturers. For the suppliers it was a way in which to cut development costs. However, now automotive manufacturers also need to pay attention to how software can be reused. Not only would reusable software allow for different vehicle configurations to be produced without high development costs, it would also enable manufacturers to retain the brand specific characteristics of their vehicles as they become more and more dependent on used software functions within vehicle system controllers. The suppliers only have to focus on making sure delivered subsystems work correctly, whereas manufacturers have additional constraints, needing to integrate different subsystems into one correctly working vehicle system.

Future vehicle design will increasingly be focusing on the development and calibration of control functions. In the year 2000 this was estimated to be 4 percent of the total production costs of a car. It is estimated that in the year 2010 that figure will increase to 13 percent,[1]. Therefore, it is of increasing importance to be able to reuse control architectures for different hardware configurations. If the hardware and software is partitioned in a modular fashion, the work division between different developers of the control architecture's functions becomes clearly defined. This also enables the protection of the brand specific functionality that automotive manufacturers implement by software. Thus, creating specified interfaces is an effective way to divide software development between automotive manufacturers and suppliers, [2].

One example of how necessary it has become to investigate this partitioning is the initiation of the AUTomotive Open System ARchitecture (AUTOSAR) partnership in August 2002. Their purpose is to facilitate the effective integration of subsystems into functioning vehicle systems as smoothly as possible through examining how different functions can be identified within a vehicle system and how the interfaces between functional units can be defined. 'The objective of the partnership is the establishment of an open standard for automotive Electric/Electronic architecture. It will serve as a basic infrastructure for the management of functions within both future applications and standard software modules. The goals include the standardization of basic system functions and functional interfaces, the ability to integrate and transfer functions and to substantially improve software updates and upgrades over the vehicle lifetime. The AUTOSAR scope includes all vehicle domains', [3]. AUTOSAR is a good indicator of just how necessary it is to investigate functions and interfaces within vehicle systems.

1.2 Transportation and Energy Resources

There are several important issues for which alternative powertrains, such as Hybrid Electric Vehicles (HEV) and Fuel Cell Vehicles (FCV), provide interesting and viable technological solutions. To begin with, the world's oil resources are not never ending. At the current level of consumption there are sufficient oil reserves to meet market demand for only approximately 40 years [4]. During this time period the price of oil will increase, opening up a market space for the use of alternative energy resources. Secondly, dependency on oil affects political stability. 63 percent of the known oil reserves are located in the Middle East [4], however, Middle Eastern countries made up only 5.9 percent of the world's total oil consumption in the year 2003, [4]. In the future, the rest of the world will become highly dependent on oil production from the Middle East, inevitably causing asymmetrical political relations. Finally, global warming and green house effects are subsequent consequences when fossil fuels are used for energy. In 1990, the transportation sector was not only accountable for 25 percent of the world's energy use but also responsible for 22 percent of the global CO_2 emissions according to [5]. Transportation based on diesel and gasoline also degrades the local air quality around urban areas. Alternative powertrains such as HEVs will provide good solutions during the time when oil prices are increasing and no other genuinely sustainable replacements for fossil fuels are available.

1.3 Motivation

The development of HEV and FCV technology, in response to energy resource needs, and the increasing industry-motivated demand for reusable control systems, provide the impetus for this research and thesis.

One of the most relevant issues with HEV and FCV technology is that there are several actuators which can perform the same function, such as accelerating and braking. These actuators need to be coordinated within a control system. In conventional cars and trucks there is a starter and a generator, whereas in HEV vehicles these electric motors are sized up and even combined. The starter is also used to assist during acceleration and the generator can be used during braking to regenerate the braking energy into electric energy. The conventional battery is sized up to be able handle larger energy storage. Then the actuators, electric motor and combustion engine are coordinated to perform the function acceleration. When the function braking is performed the electric motor and mechanical brakes have to be coordinated. This is called a *parallel* HEV, due to the fact that the combustion engine still has a driveline that allows direct traction to the wheels in combination with the electric motor. However, even this simplest type of HEV now has at least two ways to apply traction during cruising or acceleration and two ways to brake during deceleration.

There are three main types of powertrain configurations within HEV technology. The first is *parallel*, as mentioned earlier. The second, *series*, in contrasts to *parallel* in that the converter, such as a combustion engine or a Fuel Cell does not have a mechanical connection directly to the wheels. The third type, *split*, is where a planetary gear is used to combine the combustion engine, electric motor and generator. Every configuration has different possibilities for how the energy flow within the powertrain can be handled and coordinated, see [6] for more information.

As mentioned in section 1.2, research on HEVs is mainly driven by environmental reasons. But more electrified powertrains also introduce new challenges such as X-by-wire. X-by-wire is a concept where mechanically and hydraulically controlled systems, such as rack-steering and braking, are replaced by electromechanical systems. A future scenario could be the use of Autonomous Corner Modules (ACM) [7]. The ACM concept allows every wheel to independently control driving and braking torque, and also control the normal force, and the wheel's steer, camber, and caster angles. This will be a large leap from how today's cars are designed, developed, produced, and in particular controlled.

1.4 Objective

The objective of the thesis is to identify how a generic and reusable control architecture can be constructed for HEVs and FCVs. Here the focus is on how the driver's intentions finally generate the motion of the vehicle with different hardware configurations.

1.5 Limitations

This thesis does not consider how the computational architecture should be constructed for the control architecture, in other words, how many computational nodes should be used. Nor does it address how the system could be realised and implemented in a fail-safe manner.

1.6 Main Contributions

The main contributions of the thesis are:

- Suggestion of Functional Units within a generic control architecture for HEVs and FCVs, and conventional vehicles as special case.
- Suggestion on how the partitioning of tractive force actuators between Chassis and Power Supply can be made.
- Suggestions on how generic interfaces between FUs should be constructed and used to allow for a wide range of hardware configurations.
- Verification of Functional Units and generic interfaces by virtual prototypes and the use of a scale model car.

1.7 Work Split between Authors

1.7.1 Vehicle Dynamical Aspects of Hybrid Electric Vehicles

HEV technology will particularly influence the dynamical aspects of vehicles by introducing electric motors as tractive force actuators. This in turn changes and enhances both overall performance and vehicle stability. In partnership with Johan Andreasson (KTH), Project HEV Driving Dynamics, a joint venture was begun with the purpose of identifying how a generic control architectures[1] can be

[1]The working name of the proposed architecture is JALLa- Johan Andreasson and Leo Laine architecture.

used within as Vehicle System Controllers for future HEV technology. One of the major tools for evaluating the suggested architecture was the construction of virtual prototypes. The focus for the author was the structuring of the complete system with its main functions and the derivation of generic interface signals.

1.7.2 Energy Management Aspects of Hybrid Electric Vehicles

The development of HEV technology is primarily motivated by the fact that the fuel consumption can be reduced compared to that of conventional vehicles. Although this thesis does not focus on the Energy Management of HEVs, a joint venture was started and realized together with Jonas Hellgren (Chalmers), project HEV Energy Management, with the purpose of building a Scale Model Car (SMC). The conceptual design of the powertrain and the energy management algorithms was performed by Jonas Hellgren, [8]. The functional partitioning and interfaces between Functional Units were implemented in the SMC's controller by the author. The actual building of the car was performed by master's theses students, [9] and [10].

1.8 Outline of the Thesis

The outline of this thesis is as follows: Chapter 2 provides an overview on vehicle system control architectures. Chapter 3 illustrates the suggested functional levels, Functional Units, and generic interface signals within the proposed generic control architecture. In Chapter 4 discusses the constructed prototypes. Finally, the concluding remarks are given in Chapter 5.

Chapter 2

On Vehicle System Control Architecture

This chapter provides an introduction to computerised controllers and an overview of how partitioning is made for vehicle system control architectures.

2.1 Computerised Controllers

The combustion engine was the first automotive actuator to receive a computerised controller, generally called the Electronic Control Unit (ECU). By the early 1980s they were introduced into vehicles on a large scale. Soon after a number of other application were installed with ECUs as controllers, eventually leading to a need for intercommunication between the ECUs. At the time, the amount of required wiring prevented the signals from being wired individually as separate cables. This problem was solved by a method called multiplexing which allows several channels to be carried within one cable. In the year 1983, Robert Bosch GMBH began an internal project to develop an invehicle network. The result of this project was the Controller Area Network (CAN) which was officially introduced the year 1986. By 1992 the CAN network protocol was used in production cars [11] and is now the dominating standard for connecting ECUs.

The physical layers within CAN have two different speeds, low[1] and high [2]. The standard CAN protocol is event driven with prioritised signals sent on the network as messages. All messages with high priority are sent each cycle, whereas messages with low priority are cancelled if needed. This leads to a stochastic transmission of low priority messages. The next generation of computational networks are time-triggered protocols [12], such as the TTCAN [11], TTPC [13], and

[1] up to 125 kbit/s

[2] up to 1 Mbit/s

Flexray [14], created in order to accommodate future safety critical applications such as x-by-wire.

2.2 Partitioning

There are two different ways of approaching the structuring of control architectures; see e.g. [15] and [16]. The first, computational partitioning, concerns how the vehicle system control software is computed. Although it is not considered in this thesis, a brief overview is however provided. The second is functional partitioning, which shows how the software itself is partitioned. Explanations are also given as to the different types of functional partitioning and their various advantages.

2.2.1 Computational Partitioning

Computational partitioning considers how computing resources should be distributed across different computer nodes. One type is *centralised* partitioning, which concentrates all of the sensors and actuators on to one node. Another is *distributed* partitioning in which the sensors and actuators are attached to several nodes, and in turn are connected by a communication bus. *Distributed* partitioning can also be *topographically distributed*, in which the distribution is placed near the subsystem under control, or additionally it can be *functionally distributed*, in which the distribution is decided not by location but instead by functional responsibility.

2.2.2 Functional Partitioning

Whereas computational partitioning focuses initially on the placement and interconnection of nodes, functional partitioning has a completely different approach, concentrating primarily on how functions are prioritised and executed within the computational nodes. There are mainly three different types of functional partitioning, *centralised*, *peer-to-peer* and *hierarchical*.

Centralised

In *centralised* functional partitioning, one top level function is used to control the whole system, see Figure 2.1. This central function contains all sensor information and can directly send requests to the specific actuators.

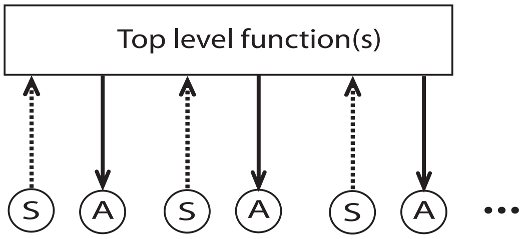


Figure 2.1: Centralised functional partitioning. Dashed and solid lines illustrate information and requests respectively. Hardware is illustrated by A=Actuator and S=Sensor.

The advantage to centralised functional partitioning is that information from all sensors are simultaneously received. The main draw back is that the whole function is affected if the hardware configuration is changed.

Peer-to-peer

In *peer-to-peer* functional partitioning, no top level function is used to control the whole system, instead only local functions are used, see Figure 2.2. The coordination is achieved by sending states as information between the local functions. Every local function attempts to sub-optimise its own function.

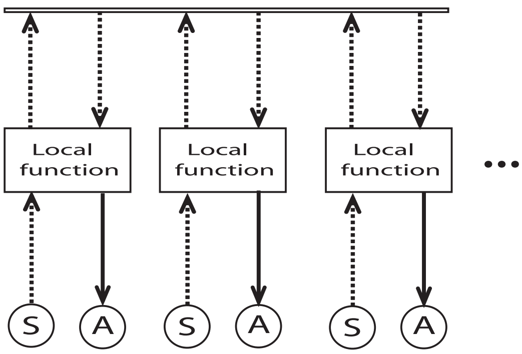


Figure 2.2: Peer-to-peer functional partitioning. Dashed and solid lines illustrate information and requests respectively. Hardware is illustrated by A=Actuator and S=Sensor.

Peer-to-peer functional partitioning is the most modular when compared to *centralised* and *hierarchical*. The drawback with *peer-to-peer* functional partitioning is that conflicts between the local functions are hard to avoid.

Hierarchical

Hierarchical partitioning contains top level and local functions, giving both better modularity than *centralised*, and better coordination between local functions than *peer-to-peer*. *Hierarchical* functional partitioning provides the ability to easily add, delete, and modify hardware [15]. It reduces the complexity of the system by having requests coming from the top level functions down to local functions, in this way creating a causal flow of requests. One drawback with the *hierarchical* is that enough information must be sent to top level functions to allow decisions on coordination to be performed.

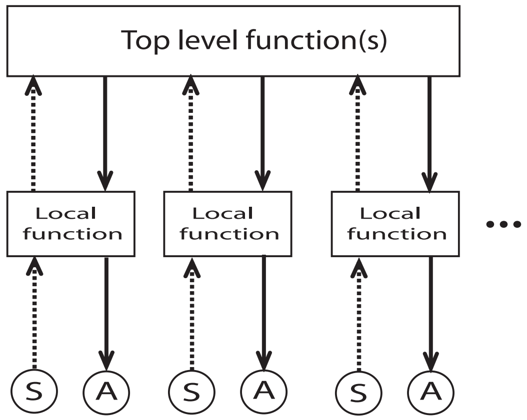


Figure 2.3: Hierarchical functional partitioning. Dashed and solid lines illustrate information and requests respectively. Hardware is illustrated by A=Actuator and S=Sensor.

If the requests and information signals are made reusable for different hardware configurations only small changes would be needed in the top level functions.

Chapter 3

Generic Vehicle System Control Architecture

This chapter starts out with definitions of frequently used words and explains the methods and principles used to derive the Generic Vehicle System Control Architecture. It then provides the specifics of the suggested architecture.

3.1 Definitions

The definitions provided here are some of the most frequently used words within this thesis, starting with function and Functional Unit.

Definition 3.1.1 *Function*
Action or activity that must be accomplished to achieve a desired outcome.

Definition 3.1.2 *Functional Unit*
Entity of software and/or hardware capable of accomplishing a specific function.

A vehicle system can be seen as a set of Functional Units. Communication between the Functional Units are needed to be able to divide the system. These boundaries of communication are defined as interfaces, see Definition 3.1.3.

Definition 3.1.3 *Interface*
Shared boundary between two Functional Units, such as request signals, information signals and/or physical connectors.

Due to the fact the hardware is also divided into different Functional Units physical connectors also become Interfaces.

3.2 Methods and Principles

A control architecture using hierarchical functional partitioning was chosen as the basis for this research and thesis. *Hierarchical* functional partitioning was utilised for the VSC architecture based on four main reasons. To begin with, this partitioning provides better modularity compared to that of *centralised* functional partitioning. Secondly, it allows for easier coordination of the local functions than that of *peer-to-peer*. Thirdly, *hierarchical* functional partitioning protects the automotive manufacturers' top level brand specific functions. Finally, it accommodates the establishment of generic interfaces between top level functions and local level functions[1].

The Vehicle System Control (VSC) architecture is designed to be generic so that it would have not only the capability to easily handle today's vehicle configurations but also the ability to already handle any vehicle configurations that can be foreseen a decade from now. In order for this to happen, specific hardware with local controllers need to be easily exchangeable with minimal effect on the VSC. An example of how a partitioning is made by functional decomposition in a VSC for a parallel HEV is shown in [17], however the example lacks the possibility to be generic because it focuses on specific actuators instead of their functions. Thoughts on functional decomposition are given in [18]. Definition 3.2.1 on functional decomposition is used as a guideline for partitioning the functions and Definition 3.2.2 is used as guideline when deriving the generic architecture.

Definition 3.2.1 *Functional decomposition*
The following statements characterise an architecture with functional decomposition:

I. *Functions are placed into different levels according to their coordinating authority over other functions.*

II. *Information on the system status can be observed by all functions and is allowed to flow in all directions, up, down, and across in the hierarchy.*

III. *Requests are only allowed to flow down to lower level functions. This upholds a causality of orders within the hierarchical architecture.*

IV. *Brand characteristics should only be contained within the top level functions.*

V. *Low level functions should have control over hardware health and durability.*

[1]This allows automotive suppliers and manufacturers to divide their software development.

Definition 3.2.2 *Generic control architecture*

I. *The control architecture should be hierarchical by functional decomposition.*

II. *Interfaces between top level and lower level functions should be made hardware independent.*

III. *The control architecture should be designed so as to accommodate any foreseeable future hardware developments for the system under consideration.*

Definitions 3.2.1 and 3.2.2 provide the essential characteristics for the structuring of reusable control architectures when applied to vehicle systems. For example, Definition 3.2.1, Item IV, allows manufacturers to retain ownership of brand specific functions while suppliers can provide controls for various subsystem functions. Through this, vehicle manufacturers can change vehicle characteristics such as optimizing drivability and fuel economy. Additionally, Definition 3.2.1, Item V, makes the vehicle supplier responsible for the durability of their hardware. Definition 3.2.2, Item I, declares that hierarchical functional partitioning should be used to allow better coordination between functions than peer-to-peer since top level functions for coordination exists. Additionally, Definition 3.2.2, Item II, allows hardware to be exchanged without redesigning the functional architecture. Finally, 3.2.2, Item III, states that one should try to predict the hardware and software development for the near future in order for the generic architecture to reach its full potential.

The procedure for deriving the architecture is illustrated in Figure 3.1, starting by identifying how many functional levels are needed, then by determining the Functional Units, and finally by defining the interfaces between those Functional Units. The last step is a reusability check to verify that the architecture handles different hardware configurations. As shown in Figure 3.1, the iterative process has three main back loops. The first back loop (1) is for ascertaining the number of required functional levels. The second back loop (2) is for determining if the Functional Units are complete enough to describe the system. The third back loop (3) is to confirm if the interfaces are generic enough. This final iteration is the most extensive in order to find, confirm and establish interfaces that work for several hardware configurations. This was mainly tested by prototype models, see [19] and Paper IV [20].

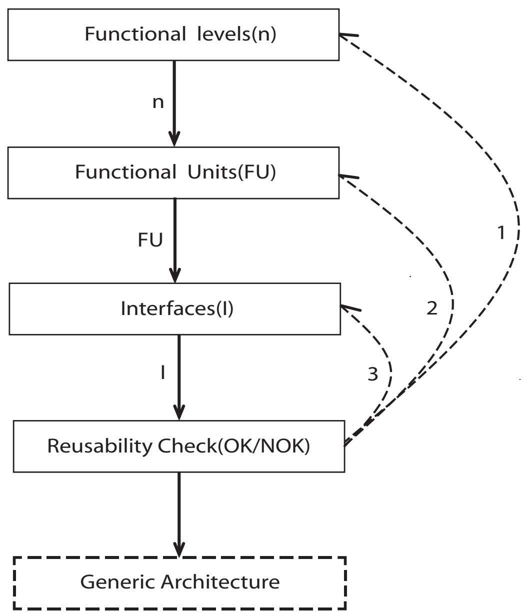


Figure 3.1: Illustration of the iterative process used for deriving the generic architecture, in which n=functional Levels, FUs=Functional Units, and I=Interfaces.

3.3 Functional Levels

The purpose of utilizing several functional levels is to distinguish a top level of Functional Units which should be seen as the coordination functions of the lower level functions. Three functional levels were identified. The top level is the Main Control (1). The second level contains the vehicle's functional tasks (2). The third level is the actuator and sensor level (3), see Figure 3.2. Three levels were also identified in [21] for a hierarchical control structure of HEVs.

By using generic interfaces between the FU's on different levels the top level can be made as hardware independent as possible.

3.4 Functional Units

Both the software and hardware of the vehicle system are decomposed, according to Definition 3.2.1, into Functional Units (FU), see Definition 3.1.2.

The Proposed FUs are described in Section 3.4.1 and Section 3.4.2 for functional level 2 and level 1 respectively. The proposed FUs within the VSC are shown in Figure 3.3.

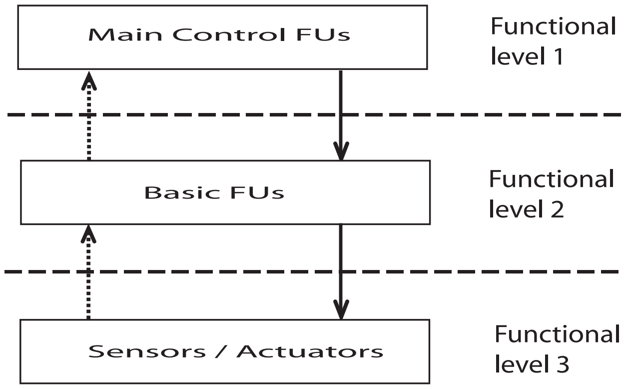


Figure 3.2: The three suggested functional levels used within the generic architecture.

3.4.1 Proposed Functional Units for level 2

Five basic FUs, common for every ground vehicle, were identified for functional level 2:

- Chassis (Ch) -generates ground motion.
- Driver Interface (DIf) -interacts with the driver.
- Power Supply (PS) -supplies power for ground motion and auxiliary systems.
- Auxiliary (Aux) -includes systems not needed for generating ground motion.
- External Information (EI) -receives and shares information with systems outside the vehicle.

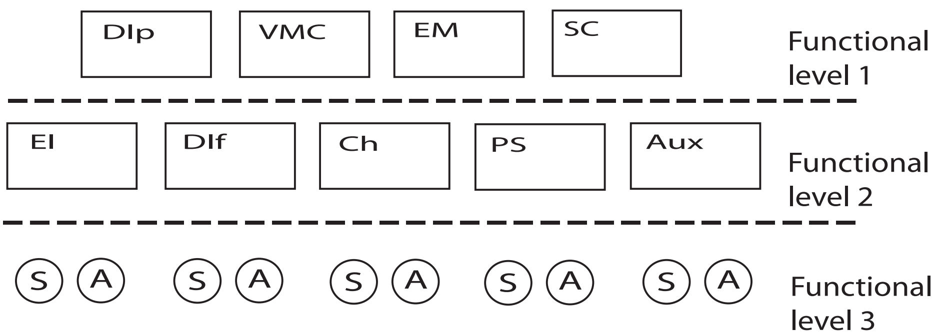


Figure 3.3: The suggested Functional Units within the generic VSC.

The first three FUs are seen as the least common factors for every ground vehicle. The last two are seen as optional FUs. Further detail on these FUs will be provided in Sections 3.4.1 and 3.4.2. These functions are also discussed in Paper I, [22].

Driver Interface (DIf)

The driver communicates with the vehicle through Driver Interface(DIf). The DIf function reads information from sensors such as the steering wheel, gas, and brake pedals or a joystick. It then gives force feedback to driver.

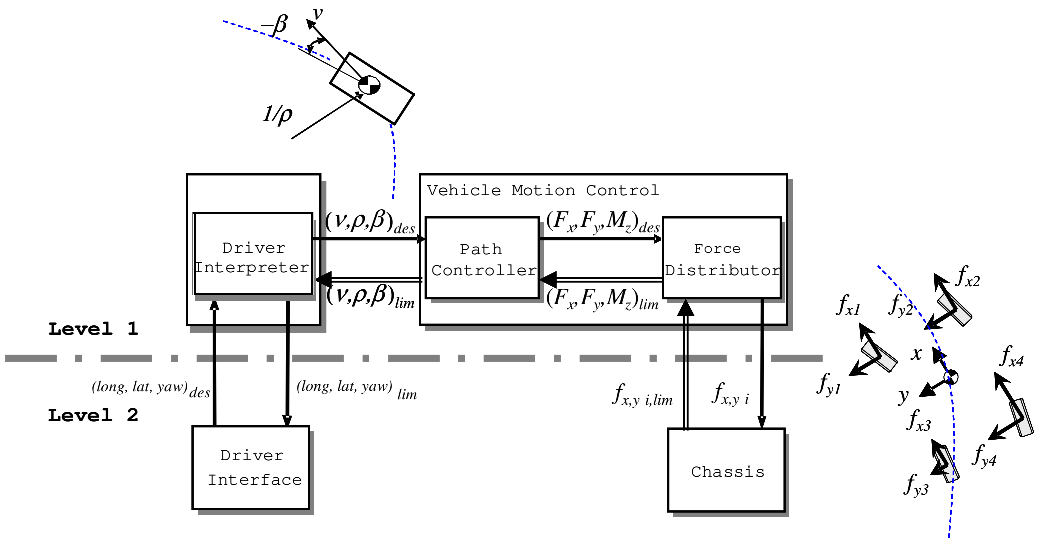


Figure 3.4: DIf function in relation to the other functions within the proposed FUs.

To define ground motion on a surface plane three degrees of freedom are required: *longitudinal*, *lateral*, and *yaw motion*. Figure 3.4 shows a sketch of how DIf is related to the other FUs within the proposed FUs. With a steering wheel the yaw motion and lateral acceleration are an indirect combination of the actual longitudinal speed and the applied steering angle. However, the driver could completely decide this with a three-axis joystick. Then forward backward would be the longitudinal motion, left-right would be the lateral motion, and turn would be the yaw motion.

Chassis (Ch)

Chassis (Ch) is the basic function that generates the vehicle's motion. Traditionally, Chassis has been controlled separately from safety systems like Anti-lock

Braking System, Traction Control System, and Vehicle Stability Control. In [23] these systems are combined into one supervisory controller Vehicle Dynamics Management (VDM). The VDM uses a hierarchical functional structure to go from the driver's input, calculate the desired global force and moments of the vehicle, and then finally distribute the forces between the available wheels. The VDM presented in [23] handles only the traction and braking. It also assumes that the tyre forces are limited only by the tyre's characteristics[2].

In the proposed Ch function, steering and actuator limits are included and considered to be applied independently. The Chassis function has been decomposed into building blocks in order to allow for any future vehicle configurations, especially including the possibility of having every wheel controlled independently when steering, traction, braking and normal forces are considered. This way the *Wheel Unit (WU)* building block is by default decoupled from the other WUs. These WUs apply the horizontal and vertical forces to the ground, see also Figure 3.4.

There are however, several cases where the WUs cannot be considered as decoupled. For example, in a conventional vehicle with rack steering on the front wheels, the steering angle is coupled between the WUs. Restrictions between WUs are included by the building block *Restrictors (Rs)*. The Rs between WUs are called *inter-Rs*. The limits of the connected actuators to a WU are called *intra-Rs*. The use of Rs is further demonstrated in Paper II, [24].

To handle several combinations of WUs mounted on different frames which can move independently, a third building block is introduced, *Bodies (Bd)*, which allows different combinations of ground vehicles be defined, for example an articulated bus. This is further explained in [25].

To summarise, by using the building blocks WU, Rs, and Bd several types of chassis configurations can be defined, and therefore handled by the same supervisory function at the top level, see also Paper II.

Power Supply (PS)

Power Supply (PS) is the FU that provides power for the generation of ground motion and Auxiliary systems. Conventional vehicles usually have a combustion engine that provides the power. However, in HEVs and FCVs the storage and production of onboard electricity is substantial. Therefore a more suitable name for this function is PS.PS has two main physical interfaces with the Ch. One is a mechanical connector that allows rotational power ($P_{mech} = \tau \cdot \omega$) to be transferred between PS and Ch. The tractive force actuators and converters are either placed within Ch or PS depending on their topology. If a tractive force

[2]All wheels can have unlimited braking and traction forces.

actuator or converter is placed before a differential then it belongs within PS. If the actuators are directly mounted to the wheels then they will belong to Ch. This is illustrated in Figure 3.5.

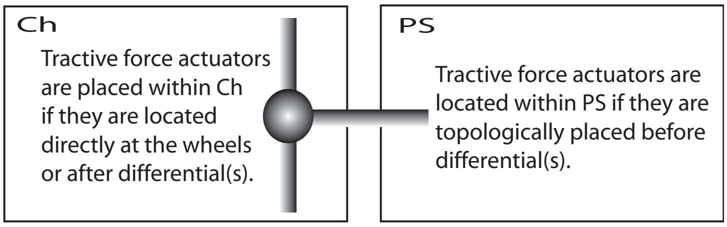


Figure 3.5: The placement of tractive force actuators.

The second physical interface between PS and Ch is electrical, allowing electrical power ($P_{el} = u \cdot i$) to be exchanged between the functions, see also Figure 3.5.2. The physical interfaces and the partitioning of tractive force actuators are further explained in Paper I and Paper III [26].

To allow for several types of PS configurations to be defined a set of building blocks are introduced:

- *Buffers* are blocks that can store energy. These can be either fuel, electrical, or mechanical[3].
- *Converters* change power from one energy form to another. These can be for example, changing chemical energy to mechanical[4], electrical energy to mechanical[5], or chemical energy to electric[6].
- *Transformers* are blocks that alter the potential within an energy form. These can be, for example, altering the mechanical to mechanical[7] potential, electrical to electrical[8] potential, or fuel to fuel[9] potential.
- *Nodes* are blocks where power flow can be divided within an energy form. These can be either mechanical, electrical, or fuel.
- *Connectors* are physical interfaces between PS and other FUs within functional level 2 such as Ch and Aux. These can be either mechanical, electrical, or fluid connectors.

[3]Example: flywheel
[4]Example: combustion engine
[5]Example: electric motor
[6]Example: fuel cell
[7]Example: Gear
[8]Example: DC/DC converter
[9]Example: pump/compressor

- *Zeros* are blocks used to illustrate the end of a power flow branch. These can be, for example, mechanical[10], electrical, or fuel energy flow branches.

To summarise, the building blocks listed above allows a variety of PS configurations to be defined. Additionally, using these building blocks also enables the possibility to formulate an energy network flow model for the highest level. Thus the same generic interface signals could be used for different PS configurations, see also Paper III.

Auxiliary Systems (Aux)

Auxiliary systems are here defined as the functions that are not related to the vehicle motion. Examples of Aux functions could be lights, air-conditioning, and lifting equipment on commercial vehicles.

External Information (EI)

The External Information function gathers together functions for communication with systems outside the vehicle, such as other vehicles or traffic flow information.

3.4.2 Proposed Functional Units for level 1

In order to control and coordinate the vehicle's basic FUs, functional level 1 introduces the following supervisory FUs:

- Driver Interpreter (DIp) -interprets the driver's intentions from DIf,in order to control the ground motion according to driver's intentions.
- Energy Management (EM) -supervises and coordinates the energy flow within PS.
- Vehicle Motion Control (VMC) -attempts to follow the desired path and coordinate the WUs within Ch.
- Strategic Control (SC) -coordinates any conflicts between EM and VMC and has finalising authority over orders sent to functional level 2.

Driver Interpreter (DIp)

DIp is the supervisory function for DIf. The signals sent from DIf are translated into a desired motion within DIp, see also Figure 3.4. It also gives feedback to the driver if the achievable limits of the vehicle are exceeded.

[10]Example: Open clutch

Energy Management (EM)

EM is the supervisory function for PS. It contains a subfunction called *SOC Controller* which calculates a State of Charge(SOC) target by considering vehicle states such as speed, GPS[11] positioning, and traffic flow information. Additionally it can also include a subfunction called *Power Controller* that coordinates the energy flow within PS. Power Controller can be utilized in different ways, such as to minimize energy losses or minimize fuel consumption.

Vehicle Motion Control (VMC)

VMC is the supervisory function for Ch. It contains two subfunctions *Path Controller* and *Force Distributor*. The desired motion requests from DIp are handled by VMC's subfunction *Path Controller*, which tries to follow the desired path and calculates the global horisontal forces F_x, F_y and the global yaw moment M_z. These requests are then sent to the *Force Distributor* which distributes the global forces and global moment into specific WU forces $(f_{x,i}, f_{y_i}, f_{z,i})$, where i is the WU number. This is also illustrated in Figure 3.4. Further details on VMC and case studies can be found in Paper II.

Strategic Control (SC)

SC handles the interaction between the different functions found on level 1. All requests from top level functions have to pass through SC before orders are finalised to functional level 2. If either EM or VMC have defined themselves to be in a critical state, arbitration between the requests are made by SC. SC has no knowledge of or control over VMC or EM functions. It only makes decision on the estimated states given by VMC and EM, see also Paper III and Paper IV.

3.5 Interfaces

Interfaces are the signals and physical connectors between FUs, see also Definition 3.1.3. To make functional level 1 as hardware independent as possible, Definition 3.2.2 was used as a guideline to delineate the proposed generic interface signals. Section 3.5.1 shows the proposed interfaces between FUs in Level 1 and Level 2. Section 3.5.2 discusses the proposed interfaces between FUs within Level 1.

[11] Global Positioning System

3.5.1 Proposed Interfaces between Level 1 and Level 2

In order to make FUs generic for several HEV configurations it is important that the interfaces between all FUs and Functional levels be suitable for a wide range hardwares. The following sections explain the proposed interfaces for FUs DIf-DIp, Ch-VMC, and PS-EM.

Interfaces for DIf-DIp

For DIf and DIp only the driver's interface signals needed to control the desired vehicle motion are included. DIf Communicates mainly with the supervisory function DIp. DIf sends out the following interface signals:

```
bus.DIf.longitudinal_desired
bus.DIf.lateral_desired
bus.DIf.yaw_desired
bus.DIf.direction_desired
bus.DIf.cruise_desired
```

The longitudinal, lateral, and yaw signals describe the driver's intentions defined by normalised signals, $[-1, 1]$. The direction signal describes whether the car is moving in forward or reverse, which is defined by an integer, $\{-1, 1\}$. The cruise signal defines if the cruise control is on or off. One DIf hardware that would be immediately suitable for this signal setup is a three axis joystick[12].

The DIp function receives the signals from DIf and determines the desired path. This path is defined by the desired tangential speed to the path v_{des}, the desired curvature of the path ρ_{des}[13], and the desired vehicle slip angle along the path β_{des}. DIp sends out the following interface signals:

```
bus.DIp.v_desired
bus.DIp.rho_desired
bus.DIp.beta_desired
bus.DIp.longitudinal_limit
bus.DIp.lateral_limit
bus.DIp.yaw_limit
```

The longitudinal, lateral, and yaw limits are sent back to DIf from DIp through SC which finalises the orders. These limits are used to give driver feedback which is a function handled by DIf.

In order to handle a conventional mechanical steering within the generic architecture framework there needs to be a connector of type mechanical between DIf and Ch. The steering is then out of scope for VMC function to influence[14]. How-

[12] Three axis joysticks have forward-reverse, left-right, and left-right turn motions.

[13] ρ_{des} is defined by the inverse of the curve radius R≥ 1, this gives $\rho_{des} \in [-1, 1]$.

[14] In this case the difference between possible lateral forces and estimated lateral forces is set to

ever, if power steering is available then small lateral force limits would be sent from Ch to VMC. In this scenario the DIp function serves only as an observer of the lateral motion.

Interfaces for Ch-VMC

The generic interfaces between Ch and VMC were determined by using an abstraction model to define and allow for the constraints of different chassis configurations. This abstraction model is illustrated in Figure 3.6. The ellipses shown on level 1 in Figure 3.6 describe the tyre force limits. These limits depend both upon actual normal force, and on road and tyre characteristics for each individual WU.

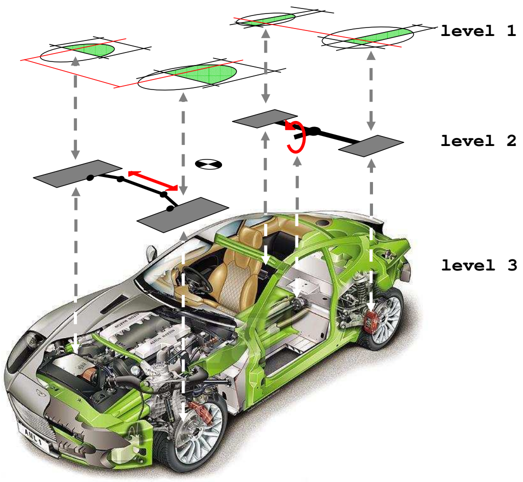


Figure 3.6: Illustration of an abstraction model of Chassis consisting of a vehicle (Level 3), the current chassis configuration of four WUs, two restrictors (rack steer and differential), and one body (Level 2), and the force distribution problem in VMC (Level 1). The car is from [27].

Level 1 in the abstraction model is seen as a force distribution problem, Paper II and [28]. When and if Autonomous Wheel Corners [7] is introduced, the interface of using WU force limits will allow for several ways to generate the desired vehicle ground motion. This can be compared to control allocation problems common in aircrafts and underwater vehicles.

The possible WU forces are limited by their actuators and tyre friction limits, as illustrated in Figure 3.7. Longitudinal actuator limits are defined by the mechanical brakes, electric motors, and differentials. Lateral actuator limits are

zero for WUs which are the directly steered, $\Delta f_y = 0$.

defined by the steering and camber. The actuator limits for a WU are often a result of a combination of actuators, for example the total of mechanical brake and electric motor forces.

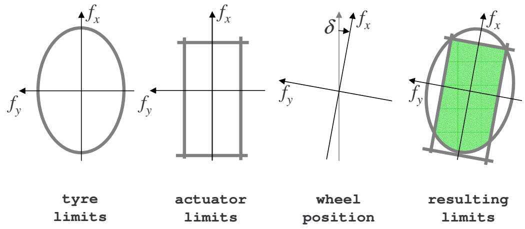


Figure 3.7: Possible WU forces are limited by the tyre friction ellipse and actuator limits. The shaded area illustrates the possible forces that can be achieved by the WU.

Ch mainly communicates with the supervisory VMC function. Ch sends out the following interface signals:

```
bus.Ch.Bd.states_estimated
bus.Ch.WUi.Bd
bus.Ch.WUi.delta_estimated
bus.Ch.WUi.r_estimated
bus.Ch.WUi.Rs
bus.Ch.WUi.fxy_estimated
bus.Ch.WUi.fxy_ellipse_limit
bus.Ch.WUi.fxy_el_limit
bus.Ch.WUi.fxy_mech_limit
```

The estimates of chassis velocities and accelerations are given in a body state vector. The WUi.Bd describes the i-th wheel and the body to which it is mounted. Rotation of tyre force mapping is given in the WUi.delta signal. The signal r estimated provides the location vector for the WU from the centre of gravity of the body upon which it is mounted. The type of restrictors connected to the wheel are provided in the WUi.Rs signal. Finally, the estimated WU force states and limits are also given.

VMC sends out the following interface signals:

```
bus.VMC.el_F_res_desired
bus.VMC.mech_F_res_desired
bus.VMC.v_res_desired
bus.VMC.v_estimated
bus.VMC.rho_estimated
bus.VMC.beta_estimated
bus.VMC.v_limit
bus.VMC.rho_limit
```

```
bus.VMC.beta_limit
bus.VMC.fxy_desired
bus.VMC.state
```

The three first interface signals are for EM to receive the resulting desired forces generated by electrical, el_F_res, or mechanical, mech_F_res, actuators and the desired vehicle velocity along the path, v_res_desired. The next six signals are for communication with DIp over the estimates of states and limits of the desired path (v, ρ, β). VMC distributes the wheel forces within Ch with fxy_desired, see also Paper II and [28]. Finally, VMC sends a state value $\{0, 1\}$ to SC to declare if the vehicle is in a critical dynamical state, for more information see Paper III.

Interfaces for PS-EM

PS mainly communicates with its supervisory function EM. There are several ways of configuring the interfaces for PS and EM. This section discusses three different suggested approaches. The set of interface signals between PS-EM will be further investigated as future work by the author.

EM using only a SOC controller

If the EM function is only made up of a SOC Controller, then PS only has to send the information required by the SOC Controller within EM. In this situation the Power Controller would be located within PS itself, instead of within EM, see also Section 3.4.2.

In this scenario the SOC controller within EM would need the following signals from PS:

```
bus.PS.Pbuff_i_limit
bus.PS.buff_potential_i_limit
bus.PS.buff_efficiency_i_limit
bus.PS.buff_SOC_estimated_i
```

where the potential, efficiency, and SOC estimation from the i:th buffer is sent to EM.

The Power Controller within PS would need the following signals from EM:

```
bus.EM.SOC_Target_i
```

where the only signal from EM to PS would be the desired SOC target value for the i:th buffer.

EM using SOC- and Power Controller

HEVs and FCVs will have several possible configurations within PS. In this case abstraction models can be used to describe the PS in a systematic way, as shown in Figure 3.8. In Level 2 building blocks are used to set up the model, as described in Section 3.4.1 and Paper III. In Level 1 the model is used to generate a network energy flow problem with different costs for using the different paths between the nodes. The cost function could be to minimise the energy losses for the energy flow. This is illustrated in Paper III.

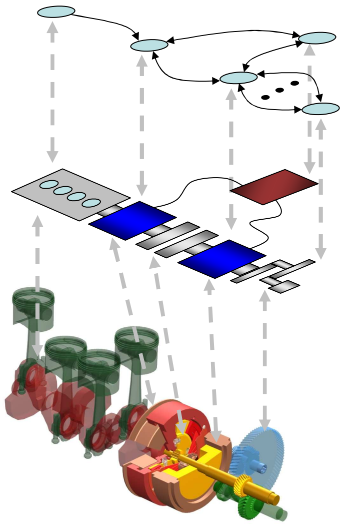


Figure 3.8: Illustration of an abstraction model of Power Supply. A Power Supply configuration (Level 3), described by building blocks (Level 2), and the actual energy flow problem (Level 1). The powertrain is from [29].

In this case when EM includes both SOC controller and Power Controller the following interface signals are suggested from PS:

```
bus.PS.Rm
bus.PS.power_ij_estimated
bus.PS.potential_ij_estimated
bus.PS.efficiency_ij_estimated
bus.PS.power_ij_limit
bus.PS.potential_ij_limit
bus.PS.efficiency_ij_limit
bus.PS.SOC_estimated
```

The Rm is a relation matrix which defines how the i times j building blocks are topologically connected. Estimations of current power, potential, and efficiency are also suggested as interface signals. The limits of power x, potential ϕ, and efficiency η of the flow ij path are important in order for EM to be able to perform the power coordination of the PS, an illustration is shown Figure 3.5.1. Finally, the estimation of the SOC is also given. EM is suggested to have the following interface signals:

```
bus.EM.x_desired
```

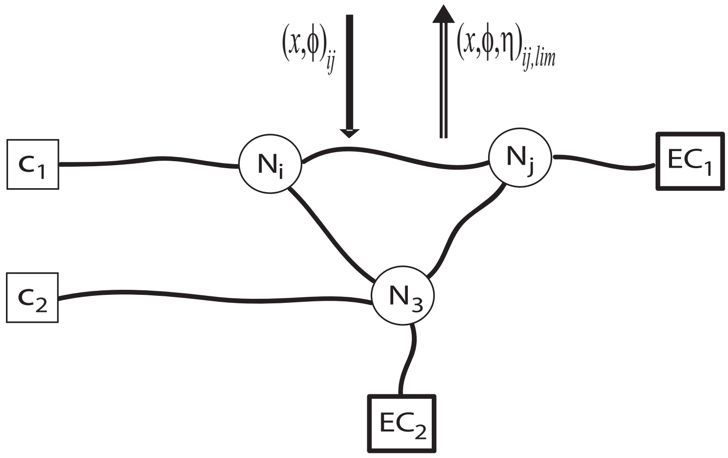


Figure 3.9: Illustration of an abstraction model of PS where the request signals and information about the ij-energy path is shown. Connectors=C, Energy Carriers=EC, and Nodes=N.

```
bus.EM.el_F_res_limit
bus.EM.mech_F_res_limit
bus.EM.power_aux_limit
bus.EM.state
```

The desired state vector, x_desired, is sent to PS. The resultant limits of the electrical, el_F_res_limit, and mechanical forces, mech_F_res_limit, are sent to VMC through SC. Limits of power are sent to Aux. Finally, EM sends the critical state to SC.

EM with a Simple SOC- and Power Controller as used within SMC

In the Scale Model Car the following interface signals were used from PS:

```
bus.PS.Pdem
bus.PS.Pppu
bus.PS.Pbuff
bus.PS.SOC
bus.PS.rot_speed
```

where the estimates of performed requests of PPU and buffer power are given by PS. Additionally the estimated actual SOC and the rotational speed of the electric motor are also provided by PS.

EM had the following corresponding set of signals:

```
bus.EM.x_vel
bus.EM.Pppu
```

```
bus.EM.Pbuff
bus.EM.brake
bus.EM.state
```

where desired vehicle velocity request is based upon EM status. The desired PPU and buffer power are also given. EM decides which brake mode to use, and whether or not, the motor should use regenerative braking or not. Finally, EM also decides the energy state within the vehicle. Further details can be found in Paper IV. To Summarise, there are several ways to set up the generic interface signals between PS and EM. However, according to this research, the most appealing method is to use building blocks to define the configuration of PS and generate an energy flow network model for functional level 1 which defines the interface signals between PS and EM.

3.5.2 Proposed interfaces between FUs within Level 1

This section illustrates the coordination found within Level 1 FUs, emphasizing especially the interaction between EM and VMC.

Interfaces for coordination between EM and VMC

In order to coordinate between EM and VMC, information needs to be passed between them. VMC has to define how much electrical and mechanical force must be generated to assure the desired motion. Additionally, VMC needs to receive information from EM on any existing mechanical and electrical force limits which can be delivered by PS, see Figure 3.5.2.

Both EM and VMC send state signals to SC. If there is a critical state the SC decides which of the two will be prioritised. This will directly affect how force limits sent by EM to VMC will be considered. Ways in which the state controller within SC can be formalised are shown in Paper III and Paper IV.

3.6 Summary of Proposed FUs and Main Interfaces

To summarise, the proposed FUs within the suggested architecture are shown in Figure 3.6. In this figure EM contains both a SOC Controller and a Power Controller, see further 3.4.2. The critical state is decided by Path Controller inside VMC and by SOC Controller inside EM. The SC has two main functions, arbitrating and finalising the desired values into orders for function Level 2. The black solid lines shows the signal flow between EM and VMC and how states are sent to SC.

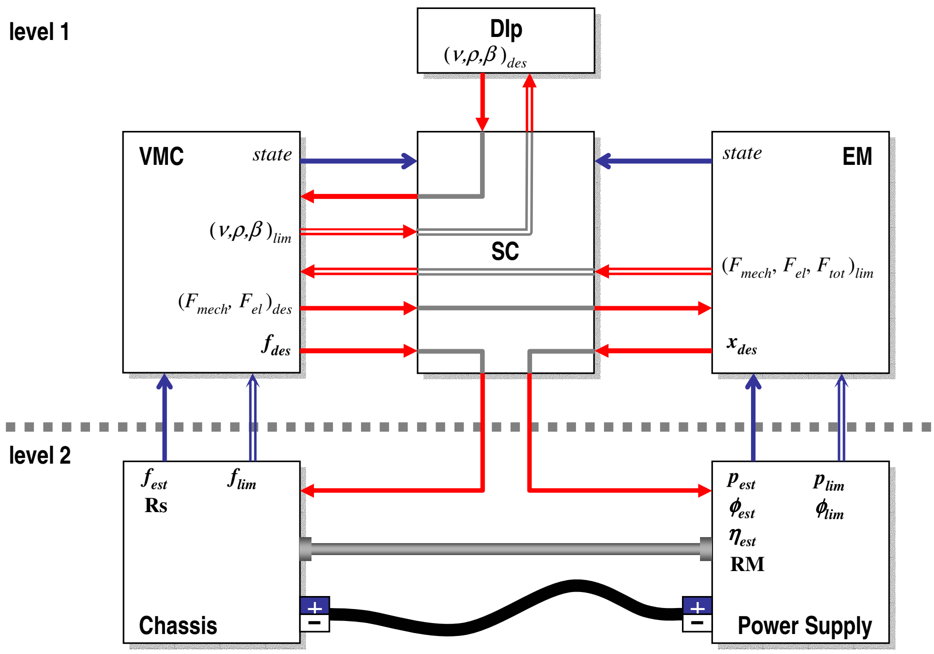


Figure 3.10: Part of the suggested vehicle control architecture with focus on the interaction between VMC/Chassis and EM/Power Supply. Double lines indicate limits and single lines are desired signals and states.

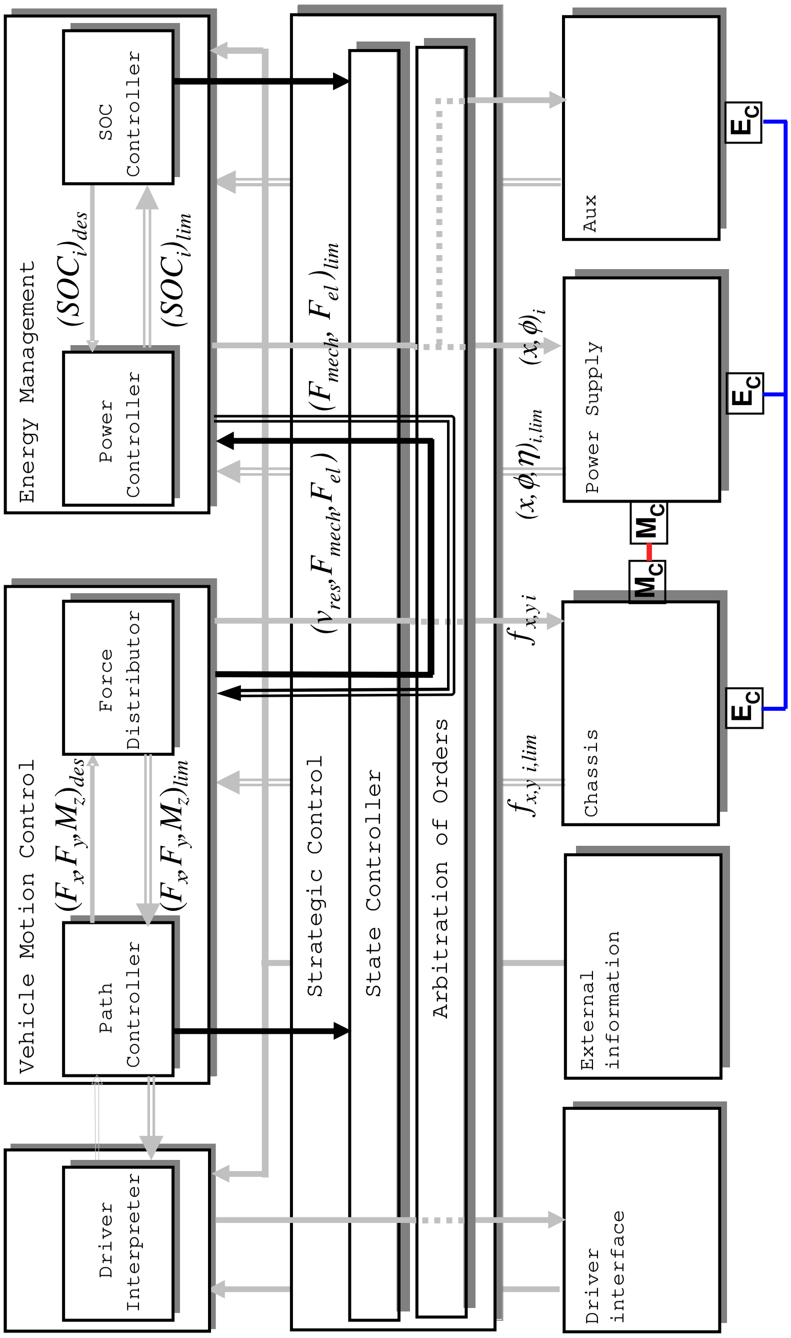


Figure 3.11: Proposed FUs with main interfaces. The black lines indicate the interaction between VMC-EM-SC.

Chapter 4

Prototypes

This chapter discusses the implementation and testing of many of the ideas for the development of generic architecture, such as the use of three functional levels and FUs, both by virtual models and real prototypes. Section 4.1 discusses the virtual prototypes while Section 4.2 discusses the HEV Scale Model Car prototype.

4.1 Virtual Prototypes

Virtual prototypes were built in order test these ideas in the beginning stages. Modelica was found to be a useful tool to check if modularity was achieved in describing the functional units within the Vehicle System Controller had been achieved. The Modelica implementation is gathered in the Modelica library GenericVehicle `GenericVehicle`, which is further explained in [19]. The main model consists of the FUs in level 1 and level 2 and in the library these represent sub-packages. The main model is illustrated in Figure 4.1, consisting in this case of the following sub packages `DriverInterpreter`, `VehicleMotionControl`, `StrategicControl` and `EnergyManagement`. Additionally, `DriverInterface`, `Chassis`, `PowerSupply` and `Auxiliary Systems` are found on level 2. Finally the `Bus` package contains interface signals necessary for the information exchange between the FUs.

The `Chassis` sub package is based on the `VehicleDynamics` library [30] components for three dimensional Multi Body System (MBS) chassis modelling. Additionally the `PlanarMultiBody` library [31] has been used to model simpler planar chassis models. The latter are suitable when influences of load transfer due to roll or pitch can be neglected since these models speed up simulation time considerably.

The `PowerSupply` sub package includes different types of configurations such as mapped models of a combustion engine with integrated starter generator, an automated manual gear box together with battery as a buffer, and a simple fuel cell model together with a battery. Further details can be found in [19].

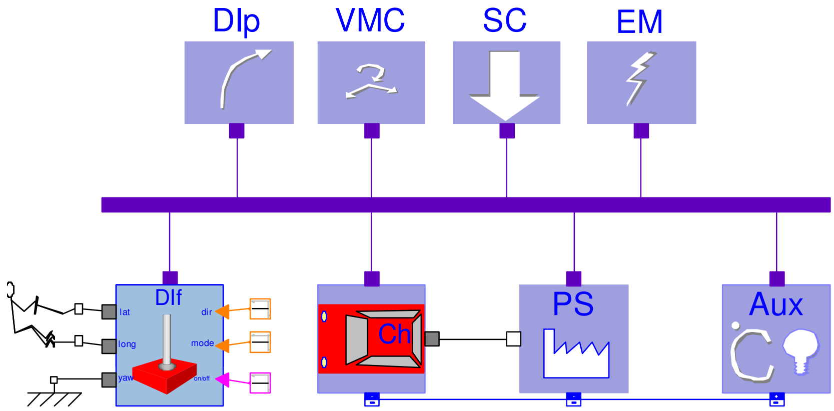


Figure 4.1: Main Model architecture in Modelica model `GenericVehicle`. The main functions within functional level 1 and 2, from [19].

However, the suggested functionality within VMC and EM was partly tested with Matlab toolboxes. This was done mainly because optimisation functions are already implemented within Matlab and its toolboxes. The use of matlab was done both for the study of abstraction models for Ch and the design of the VMC path controller and force distributor Paper II and [28]. The vehicle was modelled as a 3 degrees of freedom, longitudinal, lateral and yaw motion. The vertical load change on the tyres were considered by including the vehicle's accelerations. The Magic formula was used for modelling the tyres [32]. In Figure 4.2 a braking manoeuvre from 20 m/s to stand still is shown for a futuristic car configuration, with four individual ACM, [7]. During the braking a split-μ test was performed after 4 seconds where the ground friction was reduced from 1 to 0.1 for WU 1 and WU 3.

Figure 4.2 and 4.3 presents some interesting variables related to the braking on a split-μ. As can be seen, the vehicle's speed was decreased as expected. After 4 seconds the vehicle entered the split-μ region and the wheel forces were redistributed. WU1 and WU3 used low longitudinal forces, and WU2 and WU4 used high longitudinal and lateral forces to compensate for the loss. The vehicle's deceleration is decreased a little bit, but the controller handles this case very well. Further details about the used algorithm in VMC and simulations for different chassis configurations can be found in Paper II and [28].

The first version of the Power Controller of PS within EM was tested using Matlab/Tomlab. By using the building blocks as discussed in Section 3.4.1 the PS configuration could be built in modular fashion. By using an optimisation

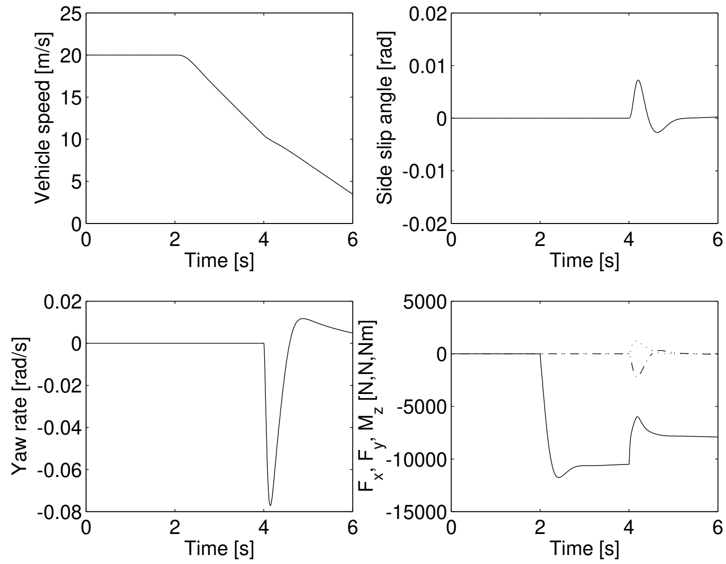


Figure 4.2: Vehicle speed, side slip angle, yaw rate, and 'virtual' control signals as function of time during the braking manoeuver. In the lower right figure the solid line corresponds to the longitudinal force, the dashed line corresponds to the lateral force and the dotted line corresponds to the yaw torque.

function the necessary generic interface signals could be identified for PS and EM, see further Paper III.

4.2 Scale Model Car Prototype

The ideas presented here for reusable control architectures were implemented and tested in a remote controlled Scale Model Car (SMC) of size 1:5. The design of the SMC included the concepts of hierarchical functional partitioning with three functional levels in addition to the suggested functional units described in Section 3.4. Reusable interface signals were defined between the functional units and an arbitration switch was used to make decisions over the giving of priority to either VMC or EM. The implemented control architecture is described in Paper IV. Although the functional partitioning was hierarchical in nature, the computational architecture was centralised within the SMC system.

The SMC is configured as a series HEV with a PS containing a battery as a PPU, supercapacitors as a buffer, a DC/DC converter, and an electric motor. The SMC is rear wheel driven and front steered. The sizing and design process is described in [8]. The actual building of the car was performed by thesis workers

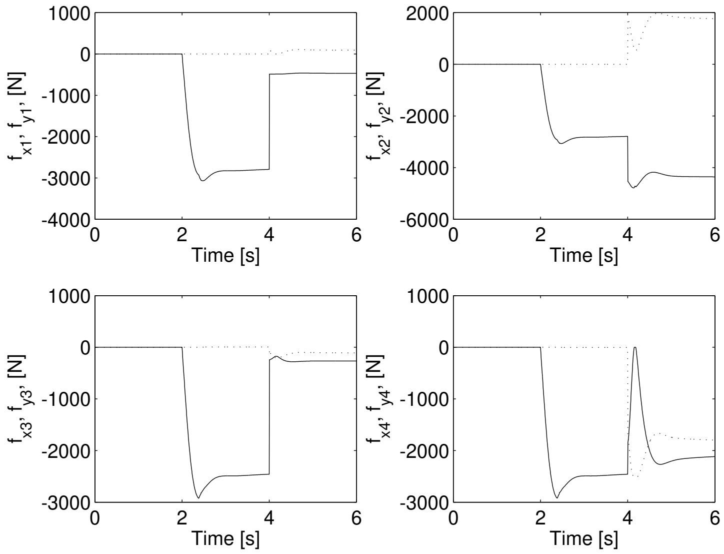


Figure 4.3: The wheel forces, (f_{xi}, f_{yi}), as function of time during the braking manoeuver. The solid lines correspond the longitudinal forces and the dotted lines correspond to the lateral forces.

[9] and [10].

To begin with a technician downloads the VSC code to the Digital Signal Processor(DSP) card[1]. The downloaded VSC code must interact with different input and output signals. A Driver gives input such as desired longitudinal and lateral motion, braking, and power switch[2]. Driver gives input such as desired longitudinal and lateral motion, braking, and power switch. Since the SMC is a HEV the use of either mechanical or regenerative braking is decided by the Vehicle System Control (VSC) Code. Sensor signals such as WU rotational speed, motor speed, accelerometers, current, and voltage sensors, are interpreted and used to estimate the vehicle internal states. These input signals are processed by the VSC and final output request signals are sent to actuators such as the electric motor, DC/DC voltage, steer servo, and mechanical brake servo. A system context class diagram of the current configuration of the SMC is shown in Figure 4.4, illustrating which sensors and actuators are found in the current configuration of the SMC.

[1]The DSP used is a TMS320LF2407A processor from Texas Instruments which is mounted on a evaluation module from Spectrum Digital. The downloaded code is written in C, which is supported by the development Code Composer from Texas Instruments.

[2]The SMC is controlled by a RC system, a Hitec Laser 4 FM transmitter, and a Hitec HFS-04MG receiver.

In Figure 4.5 three accelerations were performed with the SMC on a ground surface with low friction. The right plot in Figure 4.5 shows that the VMC state was critical during the accelerations and that the angular velocity of the rear wheel was oscillating around the front wheel value, as shown in left plot. This means that VMC decreased the desired velocity from DIp during the accelerations. Further details of the algorithms used within VMC can be found in Paper IV.

The SMC was also driven indoor on concrete. During these drives the simple energy management algorithm that is located within EM function was tested, see Paper IV. Figure 4.6 shows one drive situation. The drive cycle is shown in upper left plot in Figure 4.6. The SOC is shown in upper right plot in Figure 4.6. Finally the power demand on the electric motor and buffer power are shown in lower left and right plot respectively in Figure 4.6. The SOC has only small changes due to the fact that low buffer power was used and that the total available buffer energy was high compared to a single acceleration. The electric motor also has a negative power demand during deceleration especially during the highest deceleration at time 26.8s. The regenerated energy was stored in the buffer as shown in lower right plot. Another interesting observation was that the used DC/DC converter can only handle power flows of up to a maximum of 40 W out from the buffer, but yet can manage several times higher power flows into the buffer. Further details can be found in Paper IV.

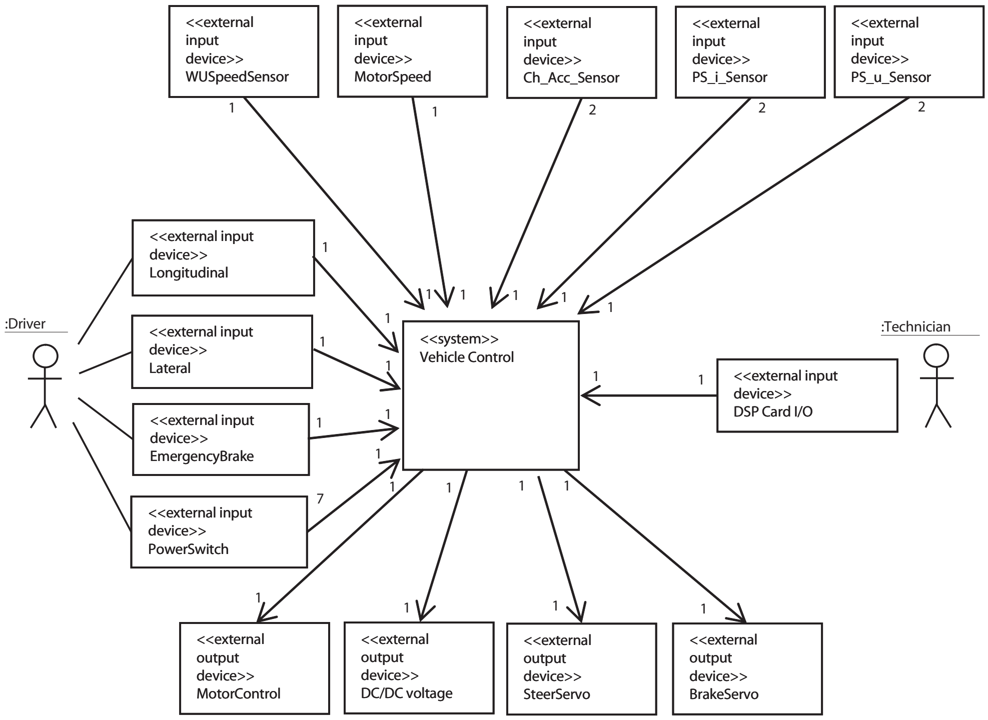


Figure 4.4: System context class diagram of the SMC according to Unified Modelling Language (UML).

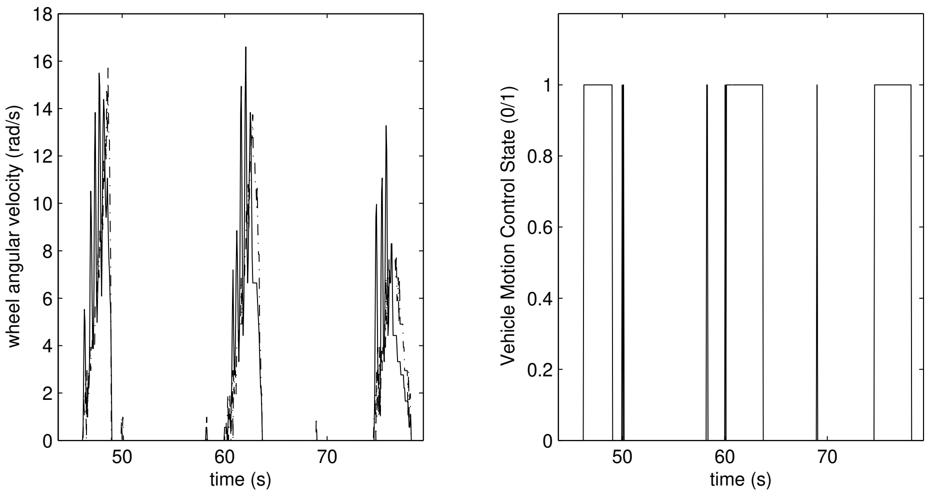


Figure 4.5: Accelerations on surface with low friction, close to ice conditions. Left plot shows front (dashed) and rear (solid) wheel angular velocity. Right plot shows the VMC state, which is critical when VMC_state = 1.

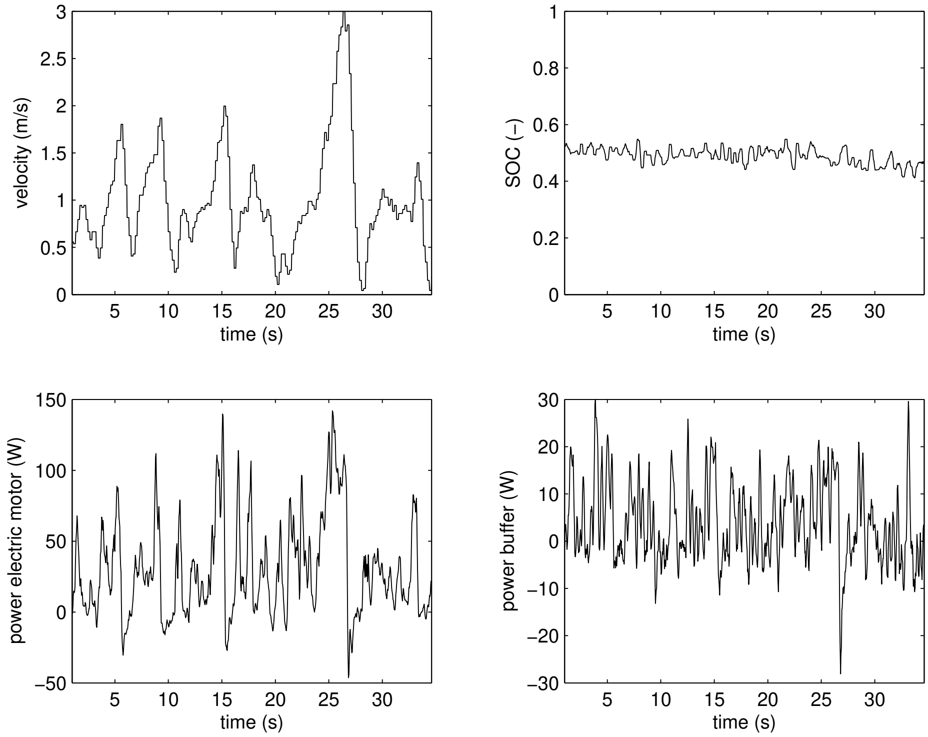


Figure 4.6: In door drive cycle test with the SMC.

Chapter 5

Concluding Remarks

The ideas presented regarding generic reusable control architecture, such as functional levels and FUs, have been implemented and tested here both by virtual computational models and real prototypes.

The suggested FUs within the control architecture have been shown to be reusable in consideration of different hardware configurations and software functionality. The key factor in making a vehicle control architecture able to handle several different types of hardware configurations is to have generic physical and signal interfaces between FUs. This is especially important between the FUs on level 1 and level 2, allowing functional level 1 to be as hardware independent as possible.

5.1 Future Work

From this research and thesis, as a foundation, more specific research questions will be addressed. Examples could be:

- Study how interfaces, such as estimates and limits on requests should be formulated between lower and higher functions, with focus on hardware used within HEVs/FCVs. *This is essential when subsystems are provided by suppliers so that enough information is sent to higher functions to allow good coordination.*

- Refine the abstractions models for Chassis and especially for Power Supply. *Although abstraction models are shown in thesis, a lot of work is still to be done especially with regards to PS.*

- Implement in real prototypes the suggested VSC or similar, and test different interfaces, especially examining how the limits of requests affect the system performance. *Verification by real prototypes has been found useful.*

- Study how computational partitioning affects different types of functional partitioning. *This is to assure computational and system safety aspects.*
- Generate objective measurements for the Vehicle System Controller with respect to its reusability, realtime performance, and simplicity. *To give a tool for deciding when the architecture is reusable enough.*

Chapter 6

Summary of Appended Papers

6.1 Paper I

This paper addresses the need for reusable control architectures including a short review of future automotive aspects. A suggestion is given on how a generic control architecture could be designed for a Hybrid Electric Vehicle (HEV), using a complete vehicle controller to allow for easy implementation of new subsystems. The main functions within the generic control architecture are defined and explained, introducing three functional levels. It is suggested that at the top, functional level 1, the following functions be included: Driver Interpreter, Vehicle Motion Control, and Strategic Control. At functional level 2 the vehicle's basic functions are introduced: Driver Interface, Chassis, Power Supply, Auxiliary Systems, and External Information. Functional level 3 includes the actuators and sensors. The allocation of algorithms within main functions are also discussed. For example, the main function Driver Interpreter is suggested to include the following three sub functions: Defining the desired path from the driver's interpreted intentions, using external information to automate the driver's intentions, such as in cruise control, and giving feedback to the driver on the vehicle's dynamical limits. Finally included in this article is the modelling of the architecture in Modelica and the demonstration of two different configurations of Sport Utility Vehicles.

6.2 Paper II

This paper explores the option of replacing some of today's most common types of vehicle safety critical systems, such as Anti-lock Braking System, Anti-Skid, and Electronic Stability Program, with the function Vehicle Motion Controller. This function contains two smaller controllers, the Path Controller which calculates the global horizontal forces and yaw moment needed to keep a desired path and the force distributor which distributes the desired global forces into Wheel Unit forces.

An abstraction model of the chassis is used to allow the same Vehicle Motion Controller to be used with several different chassis configurations. This abstraction model uses the wheel force limitation concept to define which wheel forces can be applied at each wheel.Three different types of Chassis configurations are simulated in Matlab/Simulink: A chassis with four Autonomous Wheel Corners, allowing the wheels to then independently steer, drive, and brake, a front steered chassis with a motor driven rack steer on the front wheels and rear wheels driven by a differential, and a conventional chassis which is front steered and rear wheel driven by a differential. The simulation showed that the same suggested Vehicle Motion Controller could be used for all three cases during different driving scenarios.

6.3 Paper III

This paper shows how a reusable functional partitioning of tractive force actuators can be made for Hybrid Electric Vehicles, which contain several different types of tractive force actuators, such as electric motors directly connected to a wheel or on a differential in combination together with an combustion engine. It is also shows how the interaction between Vehicle Motion Controller and Energy Management can be handled by introducing critical state controllers. These states are then sent to Strategic Control which makes the arbitration whether Vehicle Motion Control or Energy Management will have priority.

Energy Management is suggested to include the following controllers: State Of Charge, Power Controller, and Critical State. In order to handle the Power Controller for several different HEV configurations an abstraction model of Power Supply is introduced. The abstraction model uses building blocks such as Buffers, Converters, Nodes, Transformers, Sums, and Connectors to define the Power Supply configuration and the physical connectors between Chassis and Power Supply. An optimisation function is also suggested for the Power Management controller. Two different types of parallel HEV configurations are exemplified,both with a combustion engine, Integrated Starter/Generator, clutch, automated manual gear box, and a battery as a buffer. However one of them has All Wheel Drive, whereas the other has Electric four-wheel drive. The energy flow within Power Supply is calculated by Matlab/Tomlab by using a mixed integer solver to handle the different gear selections.

6.4 Paper IV

This paper discusses the implementation of a reusable control architecture in the Digital Signal Processor of a remote controlled scale model car of size 1:5. The Power Supply configuration is a series HEV with a battery as primary energy carrier to emulate a fuel cell, and super capacitors as secondary energy carrier.

The reusable control architecture uses three functional levels: Level 1 contains the Driver Interpreter, Vehicle Motion Control, Energy Management, and Operative Control; Level 2 contains the Chassis, Driver Interface, and Power Supply; Level 3 contains the actual actuators and sensors.

The Scale Model Car also has the following actuators and sensors within Chassis: mechanical braking by a servo on the front wheels, rack steering by a servo on the front wheels, a front wheel speed sensor, motor speed sensor, and longitudinal and lateral accelerometers. The

Driver Interface includes a remote control receiver. Finally, Power Supply Contains an electric motor for traction and regenerative braking, a DC/DC converter, and current and voltage sensors to measure the motor and buffer. The reusable control architecture was shown to easily allow actuators and sensors to be added and exchanged with minimum affect on the top level functions.

Bibliography

[1] B. Hardung, T. Kölzow, and A. Krüger. Reuse of software in distributed embedded automotive systems. *Fourth ACM International Conference on Embedded Software (EMSOFT 2004), PISA, Italy*, 2004.

[2] E. Coelingh, P. Chaumette, and M. Andersson. Open-interface definitions for automotive systems, application to a brake by wire system. *SAE 2002-01-0267*, 2002.

[3] AUTomotive Open System ARchitecture(AUTOSAR). http://www.autosar.org/. 2004.

[4] British Petroleum. Statistical review of world energy. *http://www.bp.com*, 2004.

[5] C. Azar, K. Lindgren, and B. A. Andersson. Global energy scenarios meeting stringent co_2 constraints- cost-effective fuel choices in the transportation sector. *Energy Policy 31*, 2003.

[6] H. Wallentovitz, J.W. Biermann, R. Bady, and C. Renner. Strukturvarianten von hybridantrieben. *Verein Deutscher Ingenieure Berichte nr 1459*, 1999.

[7] S. Zetterström. Electromechanical steering, suspension, drive and brake modules. *VTC 2002, Vancouver, Canada*, 2002.

[8] J. Hellgren. *A Methodology for the Design of Cost Effective Hybrid and Fuel Cell Powertrains*. PhD thesis, Dep. of Machine and Vehicle Systems, Chalmers University of Technology, Sweden, 2004.

[9] M. Rönnberg. *Implementation of a Control System for a Scaled Series Hybrid Electric Vehicle*. Master thesis report, Dep. of Machine Vehicle Systems, Chalmers University of Technology, Sweden, 2004.

[10] D. Gomes. *Design and Development of a Hybrid Electric Scale Model Car*. Master thesis report, Dep. of Machine Vehicle Systems, Chalmers University of Technology, Sweden, 2004.

[11] CAN in Automation (CiA) international users and manufacturers group. http://www.can-cia.de. 2004.

[12] H. Kopetz and G. Bauer. The timetriggered architecture. *In The TwentyThird International Symposium on Fault Tolerant Computing. IEEE*, 1993.

[13] H. Kopetz and G. Grunsteidl. TTP a timetriggered protocol for faulttolerant realtime systems. *Proceedings of the IEEE, vol 91:1*, 2003.

[14] The FlexRay Consortium . http://www.flexray.com. 2004.

[15] J.C. Burba. Consideration of control system architecture in Electric and Hybrid Electric Vehicles. *Proceedings of the 18th International Electric Vehicle Symposium*, 2003.

[16] M. Gäfvert. *Topics in Modeling, Control, and Implementation in Automotive Systems*. PhD thesis, Dep. of Automatic Control, Lund Institute of Technology, Sweden, 2003.

[17] A. Phillips. Functional decomposition in a vehicle control system. In *Proceedings of the American Control Conference*, 2002.

[18] M. Cantor. Thoughts on functional decomposition. *The Rational Edge*, http://www.therationaledge.com/content/apr_03/f_functionalDecomp_mc.jsp, 2003.

[19] L. Laine and J. Andreasson. Modelling of generic hybrid electric vehicles. In *Proceedings of the 3rd International Modelica Conference*, Linköping, November 2003.

[20] L. Laine, J. Hellgren, H. Kinnunen, and Magnus Rönnberg. *Reusable Control Architecture Implemented in a Scale Model of a Hybrid Electric Vehicle*. Techincal report, Dep. of Machine and Vehicle Systems, Chalmers University of Technology, Sweden, 2005.

[21] R. Zhang and Y. Chen. Control of hybrid dynamical systems for electric vehicles. In *Proc. of American Control Conf.*, Arlington, VA, June 2001.

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

[23] Y. Hattori et al. Force and moment control with nonlinear optimum distribution for vehicle dynamics. In *Proceedings of the 6th International Symposium on Advanced Vehicle Control*, Hiroshima, Japan, September 9-13 2002.

[24] J. Andreasson, L. Laine, and J. Fredriksson. Evaluation of a generic vehicle motion control architecture. In *The Fédération Internationale des Sociétés d'Ingénieurs des Techniques de l'Automobile (FISITA)*, Barcelona, Spain, May 2004.

[25] J. Andreasson. *Hybrid Electric Vehicles, Aspects on Driving Dynamics and Vehicle Architecture*. Licentiate thesis, Department of Aeronautical and Vehicle Engineering, KTH, Sweden, 2004.

[26] L. Laine and J. Andreasson. Reusable functional partitioning of tractive force actuators applied on a parallel hybrid electric vehicle. In *7th Int. Symposium on Advanced Vehicle Control, AVEC*, HAN University, Netherlands, August 2004.

[27] Auto Express. http://www.autoexpress.co.uk/news/ae_news_story.php?id=36274. 2004.

[28] J. Fredriksson, J. Andreasson, and L. Laine. Wheel force distribution for improved handling in a hybrid electric vehicle using nonlinear control. In *Proceedings of 43rd IEEE Conference on Decision and Control*, Bahamas, December 2004.

[29] Stridsberg Powertrain AB. http://www.powertrain.se/. 2004.

[30] J. Andreasson. VehicleDynamics library. In *Proceedings of the 3rd International Modelica Conference*, Linköping, November 2003.

[31] J. Andreasson and J. Jarlmark. Modularised tyre modelling in Modelica. In *Proceedings of the 2nd International Modelica Conference*, Oberpfaffenhofen, March 2002.

[32] E. Bakker, H.B. Pacejka, and L. Lidner. A new tire model with application in vehicle dynamics studies. *SAE Transactions, paper 890087*, 1989.

Paper I

Generic Control Architecture applied to a Hybrid Electric Sports Utility Vehicle

in

Proceedings of the Electric Vehicle Symposium, (EVS 20), Los Angeles, USA, November 2003.

From Proceedings of Electric Vehicle Symposium (EVS) 20, Los Angeles, CA, Nov 2003

Generic Control Architecture applied to a Hybrid Electric Sports Utility Vehicle

Leo Laine, M. Sc.
Div. of Machine and Vehicle Systems, Chalmers University of Technology
SE-412 96 GÖTEBORG, Sweden
Phone: +46 31 772 58 52, Fax: +46 31 772 13 80
leo.laine@me.chalmers.se

Johan Andreasson, M. Sc.
Div. of Vehicle Dynamics, Royal Institute of Technology (KTH)
SE-10044 STOCKHOLM, Sweden
Phone: +46 8 790 77 14, Fax: +46 8 790 93 04
johan@fkt.kth.se

Abstract

For future vehicles it is a necessity to have tight integration between different actuators/sensors. Here, functional decomposition is utilized on a Hybrid Electric Vehicle to construct a generic hierarchical control architecture.

Specific functions are identified and allocated in different functional levels. Three functional levels are suggested; main control level, subsystem level, and actuator/sensor level.

The main control contains a *driver interpreter*, *energy management, vehicle motion control* and a *strategic control*. These main functions are made hardware independent and independent of hybrid configuration. The subsystem level contains the following: *driver interface*, *chassis*, *power supply*, and *auxiliary systems*.

The suggested control architecture is validated in an object oriented modelling language. Two different power supplies (serial) and (parallel) were implemented for a Hybrid Electric Sport Utility Vehicle and changed without affecting the contents of the Main Control level of the architecture.

Keywords: control system, communication, hybrid strategy, HEV.

1 Introduction

In order to handle the complexity of several actuators/sensors interacting in future Hybrid Electric Vehicles (HEV) one has to aim for suitable control architecture. The control architecture should not only perform well but also be reusable for different hardware configurations.

One way to achieve this goal is to construct both hardware and software in a modular fashion. These modules would have their own controller. The interface signals between the modules should be general and non specific for the actual hardware to allow easy switch of configurations. A set of modules are then grouped together to form a HEV.

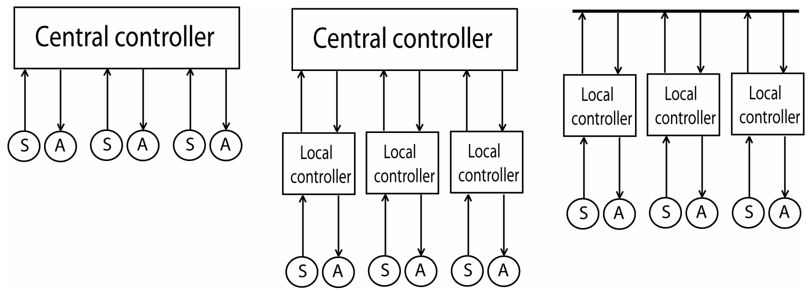


Figure 1: Illustration of a centralized (left), hierarchical (middle), and peer (right) control architecture. S=sensor, A=actuator.

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

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

The paper discusses future automotive aspects, a terminology is given and different types of control architectures are discussed, and a definition of the generic control architecture is given. The method functional decomposition is utilised and applied on a Hybrid Electric Sport Utility Vehicle and modelled in an object oriented modelling language, Modelica [1].

2 Future automotive aspects, short review

The control architectures commonly used in today's vehicles do not handle the complexity efficiently when subsystems are integrated. The automotive subsystem suppliers develop more or less independent subsystems [2]. This leads to increasing complexity when a new subsystem is introduced, as illustrated in Figure 2 (left). The vision is to have an integrated Complete Vehicle Control (CVC) where all the functions of the subsystems are emerged (right). This is even more important when new technologies based on hybrid propulsion are to be implemented.

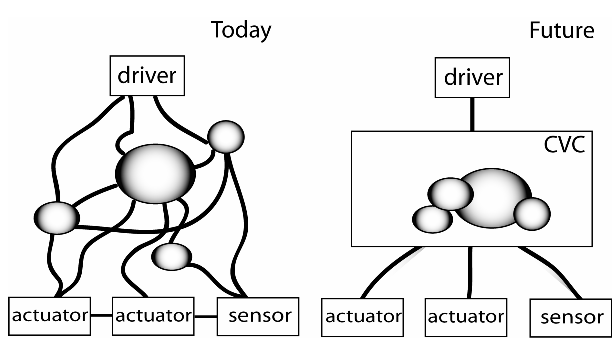


Figure 2: Illustration of how today's commonly used control architectures (left). For each new sensor, actuator or function, the complexity increases drastically. For future vehicles using a functional architecture (right), the complexity increases minimally [2].

When more onboard electric power is available by the hybrid electric propulsion the potential to replace mechanical and hydraulical actuators by electrical ones increases. This will introduce the by-wire technique in large scale in automotive vehicles. This technique will allow easier algorithmic partitioning and tighter integration of actuators to achieve better performance of the vehicle. Already some applications are implemented such as electronic throttle control and power windows. Safety critical subsystems such as steer- and, brake-by-wire must be redundant and fault tolerant before they can be implemented without mechanical backup [3]. Safety related fault tolerant x-by-wire systems for vehicles were investigated in [4]. The suggested fault tolerant architecture was demonstrated in prototype for steer-by-wire without mechanical backup. In this paper, a driver interface and a driver interpreter is introduced to handle x-by-wire control of the vehicle.

When HEVs are introduced, different configurations will be utilised and reusable control system architectures will be needed to make vehicle development feasible. In [5], a reusable architecture for hybrid powertrains is suggested. The system architecture must include a hierarchical structure that handles various engine, motors, transmission, and buffer configurations. The powertrain supervisory controller uses a torque based strategy and suites fine for parallel HEVs. In this work we try to go a step further and look at the vehicles energy sources as a Power Supply function and use force and power based strategy to control the Power Supply. This allows serial, parallel and split HEV configurations.

An open architecture for networking the control systems of an automobile called CARTRONIC was developed by Bosch GmbH [6]. It is an ordering concept for all vehicle control. The communication is divided into orders, responses, inquiries and requests. A hierarchical flow of orders is used where the vehicle coordinator places the orders and detects conflicts. Here, a similar function is performed by Strategic Control.

In this paper, all components in a wheel are seen as one function for applying force to the ground. The wheel unit function allows tight integration of the different actuators for applying longitudinal, lateral, and vertical forces within a wheel. This Wheel Unit can contain actuators such as braking, traction, suspension, and steering. The wheel as the centre of motion is also acknowledged as second x-by-wire generation in [3]. An example of how future wheel units can be designed is shown in [7]. A more detailed description of how the desired global forces are distributed to the wheel units is shown in [8].

3 Terminology

To be able to define a generic control architecture for HEVs some of the used terms are explained in this Section.

- *Complexity:* The number of actuators/sensors that have to interact defines the level of complexity.
- *Centralized control architecture:* A single controller which computes control signals for all actuators of the vehicle and has complete knowledge of the entire system.
- *Peer to Peer control architecture:* All subsystems have their own control block has knowledge of some (or all) remote states in addition to all local states. There is no supervisory control block with global knowledge of the system.
- *Function:* When something is performed, e.g. applying driving force to the wheels. This should not be confused with the specific actuators. Different actuators or sensors can sometimes perform the same task.
- *Functional decomposition*: By identifying the different functions a vehicle have one can declare the dependency between the functions and decide the hierarchy within the functions.
- *Functional level*: Depending on the function it is placed in different levels. The lowest functional level is the control of a specific actuator e.g. an electric machine for applying

driving torque, next level is the subsystem control, and the highest is the main control which controls and integrates all subsystems.

- *Generic interface signals*: The interface signals between different functions should be made hardware independent.
- *Generic control architecture*: A reusable control architecture that is not hardware dependent or configuration dependent.
- *Hierarchical control architecture:* All subsystems have their own controller (with local state knowledge) and there also exist a supervisory controller with knowledge of the entire vehicle.
- *Power supply:* Onboard energy sources in the vehicle.
- *Reusable:* The same software/hardware can be utilised in different configurations. Only small modifications should be needed. Examples of hardware configurations are parallel, serial, and split for HEV.
- *Subsystem:* A part of the whole system with clearly specified purpose, e.g. mechanical brake actuators/sensors with its control. Note that several subsystems may corporate to perform the same function, e.g. the mechanical brake subsystem together with the wheel motors can generate brake torque.

4 The suggested generic control architecture

There are different reasons for choosing a certain type of architecture. The centralized control architecture can always outperform the hierarchical and the Peer architecture. The hierarchical architecture also introduces additional conditions by using generic control signals. But if one considers the design and engineering benefits then the hierarchical architecture is a suitable partitioning scheme for HEV. In [9] hierarchical partitioning is recommended. Different partitioning schemes are also discussed in [10] and [11].

4.1 Definition of a generic control architecture for HEVs

By using the terminology stated in Section 3 one can now define the generic control architecture:

The control architecture type should be hierarchical by functional decomposition. Generic interface signals should be used between the functions. By minimum effort the architecture should be reusable and allow new subsystems to be implemented.

Evaluation of the control architecture should be made by measuring the handled complexity, performance, reusability, and the sensitivity of communication- and computational delays.

4.2 Functional decomposition

In [12] a method for functional decomposition is given considering vehicle control systems. The highest functional level is denoted here as main control. Based upon [12] the following guidance is given:

1. The function needs to be at a level high enough to allow it to coordinate lower level functions that it has authority over.
2. The information, i.e. system status, can be observed by many and is allowed to flow in all directions; up, down, and across in the hierarchy.
3. The orders to actuators are only allowed to flow down to lower level functions. This upholds a causality of the orders within the hierarchical architecture.
4. If a particular function effects the vehicle's brand characteristics (can be observed by a customer) it is qualified to the highest level (main control) only if it does not jeopardise the reusability of the main control for different HEV configurations.
5. Durability is also a consideration for choosing the level at which partition a function. Local control of any potentially damaging functions is recommended.
6. The interfaces within the control system should be generic, i.e. not hardware dependent.

Item 4 allows manufacturer to retain ownership of the brand specific functions while suppliers can provide controls for various subsystem functions. This also allows the manufacturer to change the vehicle characteristics from optimizing the drivability to fuel economy. Item 5 also matches well with the supplier and manufacturer relationships. Item 6 allows hardware to be changed without redesigning the functional architecture.

4.3 Main architecture

The Main Control consist of three major parts; Driver Interpreter (DIp) interprets the driver's demands, Vehicle Motion Control (VMC) controls the vehicle according to these demands and Energy Management (EM) assures that this is done in a energy efficient way. Additionally there is the Strategic Control (SC) which summarizes the input from them both and makes the overall decisions considering reliability and safety. The functional decomposition with three hierarchical levels is shown in Figure 3. The highest of these levels is the Main Control. The communication is handled with a network.

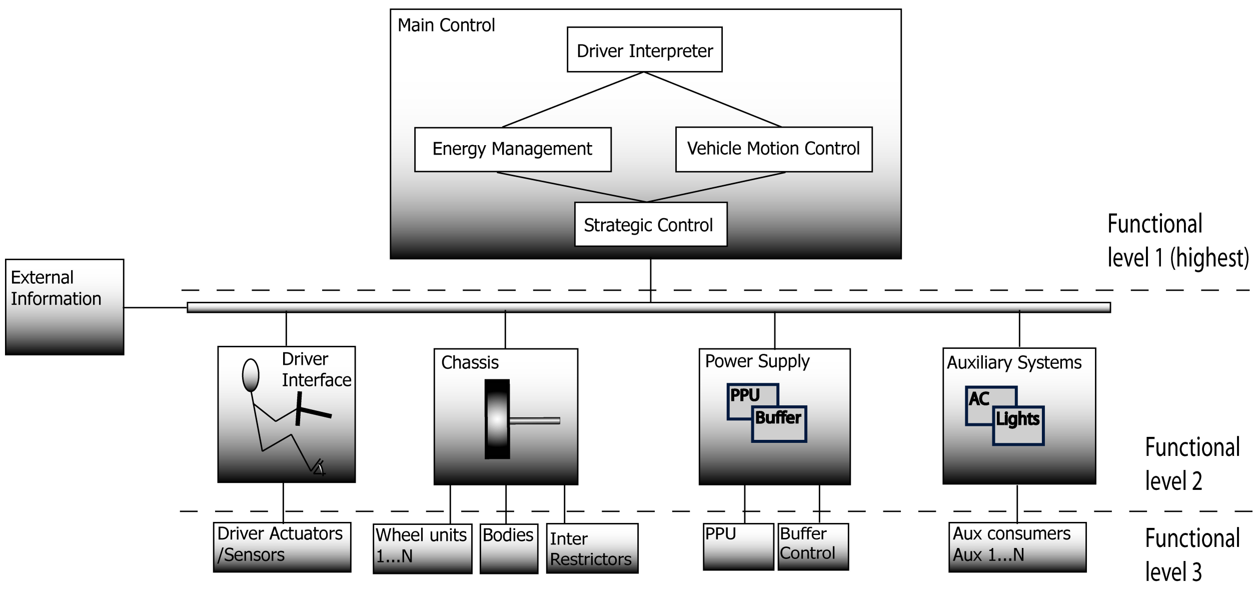


Figure 3: Schematic sketch of the functional architecture. It contains three levels.

In Figure 4 the signal flow from the Driver Interpreter to Chassis and Power Supply is shown in more detail while auxiliary systems and Driver Interface are excluded for simplicity. This illustrates how driver's intentions generate vehicle motion and the needed energy.

4.3.1 Driver Interpreter

Driver Interpreter handles the communication with the Driver Interface. The incoming signals are translated into a desired path and according to the limitations given by Energy Management and Vehicle Motion Control, feedback signals are sent to the Driver Interface.

4.3.2 Vehicle Motion Control

Vehicle Motion Control calculates the global forces Fx, Fy, and Mz that are required to generate the desired accelerations received from Driver Interpreter. Then it determines how the forces should be distributed between the Wheel Units (WU). More detailed description of the VMC and WU functions are found in [8]. The idea is to already from the beginning determine the force distribution between the wheel units and by this achieve overall performance with smooth behaviour that considers the maximum force surface (fx_i, fy_i, fz_i) for each Wheel Unit to generate desired forces within the stable region. Similar approach is also used in [13]. A conventional vehicle have different safety systems such as ABS, VSC (Vehicle Stability Control, TCS (Traction Control System), these functions are usually only used in critical situations, and thus don't have a smooth behaviour.

4.3.3 Energy Management

EM calculates the desired power needed from Power Supply considering the total tractive force and needed auxiliary power. EM calculates a State of Charge (SOC) target where it considers vehicle speed. A coefficient of desired electric regenerative braking is also calculated and sent to both Chassis and Power Supply. It considers if the SOC is higher or lower than SOC target. Logic for maximum auxiliary power use is also located in EM. The overall traction force is estimated by Energy Management and is sent to Strategic Control that finally sends the order to Power Supply.

4.3.4 Strategic Control

The SC is the part in Main Control that finally places the orders from EM and VMC. It considers if EM or VMC signals that the vehicle status is critical and then Strategic Control gives priority to primary functions as suggested from either part.

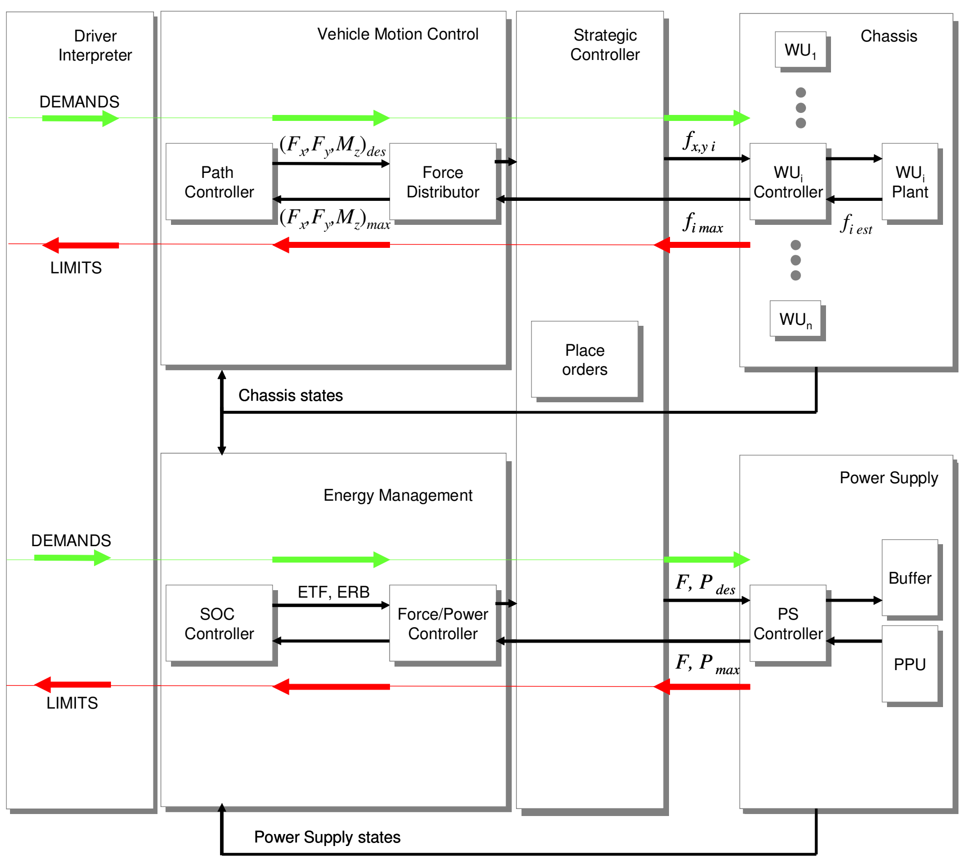


Figure 4: Signal flow between Driver Interpreter, Chassis and Power Supply. All demands have to be authorized by the Strategic controller.

Generic interfaces are utilised for the orders and the information e.g. vehicle states. By using generic interfaces, hardware can be changed without re-designing the Functional level 1. Some of the allocated functions in functional level 1 are summarized in Table 1. Table 2 shows a subset of functions for level 2. The signal interfaces between functional level 1 and 2 are made generic. There are specific subsystems within the different classes in functional level 2.

Table 1: Some of the allocated functions in Functional level 1 –Main Control.

Driver Interpreter	Definitions
Driver intentions	Determines the desired global accelerations by interpreting the information given by sensors in Driver Interface and the feedback from the Vehicle Motion Control, Driver Interpreter.
External information	If it is activated, it uses available external information, e.g. distance to vehicles ahead, traffic flow and road conditions to automate the driver's intentions. This includes functions as cruise control (keeping a desired speed)
Driver feedback	By limit feedback input from Vehicle Motion Control and Energy Management the level of feedback is determined and sent to Driver Interface. This could be force feedback on steering wheel and pedals.
Vehicle Motion Control	Definitions
Global forces	Determines the desired global vehicle forces from Driver Interpreter.
Wheel Unit forces	Determines the longitudinal, lateral, and vertical wheel forces for each wheel unit for vehicle dynamic optimal driving for current vehicle state and the desired global forces.
Limit feedback	Interprets and feeds back limitations on achievable accelerations to the Driver Interpreter.
Energy Management	Definitions
State of charge (SOC) target	By considering vehicle state (vehicle speed), driver's intentions, and environmental data (e.g. known topology, traffic information) a suitable SOC target for the buffer is determined.
Buffer	SOC regulation according to SOC target.
Traction force/power	Determines a traction force for energy optimal driving.
Split Traction force	If a parallel or split HEV configuration is the current system then the suggested level (0...1) should be generated by the electric motors.
Split Braking force	Defines the level (0...1) that should be used to regenerate energy.
Auxiliary load	Determines the maximum power allowed for the auxiliary system.
Strategic Control	Definitions
Arbitration of demands from EM and VMC	Summarizes the demands from Energy Management and Vehicle Motion Control and decides which is most critical.
Vehicle mode	Here, different vehicle characteristics are accounted for by driver's choice. The different modes could be sport, normal, or economical driving.
Authorisation of orders	Sends final orders to functional level 2.
Shut down logic	When bad state of health is sent from some actuator/sensor it is allowed to shut down by Strategic Control.

Table 2: Some of the allocated functions in Functional level 2 –Driver Interface, Chassis, Power Supply, Auxiliary systems.

Driver interface	Definitions
steering, accelerator, brake, mode	Determine the level and rate of change of the pedals and steering wheel or joystick. Sends the information to Driver Interpreter along with mode settings as e.g. sport/normal/economy.
Forward / Reverse	Determines the direction of the vehicle
Chassis	Definitions
Wheel unit control	The forces are distributed by the VMC is generated at each WU. Typically, the desired forces are translated into steering angle and tractive/braking torque.
Actuator coordination	Several actuators may perform the same function, this requires

	coordination. Typically this could be to split the requested tractive/braking force between current available actuators according to guidance given by Strategic Control.
Inter-restrictor coordination	When introducing inter-restrictors, one actuator may affect several wheel units, typically a rack steering which constrains the steering angle of two wheels.
Power Supply	Definitions
PPU	Control of the Primary Power Unit. For an ICE and transmission this would include elementary engine functions such as spark, air, fuel etc. plus shift scheduling for the transmission.
SOC/SOH level	Determines the SOC/SOH level of the buffer and send this information to Energy Management.
DC-DC	Determine the charging or discharging mode for DC to DC voltage converter.
Auxiliary Systems	Definitions
Climate control	Regulate the cabin temperature.
Lights	Regulate lights.

4.4 Power supply

The conventional powertrain concept with a combustion engine, transmission, and driveline is not a valid description for a HEV. The HEV concept includes handling of a major electricity source in combination with a conventional or parts of a conventional powertrain. A more suitable name of this function is Power Supply. The Power Supply includes both the Primary Power Unit (PPU) and a buffer and can be anything from an internal combustion engine to a fuel cell. The buffer can be an electric buffer such as a battery, super capacitor or a mechanical one e.g. flywheel. Figures 5 and 6 show how the power supply is defined for a serial, parallel, and split HEV configuration. The examples include inter-restrictors between Wheel Units. The restrictors illustrates that the driving torque applied to two wheels is restricted by e.g. a differential or an electric machine connected by a differential. Restrictors are explained in Section 4.5.3.

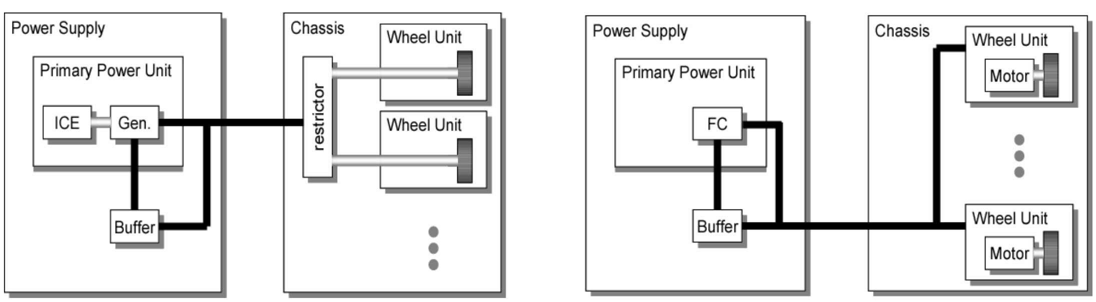


Figure 5: Illustration of power supply for serial with internal combustion engine (left) and serial with fuel cell (right) HEV configuration. ICE=Internal Combustion Engine, Gen=Generator, FC=Fuel Cell, Black line=electrical power, and Grey line=mechanical power.

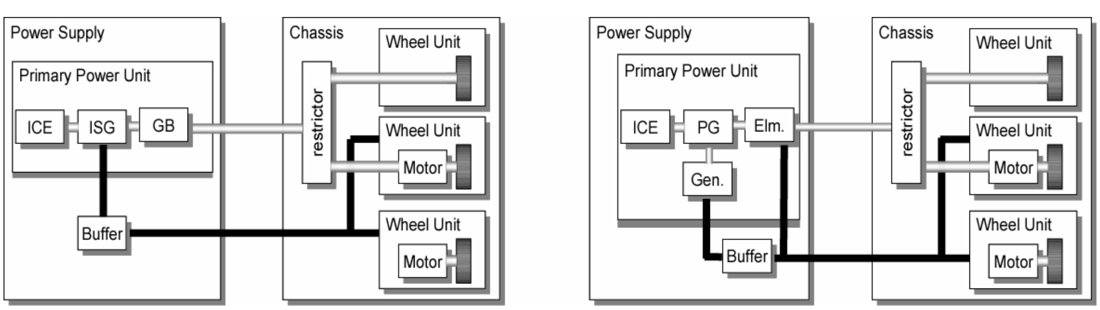


Figure 6: Illustration of power supply for parallel (left) and split (right) HEV configuration. ISG=Integrated Starter/Generator, GB=GearBox, PG=Planetary Gear, Elm=Electric machine.

4.5 Chassis

The chassis is thought of as a body onto which a number of wheel units are mounted, see Figure 7. Each wheel is then considered as an autonomous unit and by default decoupled from the other wheels. Depending on the linkage carrying the wheel as well as the available actuators, there are different possibilities to generate ground contact forces. A very simple case is a wheel with only brakes and no steering possibility and passive suspension, while other wheel units may have drive, steering, camber control and active damping.

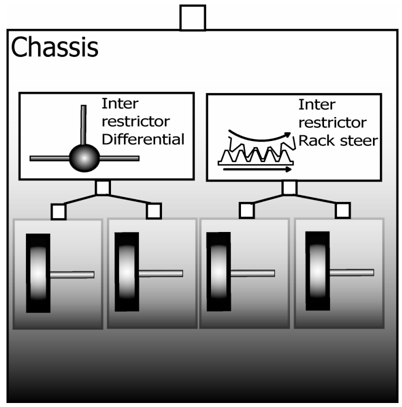


Figure 7: Schematic sketch of a chassis containing 4 wheel units and the inter-restrictor differential and rack steering.

There are various kinds of restrictions for each wheel's motion. An obvious case is the steering of a traditional car which couples the steering angle of the front wheels. To handle this in a clean and efficient way, the chassis is thought of as consisting of three types of components at any amount each; bodies, wheel units and restrictors.

4.5.1 Body

The body's main task is to frame the vehicle which essentially means to carry the wheel units as well as passenger and goods. In addition to this, the body also carries properties such as mass, inertia, and a geometric reference frame as well as sensors to measure its states. The main idea with the function body is that more than one body can be used when defining articulated busses, semi trailer combinations or week chassis. The body includes the specific wheel units that are attached to the specific body inter restrictors define the connections between the bodies.

4.5.2 Wheel Unit

The distributed forces from the Strategic control is realised at each WU that also sends information about maximum achievable force. To generate the tractive force, fx_i, the wheel unit checks how much rotational torque is available directly by Power Supply on the actual wheel unit and then coordinates the available actuators to meet the desired order. Typically the wheel unit could be realized as in [7]. More details about vehicle motion control and wheel unit are presented in [8].

4.5.3 Restrictors

Today's conventional chassis have constraints and limit the controllability of each wheel unit. To handle this in a systematic way restrictors are introduced. A restrictor can either restrict the wheel's motion relative to the body, i.e. within the wheel unit or relative to another wheel unit; these are referred to as intra-acting and inter-acting, respectively. Furthermore, these could be either 'active', meaning that they could be controlled, or 'passive' units like e.g. a standard strut. Some examples are presented in Table 3.

Table 3: Example of different type of utilised restrictors. Note that parts of a traditional powertrain that are used to distribute tractive force are as restrictors within the chassis.

	Inter-restrictor	Intra-restrictor
Active	Limited slip differential Rack steering	Wheel motor Mechanical brake unit
Passive	Differential Anti-roll	Suspension linkage Strut

5 Modelling of Hybrid Electric Sport Utility Vehicle

An object oriented modelling language is used to test how the control architecture works [1]. Two models of a Hybrid Electrics Sports Utility Vehicle are modelled. The first configuration uses a combustion engine with Integrated Starter Generator (ISG), automated manual transmission, battery buffer and electric motors at the rear wheels. This concept allows a more economical utilisation of the four wheel drive and a similar concept study was made in [14]. The second example is a future version with serial HEV configuration with a fuel cell, buffer, and autonomous Wheel Units. In Figure 8 (left) the total vehicle model is shown and in the right screen shot shows how different Power Supplies can be used due to the modularity in the architecture.

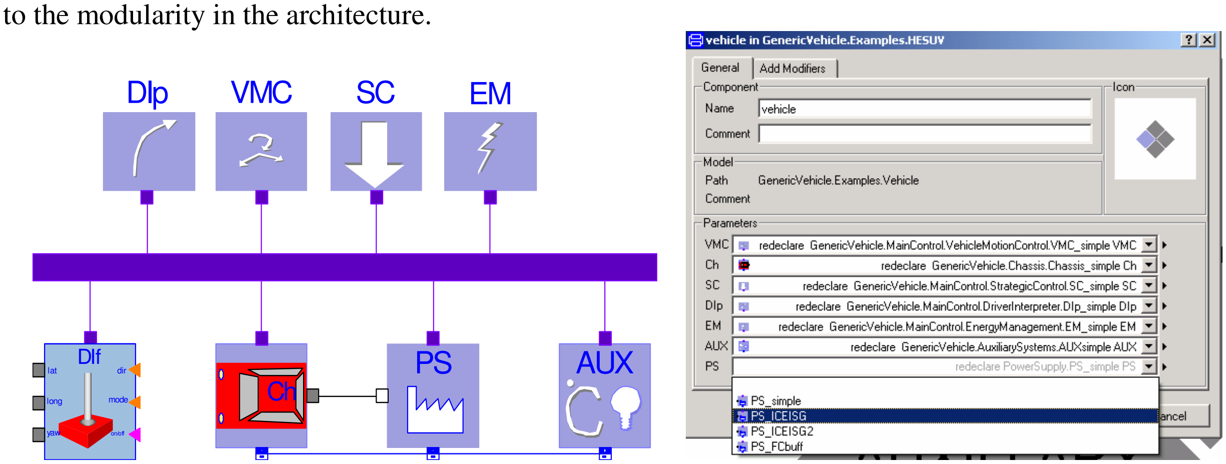


Figure 8: Total vehicle model (left). Due to the generic architecture, the configuration can be changed by selecting options from drop down boxes and no remodelling is necessary (right).

In Figure 9 the two chassis configurations are shown. The first configuration has rotational power (dotted, black) is distributed to the front wheels via the differential. In the front there is also a rack steering to constrain the wheels' steering angle and both front and rear, there are anti roll linkages. The dashed, purple line shows the bus connection and the solid, blue lines are mechanical connections. In the second configuration only electric power is used and no inter-restrictors are used since all wheel units are independent. Roll control is managed by active components in the suspensions.

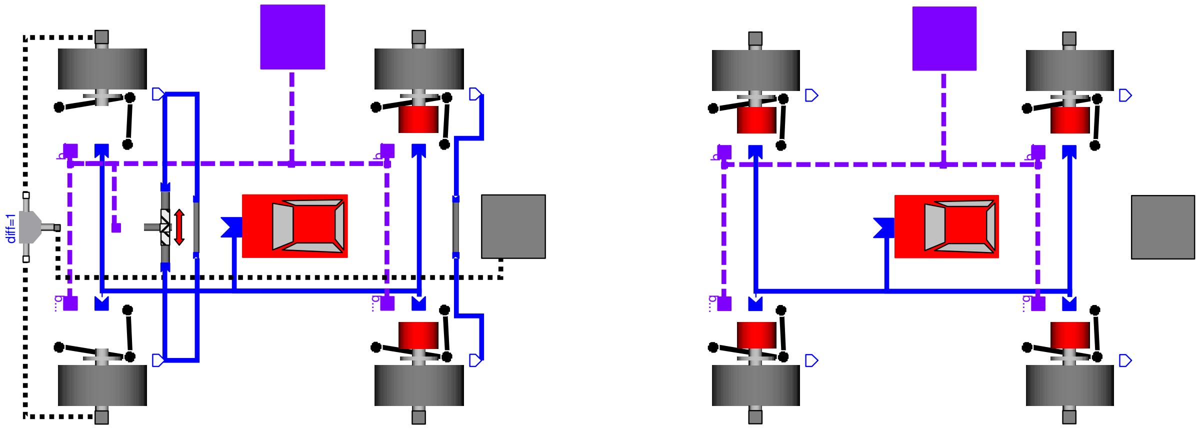


Figure 9: Chassis models corresponding to the first configuration (left) and the second configuration (right).

The two different Power Supply configurations that were implemented could be changed without affecting the rest in the generic control architecture. The chassis configuration could be changed, but further work on handling the inter-restrictors in an efficient will be made.

6 Conclusions and future work

Here a methodology and a definition for generic control architecture for HEVs are given. Hierarchical partitioning and functional decomposition is utilised to place the functions in different functional levels. The highest functional level includes the functions *Driver Interpreter, Energy Management*, *Vehicle Motion Control,* and *Strategic Control.* The second functional level includes the sub functions *Driver Interface, Chassis, Power Supply, External Information* and *Auxiliary Systems.* The generic control architecture has been implemented in object oriented modelling language and is proven to work.

In this paper, the Wheel Units are seen as a function to apply forces to the ground and by default are decoupled. But today's cars have constraints between the wheels. This is suggested to be handled by defining inter-restrictors. These will be further studied in future work along with other vehicle configurations.

7 References

[1] www.modelica.org, The Modelica Association, (2002)

[2] Erik Coelingh, Pascal Chaumette, and Mats Andersson. *Open-Interface Definitions for Automotive Systems, Application to a Brake by Wire System,* SAE 2002-01-0267.

[3] Nico A. Kelling and Patrick Leteinturier, *X-by-Wire: Opportunities, Challenges and Trends,* SAE 2003-01-0113

[4] *X-By-Wire, Safety Related Fault Tolerant Systems in Vehicles,* Project No. BE 95/1329, Contract No: BRPR-CT95-0032, X-By-Wire Consortium, 1998.

[5] Michael Larsen, Steve De La Salle and Dave Reuter, *A Reusable Control System Architecture for Hybrid Powertrains*, SAE 2002-01-2808.

[6] Torsten Bertram, Rainer Bitzer, Rainer Mayer and Asmus Volkart, *CARTRONIC - An Open Architecture for Networking the Control Systems of an Automobile*, SAE-980200.

[7] Sigvard Zetterström, *Electromechanical Steering, Suspension, Drive and Brake Modules,* VTC 2002, Vancouver, Canada, September 24 - 28, 2002.

[8] Johan Andreasson and Leo Laine, *Driving Dynamics for Hybrid Electric Vehicles Considering Handling and Control Architecture,* To be presented at IAVSD'03, 18th Int. Symp. Dynamics of Vehicles on Roads and Tracks, Japan, 24-29 Augusti 2003.

[9] Joseph C. Burba, *Consideration of Control System Architecture in Electric and Hybrid Electric Vehicles,* Electric Vehicle Symposium 18, Berlin, 2001

[10] Magnus Gäfvert, *Topics in Modeling, Control, and Implementation in Automotive Systems,* Dep. of Automatic Control, Lund Institute of Technology, Sweden, 2003.

[11] John Yook, Dawn Tilbury, Kalyani Chervela, and Nandit Soparkar. *Decentralized, Modular Real-Time Control for Machining Application.* url: citeseer.nj.nec.com/246337.html, 1997.

[12] Anthony M. Phillips. *Functional Decomposition in a Vehicle Control Systems* American Control Conference, Anchorage 2002.

[13] Yoshikazu Hattori, Ken Koibuchiyy, and Tatsuaki Yokoyama, *Force and Moment Control with Nonlinear Optimum Distribution for Vehicle Dynamics*, AVEC'02, Japan, 2002.

[14] Mikael Arnelind and Gustav Widén, *Design and Control of an E4WD system for a soft hybrid electric vehicle*, Chalmers University of Technology, Sweden, June 2003.

8 Authors

Leo Laine, Chalmers University of Technology, Sweden. Received his M.Sc. in Structural Mechanics in 1996 from Chalmers. Worked at Swedish Defence Research Establishment as research engineer until 1998. Then as a numerical consultant at ANKER – ZEMER Engineering AS. In 2002 he started his Ph.D. studies at Chalmers, Dep. of Machine and Vehicle Systems, Div. of Mechatronics. The name of the project is HEV Main Control.

Johan Andreasson, Royal Institute of Technology (KTH), Sweden. Received his M.Sc. in Mechatronics in 2001 from KTH. In 2002 he started his Ph.D. studies at KTH, Div. of Vehicle Dynamics. The name of the project is HEV Driving Dynamics.

Paper II

Evaluation of a Generic Vehicle Control Architecture

in

Proceedings of the 30^{th} *Congress of the International Federation of Automotive Engineering Societies, (FISITA'30)*, Barcelona, Spain, May 2004.

F2004F360

EVALUATION OF A GENERIC VEHICLE MOTION CONTROL ARCHITECTURE

[1]Andreasson, Johan*, [2]Laine, Leo, [2] Fredriksson, Jonas
[1] KTH Vehicle Dynamics, Sweden [2]Chalmers University of Technology, Sweden

KEYWORDS: Generic Vehicle Motion Control, Force Distribution, Restrictor, Control Architecture, Chassis Configuration

ABSTRACT

In the last decades, more and more control algorithms are added to vehicles in order to improve their behaviour, especially under critical situations. It is also likely that future vehicles will carry an increasing amount of electricity which opens up for more actuators that together with x-by-wire technology result in a wider span of possible chassis configurations than seen today. In this paper, a generic vehicle motion control architecture is suggested and evaluated on three various chassis configurations. By introducing a concept called restrictors, the architecture is able to handle vehicles of today with mechanical front wheel steer and two wheel drive to future scenarios where each wheel can be steered and driven individually.

1 INTRODUCTION

A possible scenario for future vehicles is that they will have more onboard electricity which allows hydraulic and mechanic actuators to be electrified. An example is wheel motors that could be used both as a driving torque actuator as well as a brake during regeneration of electricity. The wheel motor itself would not be able to handle the peak brake forces in all situations and thus additional brakes will also be required. An additional possibility is that every wheel can by itself change the steering and camber angle as an autonomous wheel corner [1]. Even today's chassis use several sensors and actuators to increase the controllability of the wheels.

A traditional approach has been to derive the controllers based on the actual chassis configurations and the specific task. This leads to a lot of work when actuators or sensors are added, removed or replaced. With the traditional approach it is also difficult to evaluate the benefit of a certain actuator, i.e. the vehicle characteristics is derived from the chassis configuration and not vice versa.

In [2] a Vehicle Dynamics Management (VDM) is suggested that integrates systems like Anti-lock Braking System, Traction Control System and Vehicle Stability Control into one nonlinear optimisation task. The VDM uses a hierarchical algorithm to distribute desired forces on different wheels but the method only handles traction and braking. It is furthermore assumed that each wheel is only limited by the saturation of tyre forces, i.e. all wheels must have both unlimited traction and braking. To reach better performance in the future it is necessary to use combinations for steering, drive and suspension control, as stated in [3].

In this paper, it is shown how the wide span of possible chassis configurations can be handled with the suggested vehicle motion control architecture [4, 5], without changing the control algorithms.

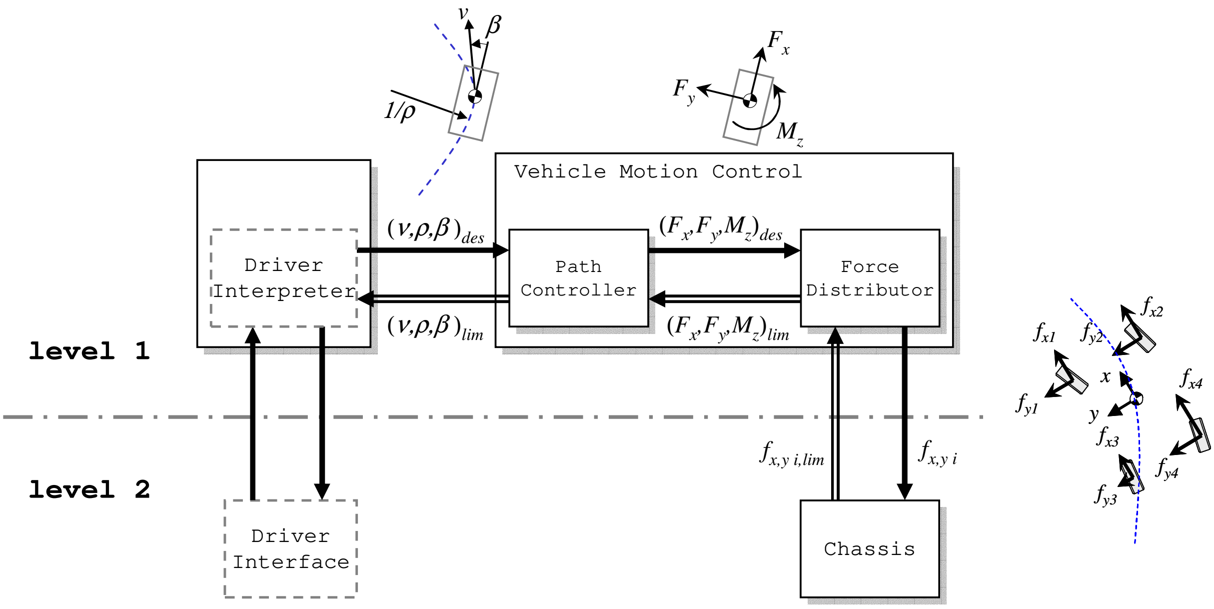


Figure 1: Desired signals from Driver Interface to Chassis. Dashed boxes are not considered in this paper.

2 GENERIC VEHICLE MOTION CONTROL ARCHITECTURE

Consider figure 1, the driver interacts with the vehicle via the Driver Interface and the signals are transformed into a desired path by Driver Interpreter. This path is then followed by the Vehicle Motion Control that sends desired forces to the chassis[1]. In critical situations, when either tyre forces, brakes or other actuators reach their limits, these limitations must be taken into account in the force distribution and in the communication with the driver.

Thus, the Vehicle Motion Control (VMC) at level 1 has three main tasks, 1) control to make the vehicle follow the intended path, 2) force distribution to each available wheel unit and 3) deriving limitations for the driver interpreter.

2.1 Path control

As illustrated in figure 1, the path is defined as speed v, curvature ρ, and side slip β. The advantage with this representation is that the path is defined also for standing still which make maneouvres such as parking possible.

Based on information of desired and estimated path, desired vehicle forces F_x, F_y, M_z are calculated. The system is nonlinear with three inputs and three outputs and there are various ways to solve such problems. In the following tests, a controller based on feedback linearisation is used. The controller includes integral action in order to remove tracking errors. It also includes an anti-windup strategy to handle the case when the control signals are limited or saturated. The control design is described more thoroughly in [6].

2.2 Wheel force distribution

The vehicle forces derived by the path controller must be distributed to each wheel unit considering the limitations due to tyre grip and actuator limitations. Thus, the problem can be described as an

[1] How these forces are transformed into wheel spin and and steering angles are described in [4].

optimisation problem, with nonlinear boundaries.

$$\begin{aligned} &\min\ s^T \mathbf{W} s \\ &s = \mathbf{B}u - v \\ &c(u) < 0 \\ &d(u) = 0 \end{aligned} \tag{1}$$

where v is the desired vehicle forces F_x, F_y, M_z, u is the desired wheel unit forces $f_{x,i}$, $f_{y,i}$, $\mathbf{B}$ is a transformation matrix and $\mathbf{W}$ is a weight matrix. The boundaries consist of inequalities $c(u)$ and possibly also equalities $d(u)$ as explained later on. This is similar to control allocation for aircrafts, see e.g. [7] for a good overview. However, while aircrafts normally have to deal with componentwise rudder deflection limitations, vehicles equipped with tyres instead have nonlinear and coupled constraints due to the tyre's friction ellipse. More information about how this is handled can be found in [6].

2.3 Limitations

For the Driver Interpreter to be able to give the driver proper feedback, the current limitations of the vehicle is given by the VMC as indicated in figure 1. This communication relates to the path limits $(v, \rho, \beta)_{lim}$ and thus, the force limitations at each wheel unit must first be transformed into a resulting force/moment potential and then transformed into the proper limits.

The transformation of the wheel unit limits is done as an approximation where the force potential $(F_x, F_y)_{lim}$ is a sum of each wheel unit's limit. The yaw moment potential $(M_z)_{lim}$ is then a function of the unused force potential, giving $f(F_x, F_y, M_z) < 0$.

The resulting path limitations are then calculated based on the following relationship between force and motion

$$\begin{pmatrix} F_x \\ F_y \\ M_z \end{pmatrix}_{lim} = \mathbf{MT}(\beta) \begin{pmatrix} a_x \\ a_y \\ z_z \end{pmatrix}^{path}_{lim} = \mathbf{MT}(\beta) \begin{pmatrix} \dot{v} \\ v^2\rho \\ \ddot{\beta} + \dot{\rho}v + \dot{v}\rho \end{pmatrix}_{lim} \approx \mathbf{MT}(\beta) \begin{pmatrix} \dot{v} \\ v^2\rho \\ \ddot{\beta} \end{pmatrix}_{lim} \tag{2}$$

where $\mathbf{M}$ is the mass properties, $\mathbf{T}(\beta)$ transforms the limits to fit the actual heading direction of the vehicle and ^{path} indicates variables resolved in the path frame, giving

$$\begin{pmatrix} \dot{v} \\ \rho \\ \ddot{\beta} \end{pmatrix}_{lim} \approx \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1/v^2 & 0 \\ 0 & 0 & 1 \end{pmatrix} \mathbf{T}^{-1}(\beta)\mathbf{M}^{-1} \begin{pmatrix} F_x \\ F_y \\ M_z \end{pmatrix}_{lim} \tag{3}$$

3 CHASSIS

The possibility to generate tyre forces is strongly coupled to the actual chassis configuration. Still, the optimisation problem in equation 1 remains the same, only the boundaries are affected. To derive the boundaries, the chassis is described in a modular way by the use of a set of components: *Wheel Units* (WU), *bodies* and *restrictors*. These components defines the chassis and its limitations as described in figure 2. Level 2 is the combination of components that corresponds to the actual chassis layout. This combination is then used to set up the boundaries for the optimisation problem in VMC, level 1. The levels in figure 2 are related to the levels in figure 1.

A future vehicle chassis with autonomous wheel corners is the most straightforward. In that chassis configuration each wheel unit's limitations can be described according to section 3.1. In the general case, some or all wheel units have limited controllability due to actuator and/or are coupled with other wheel units. To deal with this, the restrictor concept is introduced, section 3.2. In this paper, the discussion is limited to single-body chassis like cars and most buses. The use of more bodies makes it possible to define e.g. tractor-trailer combinations and articulated buses.

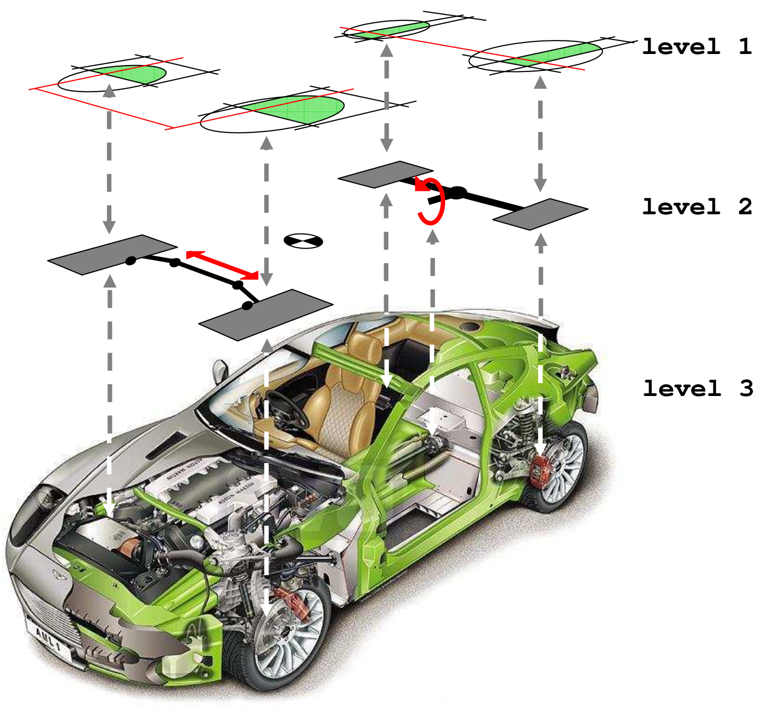


Figure 2: A vehicle (level 3) and the abstraction levels corresponding to the chassis configuration model with four wheel units, one body and 2 restrictors (level 2) and the constrained optimisation problem in Vehicle Motion Control (level 1). Car picture from [8].

3.1 Wheel force limitation concept

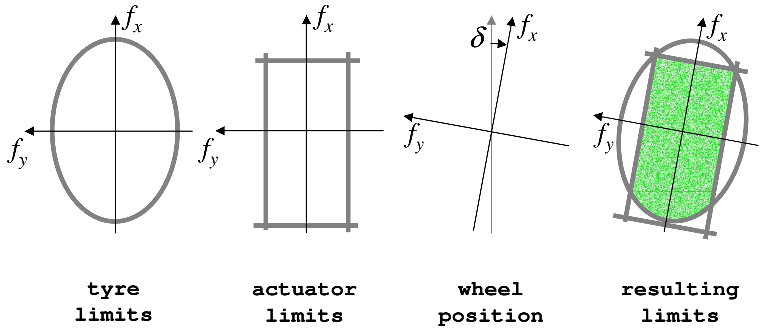


Figure 3: Resulting limits in f_x and f_y for a wheel unit.

The wheel unit limits are dependent on the tyre's friction ellipse in combination with the available actuators as illustrated in figure 3. Normal force, road condition and tyre characteristics define the tyre limits. Steering and camber define the lateral actuator limits and brakes and motors define the longitudinal actuator limits. These together define the resulting inequality boundaries for a wheel unit, $c(u) < 0$.

The actuator limits often are the result of more than one actuator affecting the same limit. This is illustrated in figure 4 where a wheel unit is equipped with brakes and also is driven by an electric machine, giving the resulting limits as a sum.

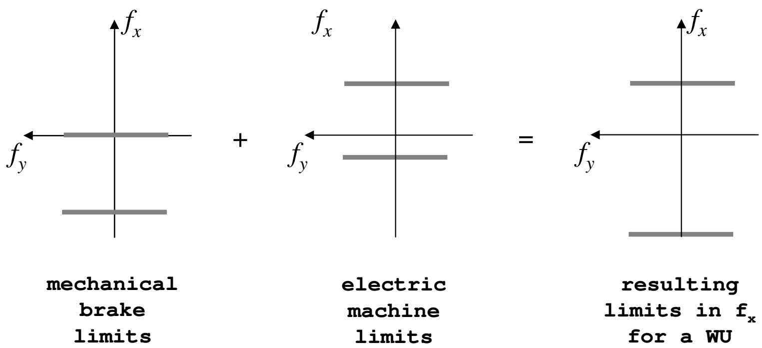


Figure 4: Resulting limits in f_x for a wheel unit with mechanical brake and an electric machine.

3.2 Restrictors

As described earlier, limitations occur due to the chassis and its configuration, typical examples are steering and braking possibilities. These and other limitations that only affect one wheel are defined as *intra-restrictors*. Examples of these and how they affect the boundary problem are listed in the upper part of table 1. A common intra-restrictor is No Steering, resulting in no possibility to affect the generated lateral force. Thus, the difference between possible and actual lateral force, Δf_y, is set to zero[2]. In the longitudinal direction, difference can be made between wheel units that are able to drive and brake and those wheel units that can only brake. The first case generates limits $f_x^{min} < f_x < f_x^{max}$ while the latter further restricts $f_x^{min} < f_x \leq 0$.

Other limitations occur due to coupling effects between wheel units, called *inter-restrictors*. This could e.g. be a mechanism that couples the steering angle of two wheels which is the standard way to steer a vehicle today. The resulting coupling of these forces are not obvious but a first estimate is that the ratio between achieved and maximum lateral force for both wheel units should be equal. This yields an additional equality constraint equation, $d(u) = 0$. If the steering rack is directly controlled by the driver, Rack Steer, the effect on the limitations is identical to a No Steer intra-restrictor. These and other cases are described in the lower part of table 1.

When inter- and intra-restrictors act together, the limits are affected as described in figure 4. However, inter-restrictors that contain constraints $d(u) = 0$ have a slightly different effect on the limits. Consider for example a two-wheel driven car with a simple differential and individual brakes. The differential restrictor states that the longitudinal forces at each wheel should be equal $f_{x_i} = f_{x_j}$ and sum up to the input force $f_{x_i} + f_{x_j} = f_{x_k}$. At the same time the brakes allow individual negative force to be applied $f_{x_{i,j}}^{min} < f_{x_{i,j}} < 0$. Then, the forces must be within the sum of the limits from the input and the brake $f_{x_i}^{min} + f_{x_j}^{min} + f_{x_k}^{min} < f_{x_i} + f_{x_j} < f_{x_k}^{max}$ and at the same time the difference between the forces must be within the span of the brakes $f_{x_i}^{min} < f_{x_i} - f_{x_j} < -f_{x_j}^{min}$. In general, it is unwise to allow $f_{x_i} \neq f_{x_j}$ under normal driving conditions since it leads to break wear and increased fuel consumption. More information about how this is handled can be found in [9].

4 EVALUATION EXAMPLES

To evaluate the architecture, two different chassis configurations are tested. The first is a future scenario with four wheel units and no inter-restrictors, which leads to wide boundaries for the optimisation problem as described in figure 5, top (1). This vehicle could typically be realised as a series electric hybrid or a pure electric. The second chassis configuration is a rear wheel driven standard car with rack steer. This configuration lacks steering possibilities on the rear wheels and these thus have narrow areas in the lateral diretion. Additionally, positive longitudinal forces can only be applied at the rear wheels. Two modifications of this configuration is evaluated; with steer-by-wire (2a) as shown in figure 5, middle, and mechanical coupling to the steering wheel (2b), see figure 5, bottom. For chassis 2a, the front wheels can be steered by the VMC but are connected which constrain the lateral forces. The latter configuration leaves only the brakes and traction to be controlled by the VMC.

4.1 Evaluation models

The vehicles are modelled in Matlab/Simulink as a body with 3 degrees of freedom for longitudinal, lateral and yaw motion. In addition, vertical load on the tyres are dependent on the vehicle's accelerations. The tyre model is a simple Magic Formula [10] implementation where the scale factor D is dependent on nominal tyre force $f_{x,y}^{nom}$, road surface condition μ and vertical load f_z such that $D = f^{nom}$ at $\mu = 1$ and

[2]This can also be defined as a small interval to fit the optimisation algorithm.

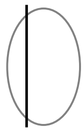
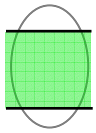
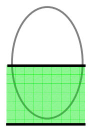
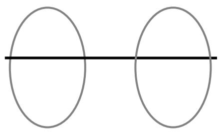
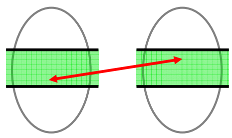
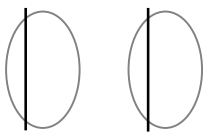
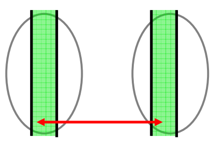
	intra-restrictors	
No Steer: Wheel Unit (WU_i) with suspension that does not allow the wheel to be steered.		$\Delta f_{y_i} = 0$
Drive and Brake: WU_i that is able to apply both driving and braking torque, typically equipped with a wheel motor.		$f_{x_i}^{min} < f_{x_i} < f_{x_i}^{max}$
Brake: WU_i that is able to apply only braking torque, typically equipped with a disc brake. Special case of **Drive and Brake**.		$f_{x_i}^{min} < f_{x_i} < 0$
	inter-restrictors	
Differential: Distributes the force from input k to WU_i and WU_j equally.		$f_{x_i} + f_{x_j} = f_{x_k}$ $f_{x_i} = f_{x_j}$
Actuated Differential: Distributes the force from input k to WU_i and WU_j with a maximum force difference Δf.		$f_{x_i} + f_{x_j} = f_{x_k}$ $\lvert f_{x_i} - f_{x_j}\rvert < \Delta f$ $f_{x_i} f_{x_j} > 0$
Rack Steer: Mechanism that couples the steering angles of WU_i and WU_j. The angle can not be controlled by the VMC.		$\Delta f_{y_i} = \Delta f_{y_j} = 0$
Actuated Rack Steer: Mechanism that couples the steering angles of WU_i and WU_j. The angle can be controlled by the VMC.		$\frac{\Delta f_{y_i}}{f_{y_i}^{max}} = \frac{\Delta f_{y_j}}{f_{y_j}^{max}}$

Table 1: Some restrictors and how they affect the force distribution problem. The equations should be considered as estimates of the effects of the restrictors and are defined as simple as possible.

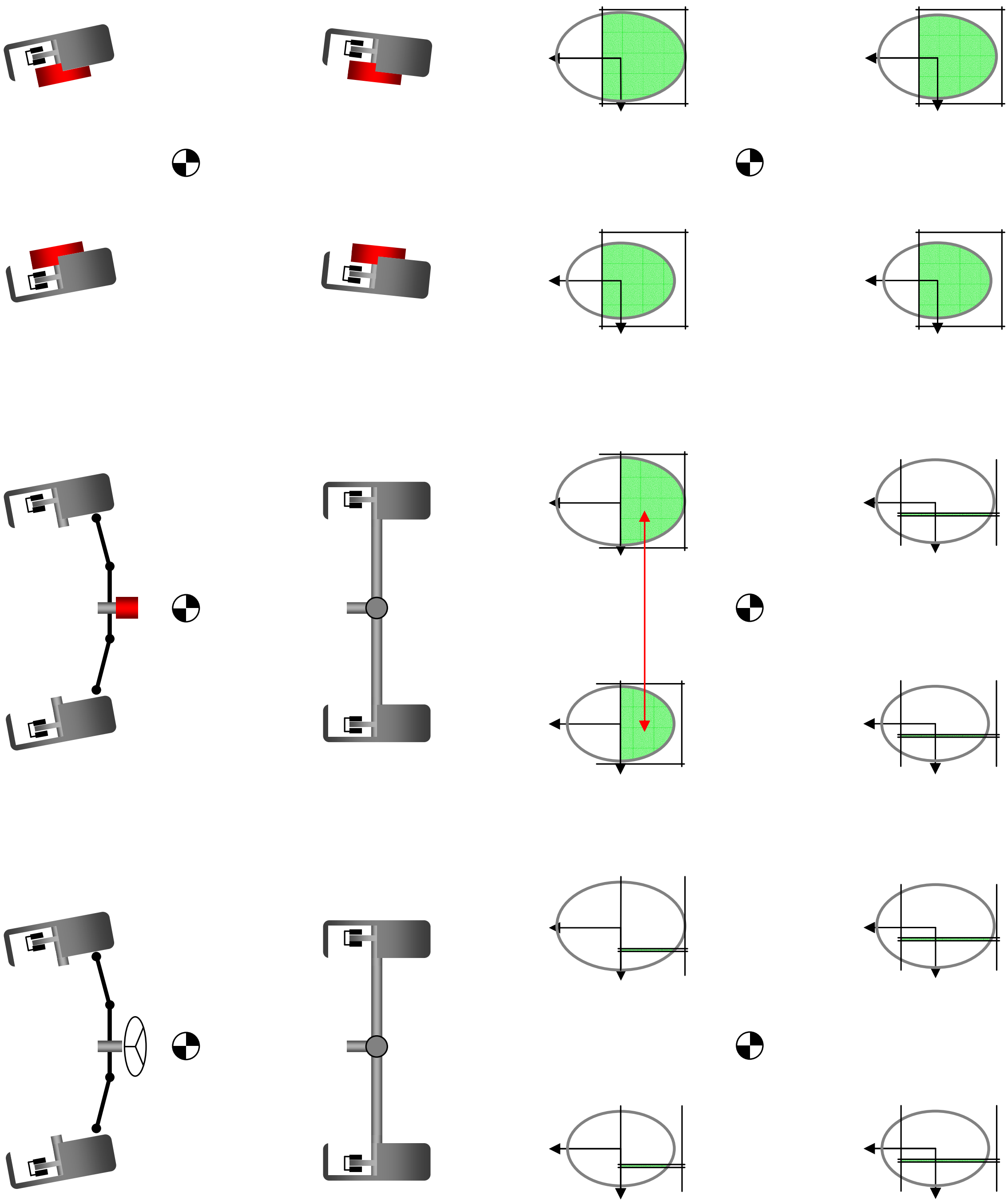


Figure 5: Chassis configurations used in the evaluations, the valid ranges for the distributed forces are indicated with shaded areas. Top: (1) with no inter-restrictors and wheel units that can be both steered, driven and braked by VMC leads to wide boundaries for the optimisation problem. Middle: (2a) with drive-by-wire rack steer and rear wheel drive. The front wheels can be steered by VMC but are connected which constrain the lateral forces in front. Bottom: (2b) with mechanical rack steer and rear wheel drive. This configuration leaves only the brakes and traction to be controlled by the VMC, giving narrow boundaries.

vehicle data	
total mass, m [kg]	1600
yaw inertia I_{zz} [kgm^2]	1800
height of COG over ground, h [m]	0.5
length of COG to front axle, a [m]	1.1
length of cog to rear axle, b [m]	1.5
track width, T [m]	1.7
tyre data	
nominal tyre force, f_x^{nom} [N]	4000
nominal tyre force, f_y^{nom} [N]	3500
nominal tyre load, f_z^{nom} [N]	4000
Magic Formula parameters, (B, C, E) [-]	(4, 1.6,-10)
actuator limits	
mechanical brake ($f_{x_i}^{min}$, $f_{x_i}^{max}$) [N]	(-5000, 0)
electric machine ($f_{x_i}^{min}$, $f_{x_i}^{max}$) [N]	(-1500, 1500)
differential ($f_{x_k}^{min}$, $f_{x_k}^{max}$) [N]	(0, 6000)

Table 2: Used data in the examples. Electric machine refer to chassis configuration 1 and differential to chassis configurations 2a and 2b.

$f_z = f_z^{nom}$.

$$D = \mu f_{x,y}^{nom} f(f_z)$$
$$f(f_z) = \frac{1}{5.7272}\left[-4.2569 + 8(\frac{f_z}{f_z^{nom}} + 0.5706) - (\frac{f_z}{f_z^{nom}} + 0.5706)^{2.1}\right] \quad (4)$$

The data for the used vehicle model is defined in table 2 along with the actuator limitations. The actuators are considered as ideal and the limitations are independent of vehicle speed. The limitations of the electric machines and the available force from the differential are set so that the chassis configurations 1 and 2 will have the same total tractive force potential. In the chassis configuration with differential the limits are given as a sum of limits for connected wheel units, as shown in table 1.

4.2 Evaluation tests

Two maneouvres were performed to evaluate the suggested architecture, braking from 20m/s to stand still (brake) and step change from straight driving $\rho = 0\text{m}^{-1}$ to turning $\rho = 0.01\text{m}^{-1}$, corresponding to 4m/s^2 at $v = 20$m/s (steer). Each of these is performed under good conditions (high-μ), $\mu = 1$, as well as for a split-μ surface with $\mu_{right} = 1$ and $\mu_{left} = 0.1$ occurring at time=4s.

The feedback linearising controller handles the demands from the Driver Interpreter and equation 1 is solved for every time-step to distribute the forces. The optimisation problem is solved using the optimisation toolbox in Matlab6.5. The weighting matrix, $\mathbf{W}$, is chosen such that errors in F_y and M_z are weighted higher than F_x. In the wheel units, only feed-forward is used to derive the wheel spin and steering angles. It is also assumed that the steering angles are small and the rotation of limits as described in figure 3 are not considered. The same constant settings are used for all chassis configurations to investigate the transparency of this method.

In figure 6 a turn on split-μ surface is presented for test chassis configuration 1. The leftmost plots show how the vehicle manage to follow the desired motion and to the right the vehicle forces (F_x, F_y, M_z) are shown. In figure 7 the distributed wheel forces are shown, the sudden step at 4s is due to the change in μ and thus redistribution is done from the left with low μ to the right wheels.

In figures 8 to 10, the steady stated force distributions are shown for all test chassis configurations and maneouvres. Each figure shows the brake maneouvre, top, and the turn maneouvre, bottom. To the left,

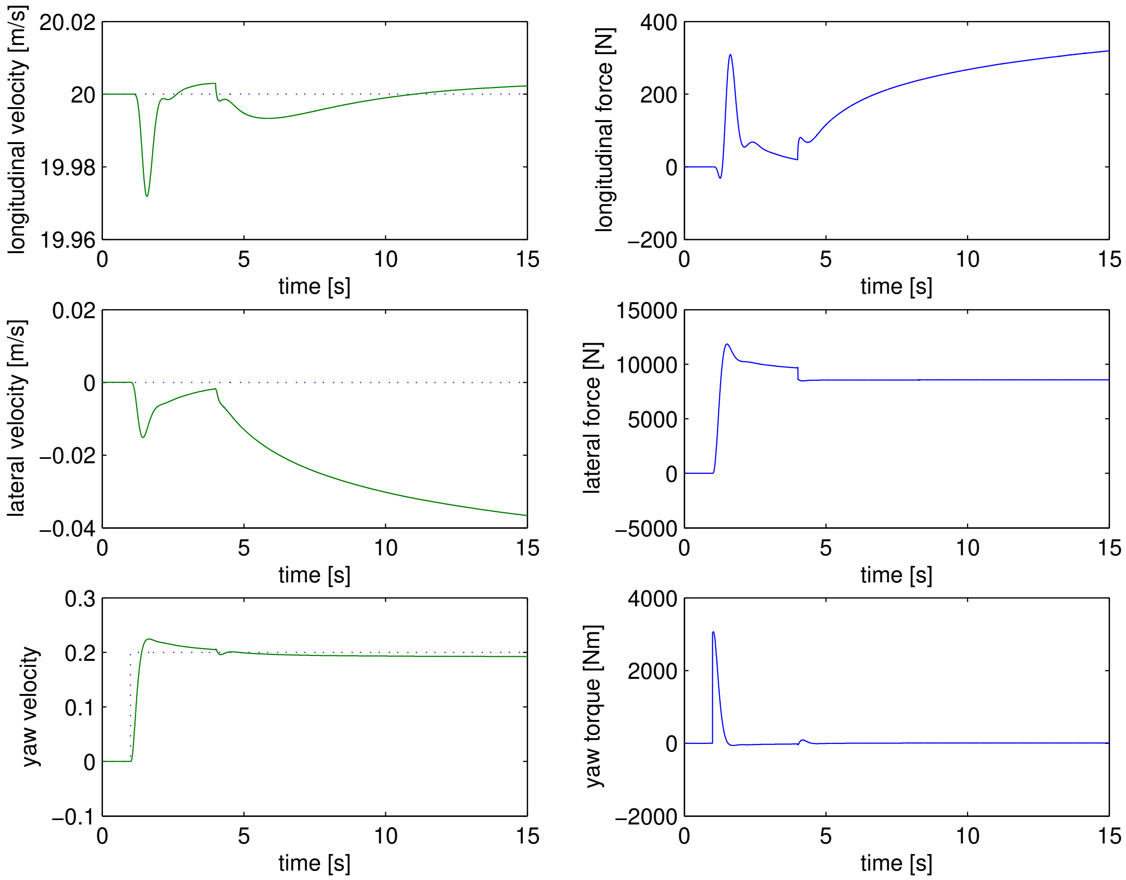


Figure 6: Desired and estimated motion along with vehicle forces (F_x, F_y, M_z).

the force distribution on high-μ is shown and to the right, the split-μ is shown.

From these tests, it can be seen that chassis configuration 1 performs as expected in all situations, figure 8. In the brake maneouvre, the forces are redistributed to the right wheels and to compensate for the yaw torque, the wheels counter-steer to keep the vehicle in the right direction.

Chassis configuration 2a shows some interesting aspects, figure 9. When braking at high-μ the forces are distributed unsymmetrically but still in a way that gives no yaw torque. For the optimisation, this solution is just as good as any. Consider also the high-μ steer maneouvre. Here, the longitudinal forces on the right side of the car is opposite which makes no sense in reality, it occurs since it is not specified in the cost function that it is a bad solution. How this is dealt with is explained in [9]. A third aspect is the fact that this configuration performs less good than expected during the split-μ tests. It was expected that, since the VMC could control the steering angle at the front, it would counter steer as for chassis configuration 1. However, this is the result of that the current optimisation cannot handle the effect of the lateral force at the front affecting the vehicle slip angle and thus indirectly the lateral force at the rear wheels. Lowering the cost for error in slide slip increases the amount of counter steer and makes the vehicle come to a stop quicker.

For chassis configuration 2b, only the brakes and traction can be controlled by the VMC and thus the ability to affect the lateral force is small. Steering on split-μ still works reasonable since the load distribution helps to build up higher wheel forces at the outer wheel. The brake performance is though poor since no means are available to compensated for the yaw torque that would come with a higher deceleration.

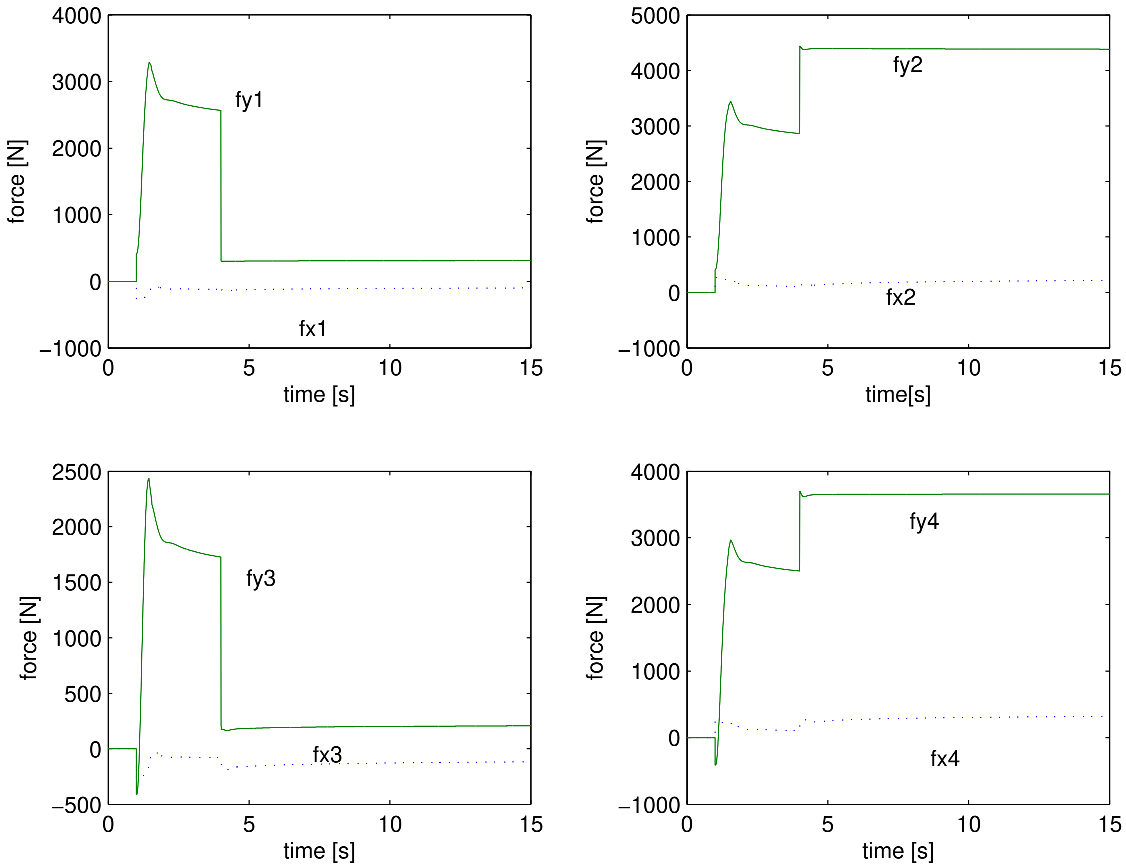


Figure 7: Force distribution for turn on split-μ with configuration 1.

5 CONCLUSIONS AND FUTURE WORK

It is shown how one vehicle motion controller can handle three different chassis configuration without modification. This is possible due to different levels of abstraction, an intermediate level 2 that describes the chassis as a combination of bodies, wheel units and restrictors and uses this to derive boundaries that in level 1 are used to control the vehicle motion.

The vehicle motion characteristics is thus only coupled to 1) the interpretation of the driver's intention, 2) the control of the vehicle forces, and 3) the cost function for the optimisation problem. This gives advantages when it comes to consolidating the brand specific qualities but it also adds a method to choose chassis configuration depending on the desired vehicle behaviour.

It is also seen that a general optimisation algorithm require a lot of computational effort and it might thus be advantageous to derive a more specific force distribution algorithm. Furthermore, it needs to be investigated how actuator and tyre dynamics affect the closed loop system. It might be necessary to include actuator and tyre dynamics when deriving the limitations. At larger steering angles at the wheels, the rotation of limits as described in figure 3 will also be of more importance.

6 ACKNOWLEDGEMENTS

The authors would like to thank the Swedish national research programme Gröna bilen.

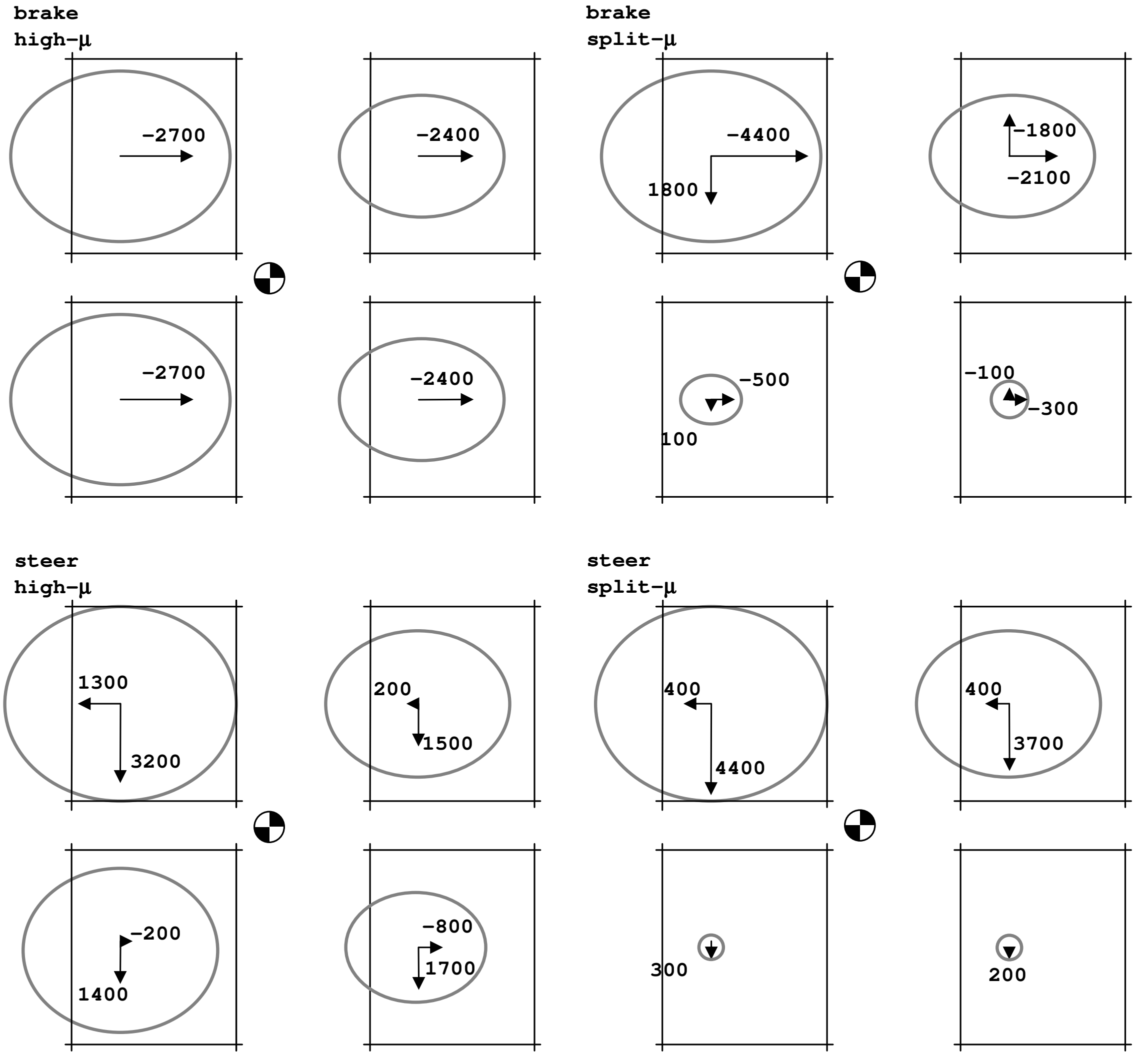


Figure 8: Steady state force distribution for chassis configuration 1, forces f_{x_i}, f_{y_i} are indicated in [N].

REFERENCES

1. S. Zetterström. Electromechanical steering, suspension, drive and brake modules. In *Proceedings of Vehicular Technology Conference, VTC :2002-Fall, Volume 3*, pages 1856–1863, September 24-28 2002.
2. Y. Hattori et al. Force and moment control with nonlinear optimum distribution for vehicle dynamics. In *Proceedings of the 6th International Symposium on Advanced Vehicle Control*, September 9-13 2002.
3. M. Yamamoto. Active control strategy for improved handling and stability. *SAE transactions, paper 91:1902*, pages pp. 1638–1648, 1991.
4. J. Andreasson and L. Laine. Driving dynamics for hybrid electric vehicles considering handling and control architecture. *18th Int. Symp. IAVSD, Dynamics of Vehicles on Roads and Tracks*, August 25-29 2003.
5. L. Laine and J. Andreasson. Generic control architecture applied to a hybrid electric sports utility vehicle. In *Proceedings of the 20th International Electric Vehicle Symposium*, November 15-19 2003.
6. J. Fredriksson, J. Andreasson, and L. Laine. Wheel force distribution for improved handling in a hybrid electric vehicle using nonlinear control. *To be published*.
7. M. Bodson. Evaluation of optimization methods for control allocation. In *AIAA Guidance, Navigation, and Control Conference and Exhibit*, August 2001.
8. Auto Express. http://www.autoexpress.co.uk/news/ae_news_story.php?id=36274.
9. L. Laine, J. Andreasson, and J. Fredriksson. Reusable functional partitioning of tractive force actuators applied on a parallel hybrid electric vehicle. Accepted to *7th Int. Symposium on Advanced Vehicle Control, AVEC*, August 23-27 2004.
10. E. Bakker, H.B. Pacejka, and L. Lidner. A new tire model with application in vehicle dynamics studies. *SAE*

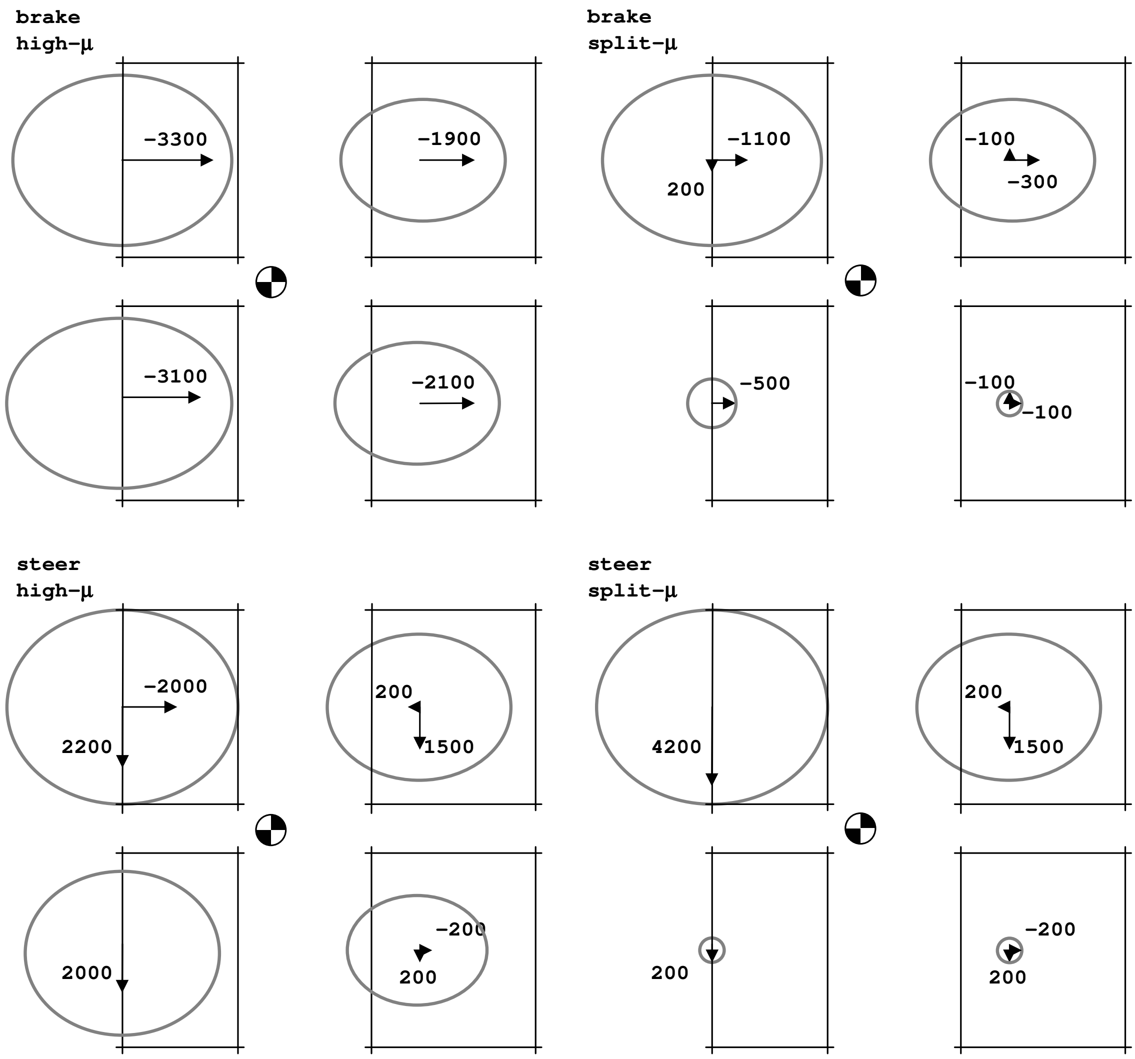


Figure 9: Steady state force distribution for chassis configuration 2a, forces f_{x_i}, f_{y_i} are indicated in [N].

transactions, paper 2003-01-0092, 2003.

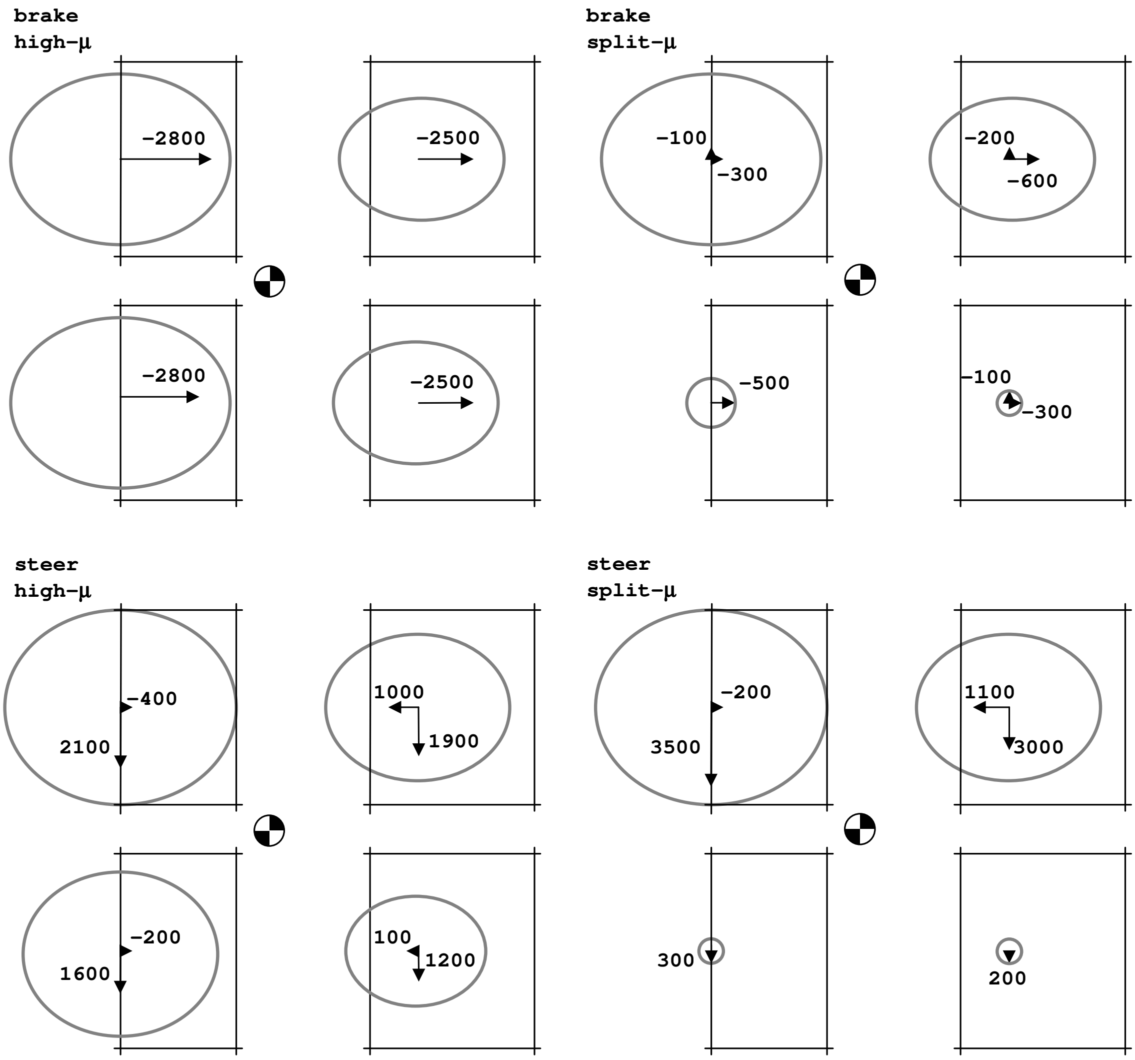


Figure 10: Steady state force distribution for chassis configuration 2b, forces f_{x_i}, f_{y_i} are indicated in [N].

Paper III

Reusable Functional Partitioning of Tractive Force Actuators Applied on a Parallel Hybrid Electric Vehicle

in

Proceedings of the 7^{th} *International Symposium on Advanced Vehicle Control, (AVEC'04)*, Arnhem, Netherlands, August 2004.

Reusable Functional Partitioning of Tractive Force Actuators Applied on a Parallel Hybrid Electric Vehicle

[1]Leo Laine and [2]Johan Andreasson

[1]Chalmers, Dept. of Machine and Vehicle Systems, SE-412 96 Göteborg, Sweden.*

leo.laine@me.chalmers.se

[2]KTH Vehicle Dynamics, SE-10044 Stockholm, Sweden*, johan@fkt.kth.se

This paper suggests a reusable functional partitioned control structure to handle different configurations of tractive force actuators that can be found in Hybrid Electric Vehicles (HEV). The structure is hierarchically defined and allow the same controllers to be used for a variety of HEVs. It also shows how coordination of the Vehicle Motion Controller and Energy Management of a HEV can be made. Modular building blocks are used to define Power Supply, this is illustrated with two different parallel HEV configurations.

1 Introduction

In order to handle the complexity of several actuators/sensors interacting in future Hybrid Electric Vehicles (HEV) and to allow easy change of hardware configuration, a control architecture with suitable functional partitioning is necessary. Coordination of actuators is a necessity for HEVs with hierarchical control structure [1]. Few have shown how the coordination is made between vehicle dynamics and power management for HEVs. In earlier work, a generic architecture is suggested [2] where a functional hierarchical structure is shown.

The architecture is based on functional decomposition and part of it is shown in Figure 1. The vehicles functional tasks are divided into four main tasks, the *Driver Interface* (DIf), the *Chassis* (Ch), the *Power Supply* (PS) and *Auxiliary systems* (Aux), see also [2]. These tasks have their generic controllers: *Driver Interpreter* (DIp), *Vehicle Motion Control* (VMC), and *Energy Management* (EM). Additionally, a *Strategic Controller* (SC) is defined to handle critical states that are declared by either the VMC or the EM.

The idea is that the highest level, level 1, should remain independent of vehicle configuration. This is made possible by defining the vehicle on level 2 based on a set of building blocks that has a corresponding set of equations and boundaries on level 1. The vehicle configuration is determined in the second level.

The functional partitioning of tractive force actuators between Ch and PS, at level 2, is topology dependent. For example, an electrical machine is placed within the Chassis if the torque generated is applied to one wheel, a so-called *wheel motor*. If the electric machine instead generates torque that is distributed via for example differentials, it is considered as a part of the Power Supply.

The outline for the paper is as follows. In section 2 we review the ideas presented in [3] for how DIp gives the desired path to VMC and how desired mechanical and electrical forces are calculated. Section 3 presents the identified tasks such as State of Charge controller and Power Management of PS within EM. In section 4 the SC is presented. It suggests how the interaction between the VMC and EM control is handled, especially for parallel hybrid electric configurations. Section 5 shows how abstraction models of the PS can be derived for the Power Management Controller within EM, it is here presented how a wide span of possible PS configurations can be handled. Similar method was used for deriving abstraction models of different chassis configurations for the VMC, see further in [3]. Finally, in Section 6, the use of building blocks is examplified on two parallel HEV configurations.

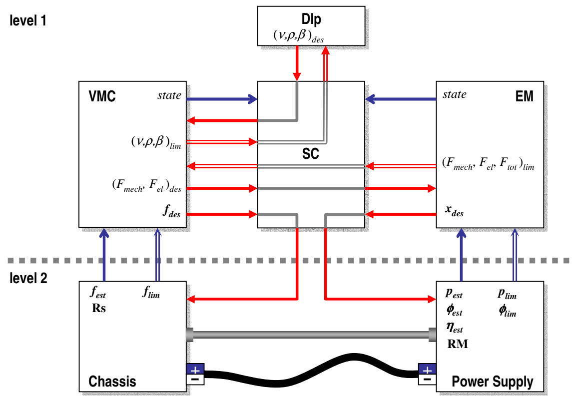


Figure 1: Part of the suggested architecture with focus on the interaction between VMC/Chassis and EM/Power Supply. Double lines indicate limits and single lines are control signals.

2 Vehicle Motion Control (VMC)

The desired path, speed v, curvature ρ, and slip β, is given by DIp to VMC which then have three main tasks, relating to motion control, force distribution and the derivation of limitations for DIp. As seen in Figure 1, VMC receives

*The authors would like to thank the Swedish National Research Programme Gröna bilen.

the estimated forces $\mathbf{f}_{est}$ and force limits $\mathbf{f}_{lim}$ of each Wheel Unit, n. These limits are dependent on the actual wheel configuration and the current tyre-road condition. VMC also receives limits from PS on how much traction can be generated, $F^{lim}_{mech,el,tot}$. Information of *Restrictors* (Rs) such as differentials, rack steer are sent up with the restrictor matrix $\mathbf{R}_s$ that allow additional or extended limitations to be accounted for. A typical example of extended limitations occurs when a wheel is mechanically coupled to PS via a differential Rs. The mechanical limits are then extended by what can be generated by the PS shaft. Additionally, there will be constraints that defines how the force can be distributed between the wheels connected to the Rs.

Once the wheel forces are known, VMC calculates the global desired mechanical and electrical tractive force to be generated by PS by knowing the amount of electrical and mechanical force for each wheel.

$$F^{des}_{mech} = \sum_{i=1}^{n} f^{des}_{x,i,mech} \cdot sgn(v_{res}) \tag{1}$$

$$F^{des}_{el} = \sum_{i=1}^{n} f^{des}_{x,i,el} \cdot sgn(v_{res}) \tag{2}$$

where both are multiplied with the sign of resultant vehicle speed v_{res}, to handle the vehicle's travelling direction.

2.1 Critical State Control

VMC also calculates what dynamical *state* of the vehicle have by studying its dynamical limits. This could be exampliﬁed by studying the actual vehicle path

$$state_{VMC} = \begin{cases} 0, & \text{if } (v,\rho,\beta) \in \mathbf{S}_1 \\ 1, & \text{else} \end{cases} \tag{3}$$

where $\mathbf{S}_1$ is the allowed set of values of possible vehicle paths.

3 Energy Management (EM)

EM controls the power flow in PS so that it can be optimised depending on strategy. To do this, there are three main tasks to be handled; 1) State-Of-Charge (SOC) control of the buffers, 2) Power Management control, and 3) PS state.

3.1 SOC Controller

The SOC controller (1) is used to define a target SOC. There are various methods and objectives to do this but a common way is to consider vehicle speed, acceleration demand and possibly also information about road topology and ahead traffic.

3.2 Power Management Control

Just as illustrated for Ch and VMC in [3]. The Power Management controller (2) within EM should use an abstracted model that is sufficient enough for its objective. In hierarchical structures the main challenge is the extraction of hierarchy of models at various levels of abstraction which are

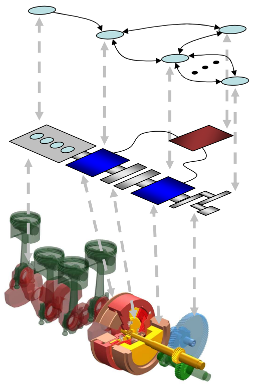


Figure 2: Example of PS on different abstraction levels. The actual power-train, here a strigear (Level 3) and the abstraction levels corresponding to the main components within PS:internal combustion engine, two electric machines, one clutch, and a gearbox (Level 2) and the constrained optimisation problem in EM (Level 1). Strigear picture from [5].

compatible with functionality and objective for each level, see e.g. [4]. This is different from model reduction in the way that the control input signals for level 1 are not the same as for control input signals at level 2. The levels of abstractions are illustrated in Figure 2.

The Power Supply control can be seen as optimisation problem such as

$$\min_{\mathbf{x}} f(\mathbf{x}) \tag{4}$$

$$\mathbf{x}_L \le \mathbf{x} \le \mathbf{x}_U \tag{5}$$

$$\mathbf{b}_L \le \mathbf{A}\mathbf{x} \le \mathbf{b}_U \tag{6}$$

and possibly nonlinear constraints

$$\mathbf{c}_L \le \mathbf{C}(\mathbf{x}) \le \mathbf{c}_U. \tag{7}$$

As described in Section 5, PS is defined by a set of building blocks that allow the constraints and relations $\mathbf{x}_L, \mathbf{x}_U, \mathbf{b}_L, \mathbf{b}_U, \mathbf{A}, \mathbf{c}_L, \mathbf{c}_U, \mathbf{C}$ to be derived directly from PS. There is thus no need to change the problem as PS is modified as illustrated in Section 6.

3.3 Critical State Control

EM decides also if PS is in a critical state (4). An example of a state decision could be

$$state_{EM} = \begin{cases} 0, & \text{if } (SOC_i) \in \mathbf{S}_2 \\ 1, & \text{else} \end{cases} \tag{8}$$

where $\mathbf{S}_2 = \{SOC_i : 0 < SOC_{i,min} < SOC_i < SOC_{i,max} < 1\}$ and ith buffer within Power Supply.

4 Strategic Control (SC)

As seen in Section 2, VMC distributes forces to each wheel and calculates a desired resulting force amount that should be available by PS mechanical and electrical outputs. The available ranges from the actuators in PS, Π_{PS} and Ch Π_{Ch} then form the total actuator limitations $\Pi_{act} = \Pi_{PS} + \Pi_{Ch}$. These are then limited by the tyre-road adhesion Π_{tyre} such that the resulting available forces should be within $\Pi = \Pi_{act} \bigcap \Pi_{tyre}$.

There are though still some issues that needs arbitration by SC. Consider for example a differential that distributes the same amount of tractive force to two wheels. These wheels additionally have individual brakes that allow forces to be unevenly distributed with the wheels. Using brakes to achieve this is though not suitable for normal driving since it leads to increased fuel consumption and brake wear.

In SC the VMC and EM states are considered. This gives priority to either part of VMC and EM. In the above example, if VMC is in a critical state=1, then mechanical braking will be allowed. If instead EM have critical state=1, it maximises the use of regenerative braking and charging. If all buffers within PS is low then the vehicle should perform smooth shutdown, with softly decreasing the vehicles maximum speed, and inform the driver about 'limp home mode'. In Table 1 an example is given how the state controller could be configured. When both states are critical,

VMC state	EM state		Priority VMC/EM
0	0	→	EM
1	0	→	VMC
1	1	→	VMC
0	1	→	EM

Table 1: State controller within Strategic Control, decision on which (VMC and/or EM) limits are active.

focus is on vehicle stability while under normal driving, low fuel consumption and wear is prioritised. The finalisation of orders from EM and VMC to level 2 is also performed by Strategic Control.

5 Abstraction models of PS

PS stores and provides energy mainly in three different domains; namely as fuel, electrical, and mechanical. As illustrated in Figure 3 the energy can be converted between these domains, either be bi-directional or one-way.

The conversion efficiency between the energy domains varies due to what physical converter is used. An internal combustion engine it can be about 0.3 (F/M), for a fuel cell

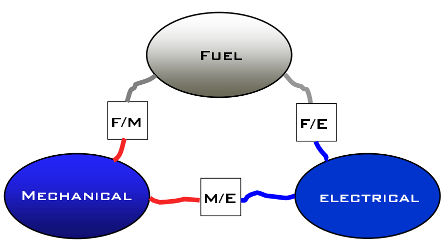


Figure 3: Energy storage domains within Power Supply.

it can be up to 0.6 (F/E), and for an electric machine as high as 0.9 (M/E).

5.1 Building Blocks

To describe the PS in a modular way, building blocks are utilised. This method described here is influenced by [6] and [7].

The PS is partitioned by using the following building blocks; *Buffers* (Bf), *Converters* (Cv), *Transformers* (Tf), *Nodes* (Nd) and *Connectors* (Cn), illustrated in Figure 4. A buffer stores energy. A converter can convert from one energy form to another. This could be done either in one direction as for a combustion engine that only can generate mechanical power from fuel, or both directions as for an electric machine that can both work as a motor and a generator. The transformer modifies flow and potential within an energy form. A typical mechanical transformer is a gear that changes the rotational speed of the shaft. An electrical transformer can instead modify the voltage. There must also be nodes that allows the power flow to split and merge and connectors that allow interaction with for example a chassis. Additionally there are *Sums* (Sm) that are lets flows merge and *Zeros* (Zo) that stops a flow.

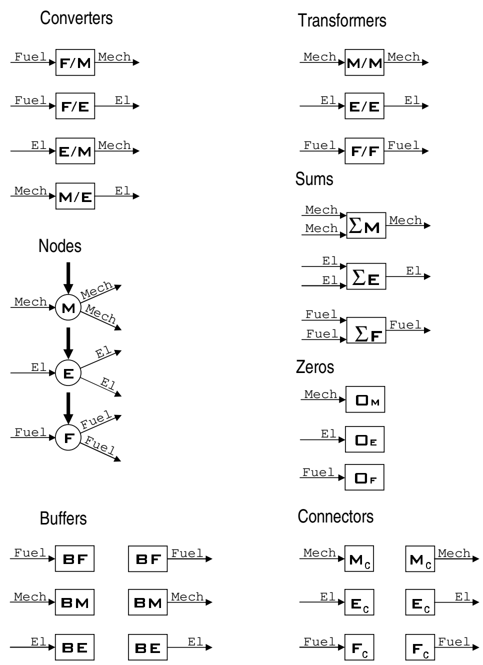


Figure 4: Used building blocks for describing Power Supply configurations and components.

Based on the following rules, these components are used to set up a Power Supply configuration at abstraction level 2:

- All flow paths must begin with Bf and end with either Bf, Cn or Zo.
- Bf, Cn and Zo have only one connection.
- Cv and Tf have always 2 connections.
- Nd and Sm have 3 or more connections.

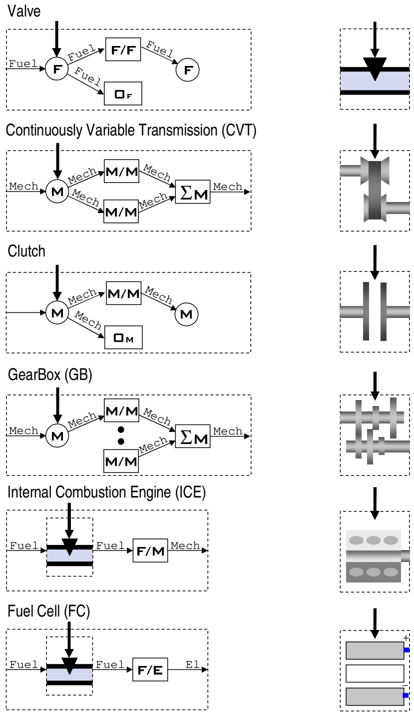


Figure 5: Examples of components modelled by building blocks.

In Figure 5 it is seen how the building blocks defines sample components such as clutch, manual gearbox, internal combustion engine illustrated by the building blocks. As Figure 5 shows nodes are used as way selector e.g. "*Engage clutch or not?*" or "*Which gear should be active?*"

To define what components are connected and how, the relation matrix, $\mathbf{RM} = (r_{ij})$ of a relation R between a finite set of nodes, buffers, and connectors, b, can be defined by

$$r_{ij} = \begin{cases} n, & \text{if } b_i \mathrm{R} b_j \\ 0, & \text{else} \end{cases} \tag{9}$$

where n is the number of positive flow paths from b_i to b_j.

Each component carries their own governing equations that describes their behaviour. These are based on the power flow $x_i = p_i$ or the normalised power flow, $x_i = p_i/p_{ref}$, through the component. Losses are defined with η such that $x_{out} = \eta x_{in}$ and limitations x_{lim} defines a span between x_{min} and x_{max} that the component can handle.

However Nd and Sm do not carry these equations since they only handle the flow. These instead have governing equations $\sum_i^n x_i = 0$ for the Nd, where x_i is the power flow and $\sum_i^n x_i = p_{out}$ or $\sum_i^n x_i = 1$ for the Sm, where $x_i \in \{0/1\}$ or $x_i \in [0,1]$.

Consider for example the CVT in Figure 5, Nd gives the equation for the power distribution between the two branches, each branch then can have losses and limitations and Sm then adds the branches, giving

$$p_{out} = x_1 \eta_1 p_{in} + x_2 \eta_2 p_{in}. \tag{10}$$

Each branch also can have limitations on how much power they can carry. Two complete examples of how a power supply is defined is shown in Section 6.

5.2 Additions to Handle Potentials

As seen above, the power flow is specified in a generic way. When considering a real application, it is often necessary to consider the potential ϕ. A typical example is shift scheduling and to deal with such problems there are further aspects to the building blocks.

Each building block has governing equations that defines the relative change of potential between then in- and outputs. Additionally, reference potentials ϕ_{ref} are needed, once per branch. This is always given by Zo and Cn and when needed also by Bf.

Say for example that there is a PS with a clutch and a Mechanical connector, M_c, to the chassis, the potential is then given by the M_c. Additionally, when the clutch is disengaged, Zo provides the additional potential needed which is equal to the angular speed of the free axle when disengaged.

For converters, the potential equation is simply $\phi_{in} = \alpha \phi_{out}$ with an apropriate factor, α, that relates different types of potential. The transformer has typically a ratio, r, such that $r\phi_{in} = \phi_{out}$.

For Nd, there are two different types, *potential constrained* or *flow constrained*. The difference is best illustrated by two examples. Consider first a mechanical potential constrained node, i.e. two axles with a gear connection with a gear ratio r=1, see Figure 6, left. The following equation defines this component

$$\begin{aligned} p_1 &= p_2 + p_3 \\ \omega_1 &= \omega_2 = \omega_3 \end{aligned} \tag{11}$$

where p_i and ω_i is the power and rotational speed respectively. The node itself do not consider other gear ratios are handled with additional Tf.

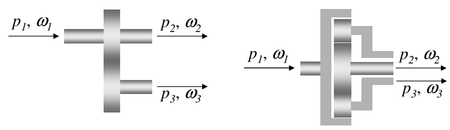


Figure 6: Potential constrained gear, left, and flow constrained gear, right.

For the flow constrained case, illustrated in figure 6, right, the equations are instead

$$\begin{aligned} p_1 &= p_2 + p_3 \\ \tau_1 &= \tau_2 + \tau_3 \\ \tau_3 &= \tau_2 \end{aligned} \tag{12}$$

where τ is the torque flow. Thus, if the power flow through the gear can be controlled so that $p_2 = xp_1$ and $p_3 = (1-x)p_1$ then

$$\begin{aligned} \omega_2 &= 2x\omega_1 \\ \omega_3 &= 2(1-x)\omega_1 \end{aligned} \tag{13}$$

This corresponds to a planetary gear with gear ratio 0.5 or a differential with gear ratio 1. Other ratios are then handled with a transformer.

The example with the CVT in equation 10 can now be completed with the potential equation. Defining $x_1 = x$ and $x_2 = 1 - x$ gives

$$\begin{aligned} p_{out} &= x\eta_1 p_{in} + (1-x)\eta_2 p_{in} \\ \omega_{out} &= (xr_1 + (1-x)r_2)\omega_{in}. \end{aligned} \tag{14}$$

6 Power Management Examples

To illustrate the method described above, two example configurations are used. Both have a parallel hybrid PS configuration with ICE, Integrated Starter/Generator (ISG), clutch, automated manual gearbox (5 speed) and a battery buffer. The first configuration has only a Mechanical connector to chassis, and the chassis is All Wheel Driven (AWD). In the second configuration there are two connectors, mechanical and electrical, to the chassis. Two wheels are driven by the mechanical connector with differential and the two other wheels are driven by wheel motors, here named Electric 4 Wheel Drive (E4WD). The chassis thus requiring both mechanical and electrical power. The interpretation of these configurations are shown in Figure 7.

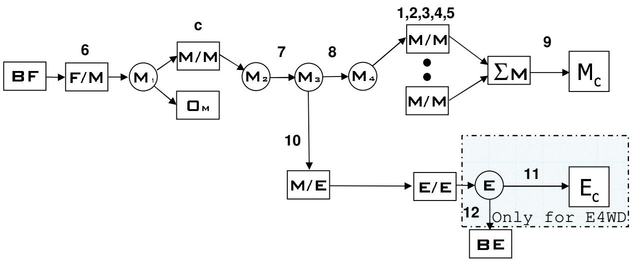


Figure 7: Abstraction model of a parallel PS with ICE, ISG, clutch, and automated manual GB, with Chassis connections: Mechanical-'Mc' and Electrical-'Ec'.

This problem can be seen as a mixed-integer problem, due to the fact that there are decision variables that are integers, e.g. gearbox and clutch.

The following objective function is set up

$$min\left(f(x) = \sum_{j=1}^{n} C_j \cdot x_j \right) \tag{15}$$

where the cost or losses are to be minimised. Here the C_j is the coefficient of the jth decision variable $\boldsymbol{x}$. By minimising the losses C_j is set to

$$C_j = (1 - \eta_j). \tag{16}$$

where η_j is the efficiency.

If losses are neglected the linear constraints, for the **A**, see Eq.4, can be derived from the nodes M_3 and E

$$-x_7 + x_8 + x_{10} = 0 \tag{17}$$

$$-x_{10} + x_{11} + x_{12} = 0. \tag{18}$$

$$\tag{19}$$

The gearbox will deliver the same amount of power and thus

$$-x_8 + x_9 = 0 \tag{20}$$

where x_i is power flow with upper and lower bounds.

The gearbox can be described by the following linear constraint

$$\sum_{i=1}^{5} x_i = 1 \tag{21}$$

where $x_i \in \{0/1\}$. This allows only one gear to be active.

The clutch can be seen as a nonlinear constraint such as

$$-x_8 x_c + x_7 = 0 \tag{22}$$

where x_8, x_7 is the power flow and the clutch with $x_c \in \{0/1\}$. To be able to use the linear mixed integer solver, 'mipSolve' [8], Eq. 22 can be simplified to

$$-x_8 + x_7 = 0. \tag{23}$$

The linear constraints for the E4WD gives the following matrix **A=**

$$\begin{matrix} x_6 & x_7 & & \cdots & & & x_{12} \end{matrix}$$
$$\begin{pmatrix} 0 & -1 & 1 & 0 & 1 & 0 & 0 \\ -1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1 & 1 & 1 \\ 0 & 0 & -1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & -1 & 0 \end{pmatrix} \tag{24}$$

Where linear constraint Eq.21 for the gearbox is excluded. The equality vector for the linear constraints is

$$\mathbf{b}_U^T = \mathbf{b}_L^T = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & P_{M_c}^{des} & P_{E_c}^{des} \end{pmatrix} \tag{25}$$

where $P_{M_c}^{des}$ and $P_{E_c}^{des}$ are the desired mechanical and electrical power needed for the chassis.

The upper and lower bounds for the power flow are defined as static values with maximum input/output power from ICE, ISG, and Buffer, as shown in Table 2.

Decision variable	η_i [-]	$x_{i,L}$	$x_{i,U}$	Unit [-]/[kW]
x_i, $i \in \{1,..,5\}$	$f_{\eta,gb}$	0	1	[-]
x_6	0.35	0	100	[kW]
x_7	0.95	0	100	[kW]
x_8	0.95	-130	130	[kW]
x_9	0.95	-130	130	[kW]
x_{10}	0.9	-30	30	[kW]
x_{11}	0.95	-50	-50	[kW]
x_{12}	$f_{\eta,Be}$	-20	-20	[kW]

Table 2: Efficiency, lower and upper bounds and desired power from Chassis.

Static values for the efficiency are used except for the gear selection x_i, $i \in \{1,..,5\}$ and the Buffer, Be , x_{12}. The efficiency of each gear is evaluated by function $f_{\eta,gb}(\omega_{eng}, \tau_{eng})$. Which considers the efficiency map of the engine, see Figure 8.

Possible engine speeds are calculated by

$$\omega_{eng} = \omega_c \cdot \mathbf{r} \tag{26}$$

where, ω_c, is the known potential at the mechanical chassis connector M_c and **r**=[3.46 1.94 1.29 0.97 0.81] are the gear ratio's. The wheel radius was assumed to be 0.3 m, and the final gear 3.24. The possible engine torque's are evaluated by

$$x_6 = \tau_{eng} \cdot \omega_{eng} \quad (27)$$

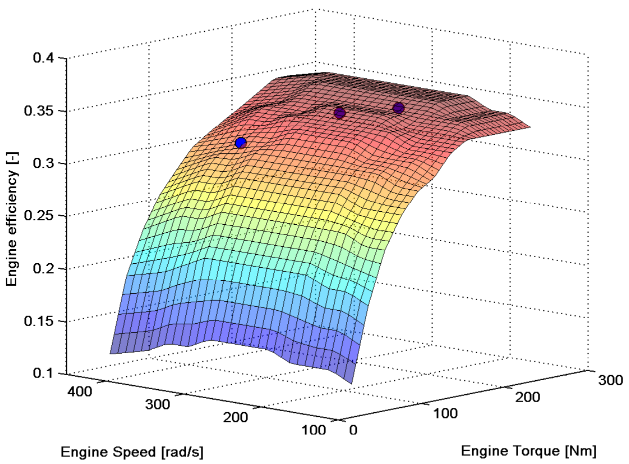


Figure 8: Efficiency of an ICE, 1.5L Prius with Atkinson cycle [9], scaled to maximum output power 100 kW.

An artificial efficiency is used for the buffer Be, which allows negative efficiency during low SOC, and positive efficiency during high SOC, this to enforce the solver to charge the buffer when ever feasible within the linear constraints (see Eq. 28)

$$f_{\eta,Be} = w \cdot \tanh(8 \cdot SOC - 4) \quad (28)$$

where w is a weighting constant.

Three different power demands where applied to the two examples E4WD and AWD all at vehicle speed v=90 km/h, as illustrated in Table 3. In column 2, a total power

$P_{M_c}^{des}$	-30	-30	10	60	60	20
$P_{E_c}^{des}$	-30	-30	10	N/A	N/A	N/A
SOC	0.7	0.2	0.7	0.7	0.2	0.7
gear	5	4	-	5	3	-
x_6	40	60	0	40	80	0
x_9	30	30	-10	60	60	-20
x_{10}	10	30	10	-20	20	20
x_{11}	30	30	-10	N/A	N/A	N/A
x_{12}	-20	0	20	-20	20	20

Table 3: Results for different power demands. Columns 2 to 4 are results for the E4WD case and the columns 5 to 7 are the results for the AWD case.

of 60 kW is demanded from the E4WD Chassis connectors. High available SOC allows the buffer to be used. The fifth gear is selected due to highest efficiency. This is also shown in Figure 8 where three dots on the map illustrates gear 3, 4, and 5. In column 3, the SOC is low and thus all power is selected from ICE, x_6. Column 4 Shows and deacceleration demand of 20 kW all power is used to charge the buffer, 50 percent via ISG.

The same total power demand of 60 kW for the AWD is shown in Table 3, column 5, here the E_c connector is not available (N/A). It also uses the buffer due to high SOC. In column 6 the SOC is low and it chooses to charge the buffer. In this case the third gear is selected as optimum. Column 7 shows a deacceleration and obviously chooses to charge the buffer, 100 percent via ISG.

7 Conclusions and Future Work

The paper shows how the interaction of the Vehicle Motion Controller and Energy Management is made. It also suggests how abstraction models of PS can be defined with building blocks. These models makes it possible to handle wide span of PS configurations without changing the controllers at the highest level of the hierarchical control structure. Two Examples illustrates how the models can be built up and a mixed integer solver was used to solve the power demands from the chassis.

Next step is to implement dynamical models of the components that give information of actual, upper and lower bounds for the decision variables and their reference potential.

References

[1] R. Zhang and Y. Chen. Control of hybrid dynamical systems for electric vehicles. *Proc. of American Control Conf.*, June 2001.

[2] L. Laine and J. Andreasson. Generic control architecture applied to a hybrid electric sports utility vehicle. *Proc. 20th Int. Electric Vehicle Symposium*, November 2003.

[3] J. Andreasson, L. Laine, and J. Fredriksson. Evaluation of a generic vehicle motion control architecture. *FISITA*, May 2004.

[4] George J. Pappas. Hierarchically consistent control systems. *IEEE Transactions on Automatic Control*, Vol. 45(6), June 2000.

[5] Stridsberg Powertrain AB. http://www.powertrain.se/.

[6] M. Åsbogård. Evaluating hybrid vehicle concepts using optimal control and generic powertrain modelling. *Proc. of the Advances in Automotive Control Symposium*, April 2004.

[7] G. Rizzoni, L. Guzzella, and B. M. Baumann. Unified modeling of hybrid electric vehicle drivetrains. *IEEE/ASME Transactions on Mechatronics*, VOL. 4(NO. 3), September 1999.

[8] K. Holmström, A. O. Göran, and M. M. Edvall. *Users Guide for TOMLAB 4.2*. www.tomlab.biz, 2004.

[9] Advanced Vehicle Simulator, ADVISOR. http://http://www.ctts.nrel.gov/analysis/advisor.html.

Paper IV

Reusable Control Architecture Implemented in a Scale Model of a Hybrid Electric Vehicle

in

Technical Report, Division of Vehicle Safety, Department of Applied Mechanics, Chalmers University, March 2005.

Reusable Control Architecture Implemented in a Scale Model of a Hybrid Electric Vehicle

Leo Laine [a,*], Jonas Hellgren [a], Henrik Kinnunen [a], and Magnus Rönnberg [a]

[a] *Div. of Vehicle Safety, Dep. of Applied Mechanics, Chalmers University of Technology, SE-412 96, Göteborg, Sweden* [1]

Abstract

A reusable control architecture for Vehicle System Control has been implemented in a scale model 1:5, of a series Hybrid Electric Vehicle with a Power Supply containing a fuel cell emulator as a primary power unit and supercapacitors as a buffer. The aim is to verify the effectiveness of the reusable control architecture with real hardware by using a scale model car. This type of architecture allows for easy exchange of hardware configurations without having to change the functional structure of the Vehicle System Controller. The structure for the Hybrid Electric Vehicle system is obtained through functional decomposition, which orders the system functions into different functional levels. Three functional levels have been defined. The highest level contains functions that are common for all foreseen Hybrid Electric Vehicles: Driver Interpreter, Vehicle Motion Control, Energy Management, and Operative Decision Control. The second level contains the necessary subsystems for a vehicle: Driver Interface, Chassis, Power Supply, and Auxiliary Systems. The third level is the actuator/sensor level. Using hardware independent signals between the functions allows for hardware configurations to be changed in modular fashion without affecting the higher functional levels. The Scale Model Car was tested and the logged data verified against simulation models both for ordinary drive cycle results and anti-skid behaviour with decent agreement.

Key words: Vehicle System Control, Hybrid Electric Vehicle, Reusable Control Architecture
PACS:

* Corresponding author. Tel.:+46 (0)31-772 5852; fax:+46 (0)31-772 3690
Email address: `leo.laine@me.chalmers.se` (Leo Laine).

[1] The authors are pleased to acknowledge the financial support from the Gröna Bilen National Research Programme within the FCHEV framework.

1 Introduction

Already today vehicles are becoming increasingly dependent on computers and their software controllers. Therefore, it is important that the control architecture be reusable, enabling different vehicle configurations to be designed with minimum effort. In order to handle the complexity of several actuators and sensors interacting in future Hybrid Electric Vehicles (HEVs) and to allow for easy exchange of hardware configuration, a control architecture with suitable functional partitioning is necessary [1], [2]. The architecture should not only be reusable but should also work with several types of hybrid powertrain configurations. It must also fulfill interface requirements between automotive suppliers and manufacturers so that brand specific qualities can be kept in-house [3], [4] [2].

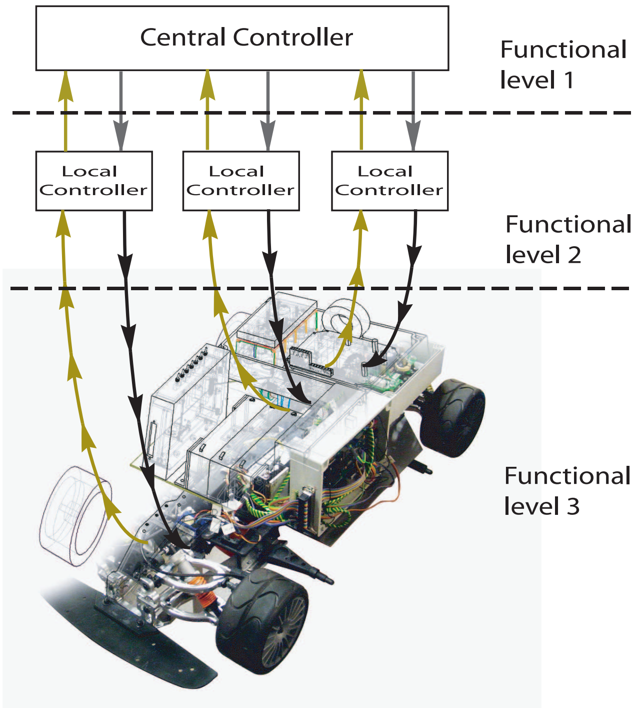


Fig. 1. Hierarchical control architecture implemented in the Scale Model Car. Functional level 1 includes the central controller, functional level 2 includes the local controllers, and functional level 3 is the actual hardware.

The objective of this study is to implement a reusable control architecture in a Remote Controlled (RC) Scale Model Car (SMC) of a series HEV, see Figure 1. The length of the car is 0.9 meter. The implemented reusable control architecture is based on hierarchical partitioning. The hierarchical structure

[2] Brand specific qualities of vehicles are more and more dependent on the used algorithms in the software, which makes it important for manufacturers to protect 'their' algorithms.

then contains three functional levels. The highest functional level consists of a central controller. Functional level 2 includes several low level control blocks. The third level is the sensor and actuator level, as shown in Figure 1. Hierarchical partitioning allows for good modularity and coordination between the different low level control blocks. If the hardware is changed within the system normally only the local controller needs to be changed, see Figure 1.

Computer based vehicle modelling and simulation are useful tools for examining different vehicle control architectures. However, since it is necessary to simplify modelled hardware in simulations it is therefore crucial to verify and test ideas with real hardware. A generic hierarchical control architecture was developed, modelled, and tested on different simulated hardware configurations [5], [6], [7], and [8]. The main features of this generic control architecture were implemented and tested on the SMC.

There are two major aspects to consider when using scale model HEVs. Firstly, building full scale HEVs is very expensive and time consuming. A more cost effective alternative is to use scale models to study vehicle behaviour and controller development [9], [10], [11], [12], [13], and [14]. Secondly, it is important to be able to predict what the scale model would correspond to in a full scale version. This can be done with dimensional analysis, such as the PI Buckingham Theorem, [16]. This method has been used to study controllers for vehicle lateral dynamics, [17] and [18]. Here in this paper, a dimensional analysis has been made on what the scale model HEV would correspond to in full scale version.

The outline of the article is as follows. Section 2 describes how the reusable control architecture is structured and gives an overview of implemented control strategies and algorithms. Section 3 shows the actual hardware used in the SMC. Section 4 illustrates how the control architecture and algorithms are implemented in the control unit. Section 5 discusses what kind of full scale vehicle the SMC would correspond with by dimensional analysis based on PI-Buckingham theorem. Section 6 discusses how the SMC performs during test runs. Section 7 includes discussions and future work. Finally, the appendix includes a nomenclature list and technical specifications on used hardware.

2 Methodology used to design a reusable control architecture

A hierarchical control architecture provides better modularity compared to that of a centralised architecture. Additionally, the coordination between local controllers is also improved compared to a that of a peer architecture. The hierarchical architecture is a suitable partitioning scheme for HEVs. In [1] hierarchical partitioning is recommended. Different partitioning schemes are also discussed in [2].

Generic interface signals were used between the local and top level controllers because they allow for hardware to be exchanged without affecting the top level controller. Generic control signals are exemplified here with a simple example, considering that we have different hardware to drive and steer a vehicle, see Figure 2. In Case 1 we have a steering wheel, brake and gas pedal. In Case 2 we have a joystick with longitudinal and lateral motion. Case 1 has three sensor signals, $[\alpha_1, \alpha_2, \alpha_3]$, sent from functional level 3 to level 2, while Case 2 has only two, $[\beta_1, \beta_2]$. Generic[3] control signals are used if and only if the signals S_1 and S_2 between Functional level 2 and 1, are equal ($S_1 = S_2$). This allows Functional level 1 to be reused despite changes of hardware configurations.

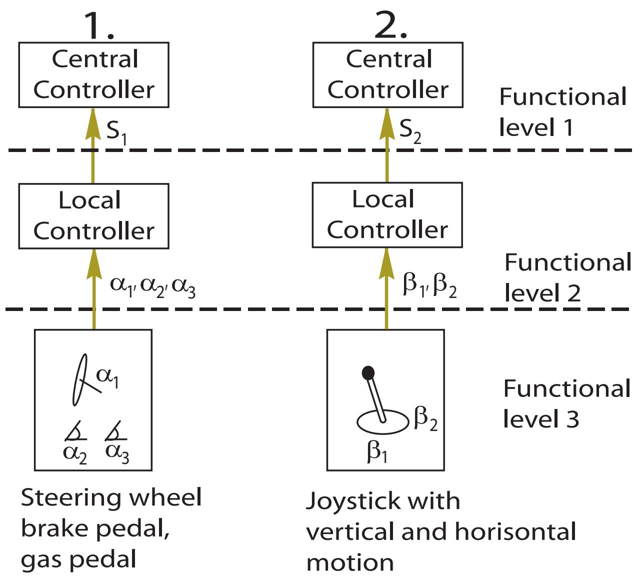


Fig. 2. Two Simple examples of using generic interface signals with different hardware. Case 1: Steering wheel, brake and gas pedal. Case 2: Joystick with longitudinal and lateral motion.

The vehicle system can be seen as a set of functions. Functions within hierarchical control architectures are consigned into different levels by functional decomposition, see Definition 2.1. In this article, reusable control architecture and functional decomposition are defined as follows:

Definition 2.1 *Functional decomposition*
Following statements characterise an architecture with functional decomposition:

(1) Functions are placed into different levels due to their coordinating authority over other functions.

[3] Generic means here hardware independent.

(2) Information on the system status can be observed by all functions and is allowed to flow in all directions, up, down, and across in the hierarchy.
(3) Commands are only allowed to flow down to lower level functions. This upholds a causality of orders within the hierarchical architecture.
(4) Vehicle brand characteristics should only be contained within the top level functions. [4]
(5) Low level functions should have control over hardware health and durability. [5]

Definition 2.2 *Reusable control architecture for HEVs*

(1) The control architecture should be hierarchical by functional decomposition.
(2) Interfaces between top level and lower level functions should be made hardware independent. [6]
(3) The control architecture should be designed so as to accommodate any foreseeable future hardware developments. [7]

2.1 Functional levels

The control architecture's overall function is to collect and analyse information about the vehicle's internal and external conditions and to initiate appropriate responses.

The control architecture is divided into three functional levels:

Level 1: The highest functional level is the main switching unit within the vehicle's architecture. It is where signals flow to and originate from. It relays messages and compares and analyses information. Using generic interface signals allows level 1 to become hardware independent.

Level 2: The second level contains the basic functional tasks of any ground vehicle. These functional tasks can include, for example, generating ground motion, interaction with the driver, power supply and auxiliary systems.

[4] Item 4 allows manufacturers to retain ownership of brand specific functions while suppliers can provide controls for various subsystem functions. Through this, manufacturers can change vehicle characteristics such as optimizing drivability and fuel economy.
[5] Item 5 makes the supplier responsible for the durability of its hardware.
[6] Item 2 allows hardware to be exchanged without redesigning the functional architecture.
[7] For example, this could include future versions of HEVs with Wheel Units, which can independently apply traction, steering, and suspension forces.

Level 3: The third level is the sensor and actuator level. These are controlled and coordinated by different basic functional tasks in level 2.

2.2 Functional level 1

Functional level 1, Main Control, consist of three major parts. Driver Interpreter interprets the driver's demands. Vehicle Motion Control controls the vehicle according to these demands. Energy Management assures that this is done in a energy efficient way. Additionally, Operative Decisions summarizes the input from Energy Management and Vehicle Motion Control and makes the overall decisions considering reliability and safety. Functional level 1 and its subsystem dependencies within the hierarchical architecture are illustrated in Figure 3.

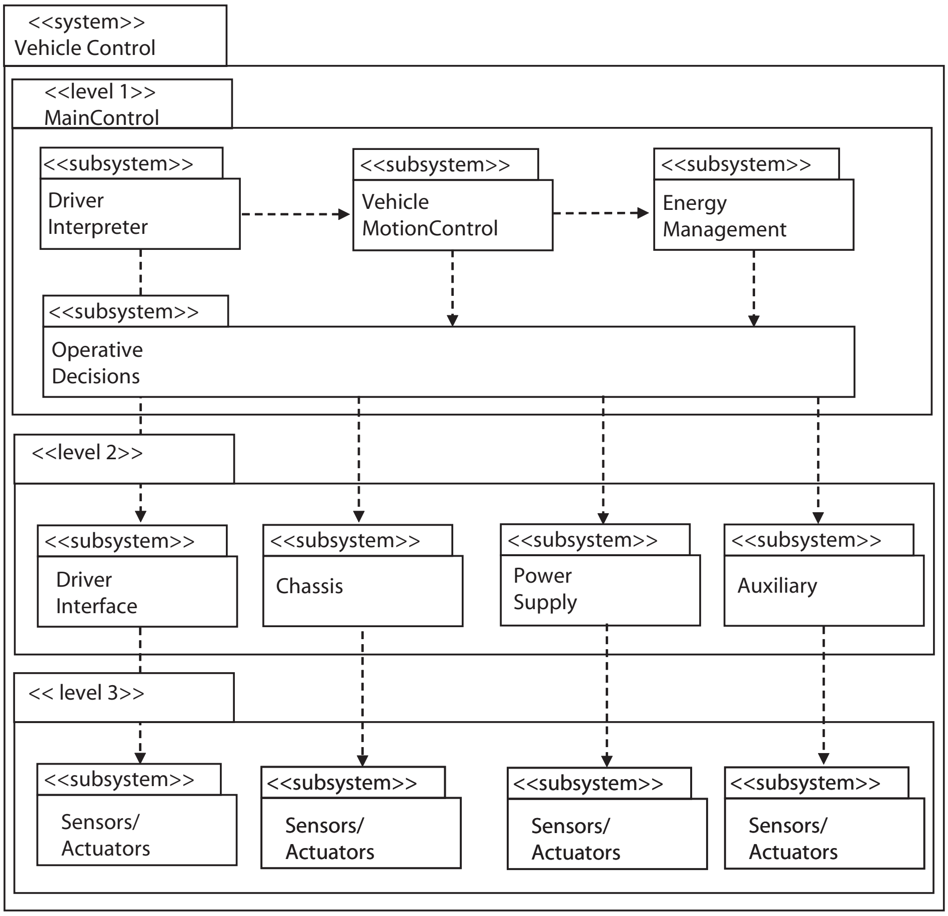


Fig. 3. System architecture and subsystem dependencies according to Unified Modeling Language (UML) [25]. The architecture is made up of three functional levels. Level 1 is responsible for system coordination. Level 2 contains the basic functional tasks of a ground vehicle. Level 3 is the actual vehicle hardware.

2.2.1 Driver Interpreter (DIp)

The driver's instructions are translated into desired motion within Driver Interpreter (DIp). This is done by reading and analysing the sensor signals received from Driver Interface on level 2.

2.2.2 Vehicle Motion Control (VMC)

Vehicle Motion Control (VMC) handles the safety aspects of the vehicle's dynamics. It assures that the vehicle is avoiding a critical dynamic state. For example, typical subfunctions could include traction control, anti-lock braking, or vehicle stability. It uses the desired motions received from DIp and Chassis sensor readings to consider what motions are possible without reaching the critical dynamic limits of the vehicle.

The vehicle's dynamic state must be within a certain allowed set of states, otherwise it is determined to be critical (state=1). For example, a simple slip controller is used here with the following expression

$$state_{VMC} = \begin{cases} 0, & \text{if } (\lambda_{rear}) \in \mathbf{S}_1 \\ 1, & \text{else} \end{cases} \tag{1}$$

where $\mathbf{S}_1$ is the allowed set of slip values for the rear wheel. When the state is equal to 1 VMC suggests that the desired longitudinal velocity from DIp is reduced with the following expression

$$VMC.x_{vel} = DIP.x_{vel} \left(1 - |\lambda_{rear}|\right). \tag{2}$$

The desired signals and state are then sent to Operative Decisions, see further in Section 2.2.4.

2.2.3 Energy Management (EM)

Energy Managment (EM) controls the vehicles energy sources for efficiency with regards to fuel consumption and wear. It decides how the energy flow is distributed between the Primary Power Unit (PPU) and the Buffer considering the current power demand for generating ground motion and auxiliary systems.

EM considers if the Power Supply is in a critical state and passes the information along to Operative Decision. The state is evaluated by the following expression

$$state_{EM} = \begin{cases} 0, & \text{if } (SOC_i) \in \mathbf{S}_2 \\ 1, & \text{else} \end{cases} \tag{3}$$

where $\mathbf{S}_2 = \{SOC_i : 0 < SOC_{i,min} < SOC_i < SOC_{i,max} < 1\}$ and is the ith buffer within Power Supply.

EM calculates a State of Charge (SOC) reference value for current vehicle states, for example, vehicle velocity. The SOC reference is a numerical value representing the current desired SOC for the buffer. One example of a simple Power Management algorithm using a SOC reference within EM is as follows

$$SOC_{ref} = 0.5 + C_0 \left(0.5 - \left(\frac{x_{vel}}{6} \right)^2 \right) \tag{4}$$

where the C_0 and C_1 are constants.

By using SOC reference values and sensor readings of the current SOC, EM distributes the requested power to both the PPU and the buffer. Here is an example of how the buffer power is then decided,

$$P_{buff} = \begin{cases} -k_1 \left(SOC_{ref} - SOC \right), & \text{if } (x_{acc}) \leq a_1 \\ k_2 \cdot P_{em}, & \text{if } (x_{acc}) > a_2 \\ k_3 \cdot P_{em}, & \text{else} \end{cases} \tag{5}$$

where a_1, a_2, k_1, k_2, and k_3 are constants.

A more sophisticated rule based algorithm for calculating the buffer power demand will be implemented according to [21].

2.2.4 Operative Decision (OD)

Operative Decisions (OD) considers the vehicle state values given by VMC and EM. If the vehicle status is critical, OD then gives priority to either VMC or EM. As an example, if VMC is in a critical state=1, OD will allow mechanical braking. If instead EM has critical state=1, OD approves the use of maximum regenerative braking and charging. In Table 1 an example is given as to how the OD state controller could be configured. When both states are critical

VMC state	EM state		Priority VMC/EM	Comment
0	0	→	EM	Prioritise efficiency if no critical state
1	0	→	VMC	Prioritise vehicle stability if VMC critical
1	1	→	VMC	Prioritise vehicle stability if both critical
0	1	→	EM	Prioritise efficiency if EM critical

Table 1
The Operative Decisions state controller gives priority to either VMC or EM depending on the vehicle's states.

focus is on vehicle stability. While under normal driving conditions low fuel consumption and minimising wear are prioritised.

The desired actions from VMC and EM are finalised into orders by OD. These orders are then sent to level 2.

2.3 Functional level 2

Functional level 2 contains the basic tasks of any ground vehicle. Driver Interface reads the sensor signals from the driver. Chassis generates the ground motion. Auxiliary systems includes all subsystems which are not necessary for generating ground motion. Finally, Power Supply generates the needed mechanical and electrical energy for Chassis and Auxiliary systems.

2.3.1 Driver Interface (DIf)

Driver Interface (DIf) reads the sensor signals from the driver. These are normalised to be values between $[-1, 1]$. All software functions associated with reading hardware used by the driver are located in DIf.

2.3.2 Chassis (Ch)

The software functionality of actuators and sensors that directly affect Chassis (Ch) dynamics are placed within Ch. Brake servos and wheel motors are examples of Ch actuators. Accelerometers and wheel rotation sensors are examples of Ch sensors. Ch is mainly controlled by VMC.

2.3.3 *Power Supply (PS)*

Power Supply (PS) contains all local controllers of actuators, buffers, and sensors which are needed to produce the vehicle's power demand. The energy can be stored in different forms such as electrical, fluid, and mechanical. A topology 'cut' is used to determine whether tractive force actuators such as electric motors are placed within PS or Ch. For example, if an electric motor is mounted before a differential its function is placed within PS. PS is mainly controlled by EM.

2.3.4 *Auxiliary Systems (Aux)*

All subsystems not directly related to generating vehicle motion are contained within Auxiliary Systems (Aux). Aux is supervised by EM.

3 Scale Model Car (SMC)

The Scale Model Car (SMC) is a standard model car of size 1:5. The Ch includes suspension, wheels, and body from a manufacturer named 'FG Modellsport'[8] [19]. Pictures of present configuration of the SMC are shown in Figure 4. Details about the design and development process of the Hybrid Electric SMC are given in [20].

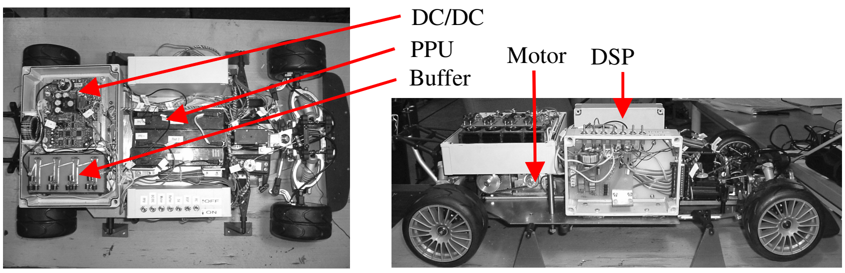


Fig. 4. Top view (left) and Side view (right) of SMC.

3.1 *Chassis Configuration*

There are mainly two actuators within Ch. The first one is the steering servo connected to the rack steer A_{st}. The second is the mechanical brakes on the front wheels which are servo controlled A_{br}. A schematic sketch of the Ch configuration is shown in Figure 5. The Figure shows also how the actuators between Ch and PS are divided by the mechanical connector 'Mc'.

[8] The standard configuration of the SMC has an internal combustion engine.

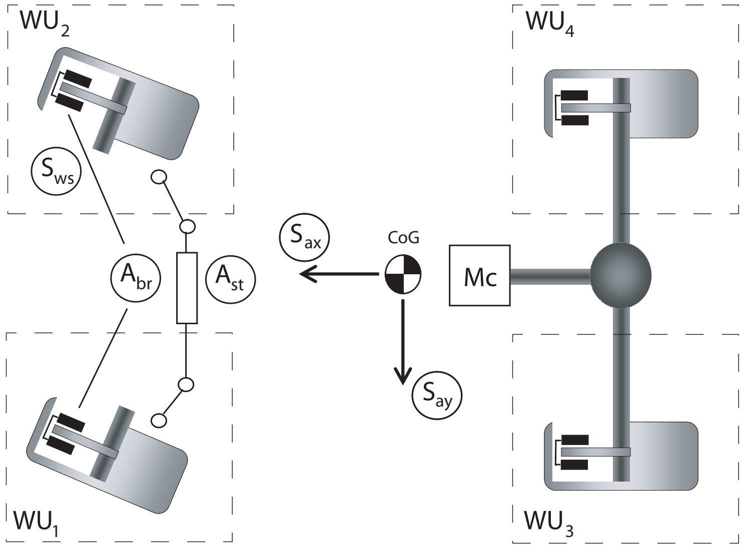


Fig. 5. Illustration of SMC's chassis configuration and which actuators and sensors are located within Ch.

There are mainly three sensors within Ch. The first one is the rotational sensor circuit S_{ws} mounted on the front right wheel (WU_2), see Figure 6. Details about the wheel rotation sensor can be found in [22]. Two accelerometers are mounted on the car giving the longitudinal S_{ax} and lateral acceleration S_{ay}.

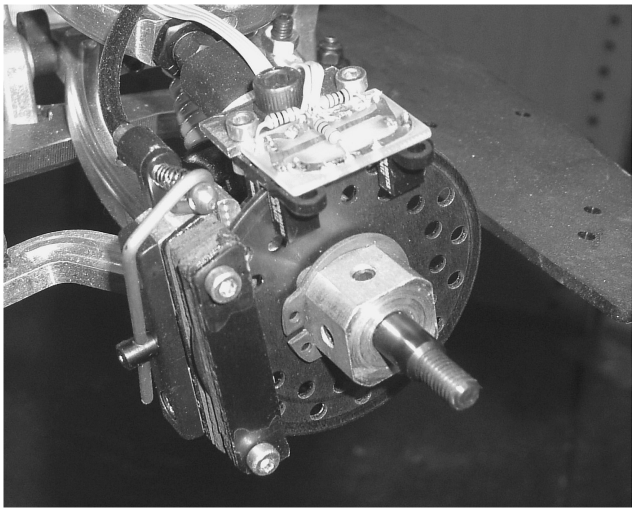


Fig. 6. The optic rotational sensor mounted on front right wheel.

One of the basic functions of Ch is also to estimate the actual vehicle speed. Following simple but still efficient algorithm is used

$$x_{vel} = \begin{cases} x_{vel,nobrake} = \omega_w \cdot R_w, & \text{if no mechanical brake} \\ x_{vel,brake} = x_{vel,nobrake} + \sum_{i=k_{nobrake}}^{k} x_{acc} \cdot dt, & \text{else.} \end{cases} \tag{6}$$

If mechanical brakes are not used the vehicle velocity is calculated as the rotational speed multiplied by the wheel radius. If mechanical brakes are applied the front wheels may slip and thus the last velocity value with no braking, $x_{vel,nobrake}$, at step $k_{nobrake}$, is used and the accelerometer reading is numerically integrated and used to estimate the actual vehicle velocity [24].

Another basic function of Ch is to estimate front and rear slip

$$\lambda_j = \frac{R_{w_j} \cdot \omega_j - x_{vel}}{max\left(R_{w_j} \cdot \omega_j, x_{vel}\right)} \text{ , where j=front, rear.} \tag{7}$$

In table A.1 a summary of vehicle dynamic parameters are listed and further details can be found in [23].

3.2 Power Supply Configuration

The present PS configuration includes a battery as a PPU used as a Fuel Cell emulator. Supercapacitors are used as a buffer, and a DC/DC converter directs the electrical power flow. An electric motor is used to convert the electrical power to mechanical power to propel the vehicle. A schematic diagram of the electrical connections of PS is shown in Figure 7. Four external voltage and current sensors are implemented. This allows supervision of the actual power flow to buffer and electric motor.

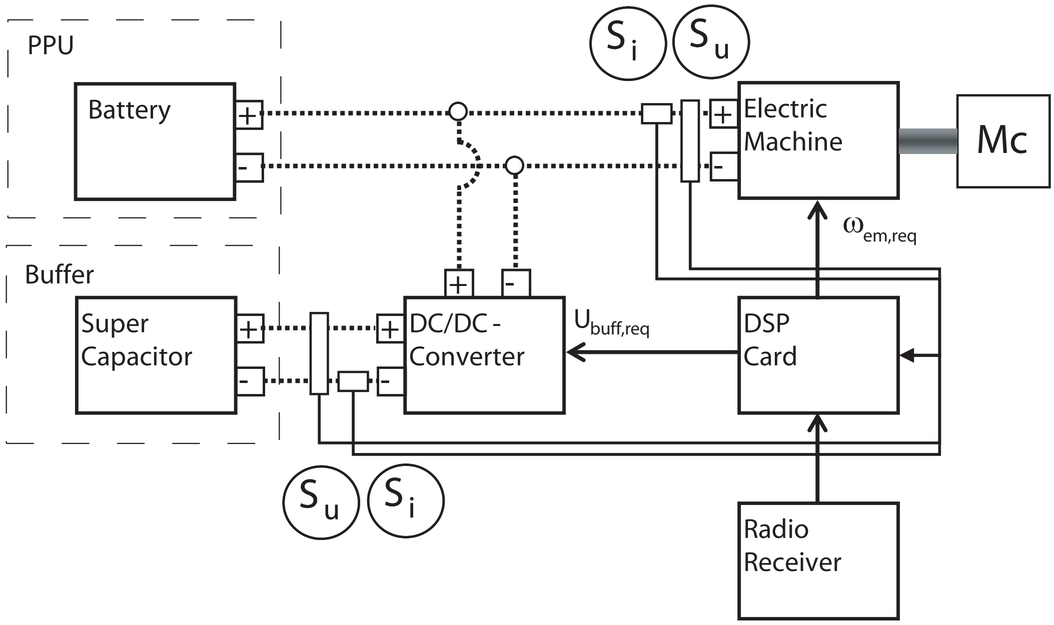


Fig. 7. Illustration of SMC's Power Supply configuration.

The electric motor is a brushless synchronous DC-motor. Power electronics are included so the rotation speed, $\omega_{em,req}$, is easily controlled. The machine can operate in 4 quadrants, in other words it can be used as a generator. The operating voltage is 24 V, fundamental data are given in Table A.2.

The buffer is made of 3 supercapacitors that are connected in series. In Table A.3 the characteristics for one supercapacitor are given. By connecting 3 supercapacitors in series an operative voltage of 7.5 V is achieved. The energy capacity is about $1100 \cdot 3 = 3300$ J. The energy allows the vehicle to be accelerated to a vehicle velocity of 3.6 m/s with maximum acceleration 1.5 m/s^2 nine repetive times. The maximum velocity of the vehicle is 4.51 m/s, which is limited by the maximum rotational speed of the electric motor.

The energy flow in and out from the buffer is handled by the full bridge DC/DC converter by a requested buffer voltage, $U_{buff,req}$. A PI-controller was implemented in PS to control the requested buffer voltage over the DC/DC converter. The input signal for the PI-controller is the difference between desired buffer power and estimated actual buffer power. The output from the PI-controller is the desired voltage for buffer. Characteristics for DC/DC converter are found in A.4.

4 Implemented Vehicle System Control Code and Structure

A Technician downloads the VSC code to the Digital Signal Processor (DSP) card [9] . The downloaded VSC code must interact with different different input and output signals. A Driver gives input such as desired longitudinal and lateral motion, braking, and power switch [10] . Due to the fact that it is a hybrid electric vehicle the decision over using mechanical or regenerative braking is decided by the Vehicle System Control (VSC) Code. Sensor signals are interpreted and used to estimate the vehicle internal states. Examples of such sensor signals are WU rotational speed, motor speed, accelerometers, current and voltage sensors. These input signals are processed by the VSC and final output request signals are sent to the actuators such as electric motor, DC/DC voltage, steer servo, and mechanical brake servo. A system context class diagram of the current configuration of the SMC is shown in Figure 8. The Figure illustrates what sensors and actuators are available in the current

[9] The DSP used is a TMS320LF2407A processor from Texas Instruments which is mounted on a evaluation module from Spectrum Digital. The downloaded code is written in C, which is supported by the development Code Composer from Texas Instruments [15].
[10] The SMC is controlled by a RC system, a Hitec Laser 4 FM transmitter, and a Hitec HFS-04MG receiver.

configuration of the SMC.

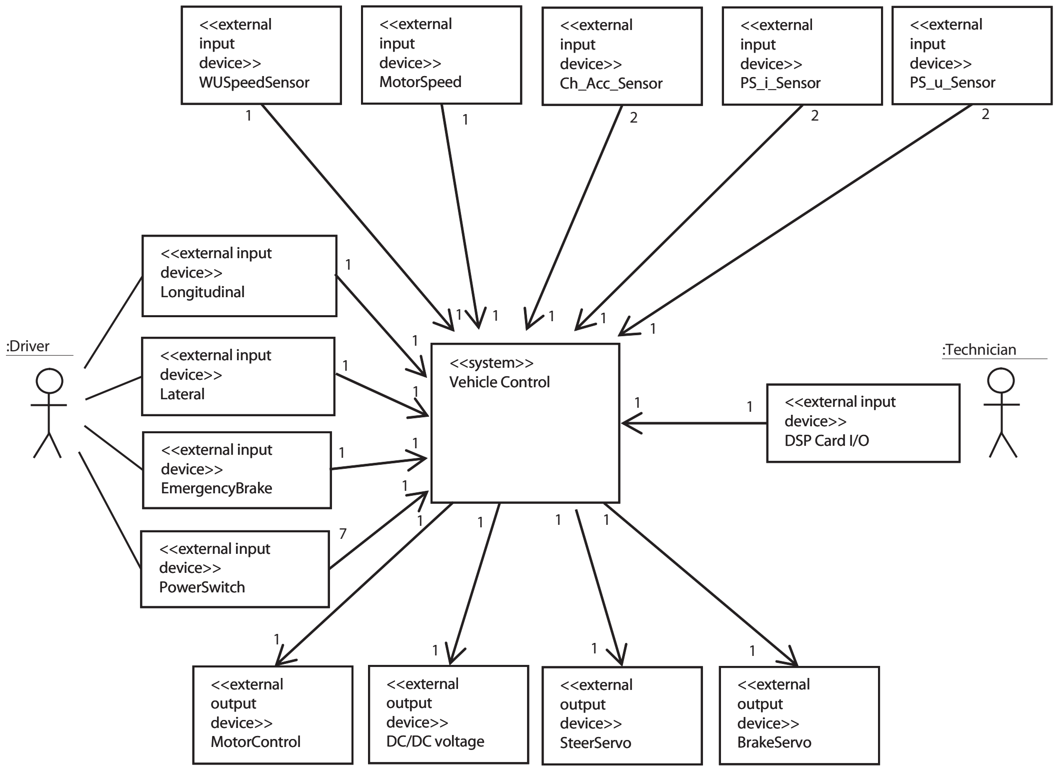


Fig. 8. System context class diagram of the SMC, according to [25].

However, it is important to make a functional decomposition of the VSC to allow easy exchange of both software functions and hardware, in other words, it should not be hardware dependent. For example if the Ch or the PS hardware is changed then only the software within these subsystems would have to be changed. This makes the control architecture reusable for a wide range of hardware configurations. The highest functional level in the control architecture will almost be unchanged when different hardware configurations are used.

During the initial modelling of the problem domain, in this case the current configuration of the SMC, the real world classes are determined. In a real-time embedded system, the real-world classes are primarily physical I/O devices like sensors and actuators, [25], as shown in Figure 9.

DIf is responsible of reading signals from the RC-receiver and making these signals normalised, [-1,1], and available to the rest of the VSC code, see Figure 9. In table A.5 the used signals from DIf are shown. If the brake signal is actually regenerative or mechanical braking is decided by the functional level 1 of the VSC code.

Ch calculates the actual vehicle dynamical states such as longitudinal speed, longitudinal and lateral acceleration, WU slip, and rotational speed of the

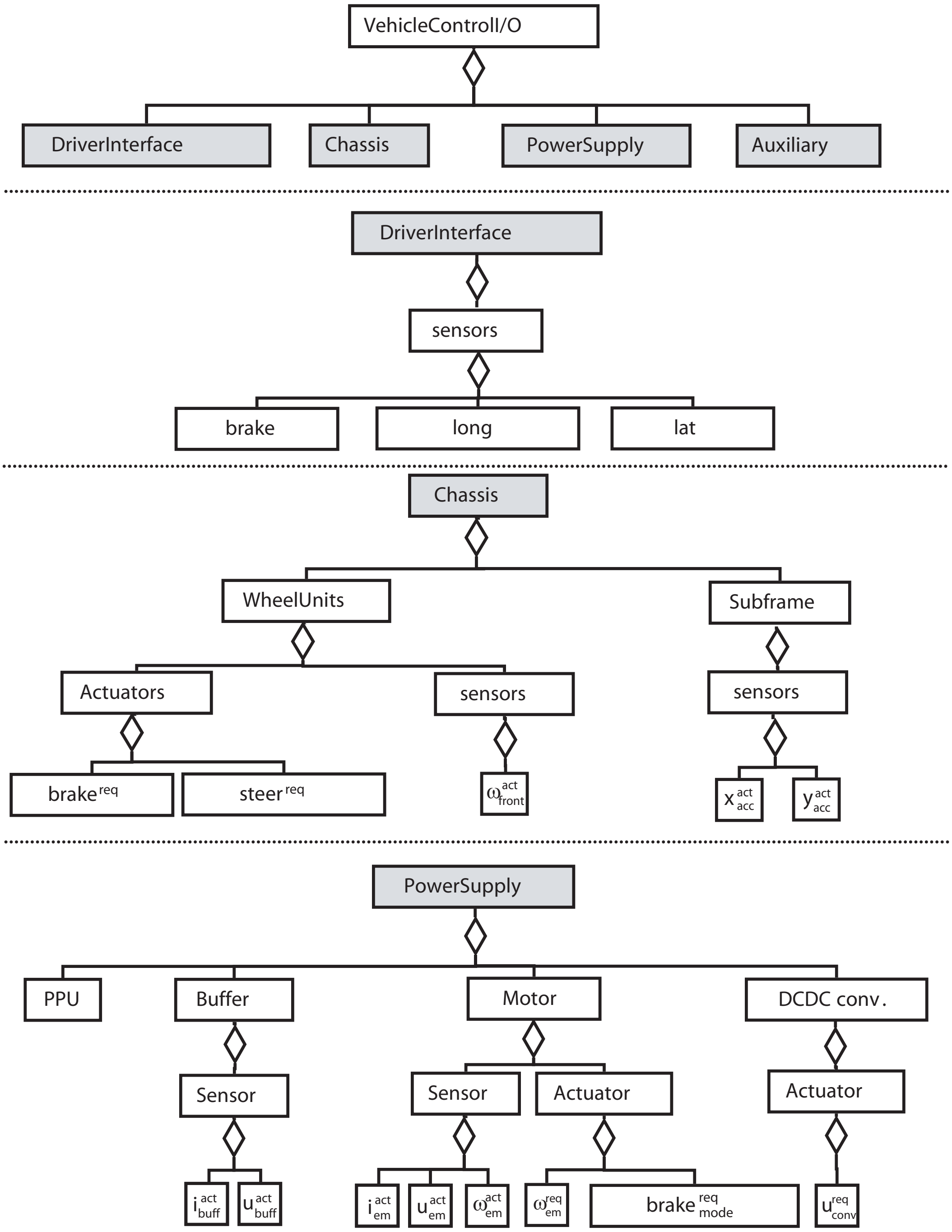


Fig. 9. Static model of the SMC, according to [25]. The diamonds represent under classes. In this figure all estimates, such as actual buffer power, are neglected. It is estimated by $P^{act}_{buff} = u^{act}_{buff} \cdot i^{act}_{buff}$.

wheels. In later versions of the vehicle, estimations of the yaw rate will be implemented. In Table A.5 Ch signals are shown.

PS calculates the actual tractive force, actual power demand from the electric motor, actual buffer power, and SOC level. The used signals are shown in

Table A.5.

Functional level 1 uses signals according to Table A.6. In DIp, the desired vehicle velocity,x_{vel}, steer angle, and level of braking is determined. Then due to the dynamical state of the vehicle the VMC considers if the x_{vel}, steer, and brake needs to be changed. Then in EM an additional desired x_{vel} is calculated which considers the state of the PS, in other words, the amount of energy that is available in PS. It also calculates how the desired power demand should be distributed between PPU and buffer. OD considers the states from EM and VMC, according to Table 1, and gives priority either VMC or EM. OD then makes the arbitration of the desired signal values and convert them to requested signal values for functional level 2.

Due the fact that only one computational node is used in the VSC, the signals are made by defining a 'structure' named 'bus' in the C code. The type definition of the bus looks like
`typedef struct bus( DIF dif; DIP dip; VMC vmc; OD od; CH ch; EM em; PS ps; )BUS;` By this way it easy to distinguish if the signal is a desired value, actual/estimated value, or a request. For example

- `bus.dip.x_vel` is the desired vehicle velocity value from DIp.
- `bus.od.x_vel` is the requested vehicle velocity value from OD.
- `bus.Ch.x_vel` is the actual/estimated vehicle velocity value from Ch.

The functions within VSC are defined as C-functions with pre-defined signals as shown in Table A.5 and Table A.6. The functions are called within a main loop in a certain order as shown in Figure 10.

First DIf function is called which reads the normalised longitudinal, lateral, and brake sensor signals. Then the DIp is called which uses the DIf signals and converts them to desired vehicle speed, steering angle, and a brake level. VMC verifies the previous dynamical state of the vehicle and makes changes to the DIp's desired signals if necessary. Thereafter EM checks the energy state of the vehicle and changes the desired vehicle velocity if necessary and gives a desired buffer power due to different pre-defined rules. The OD, considers the states of both VMC and EM and finalize the orders to the Ch and PS functions.

5 Scaling Parameters -PI Buckingham Theory

The PI Buckingham theorem states that if two similar physical systems in different scale can be described by the same differential equations and can

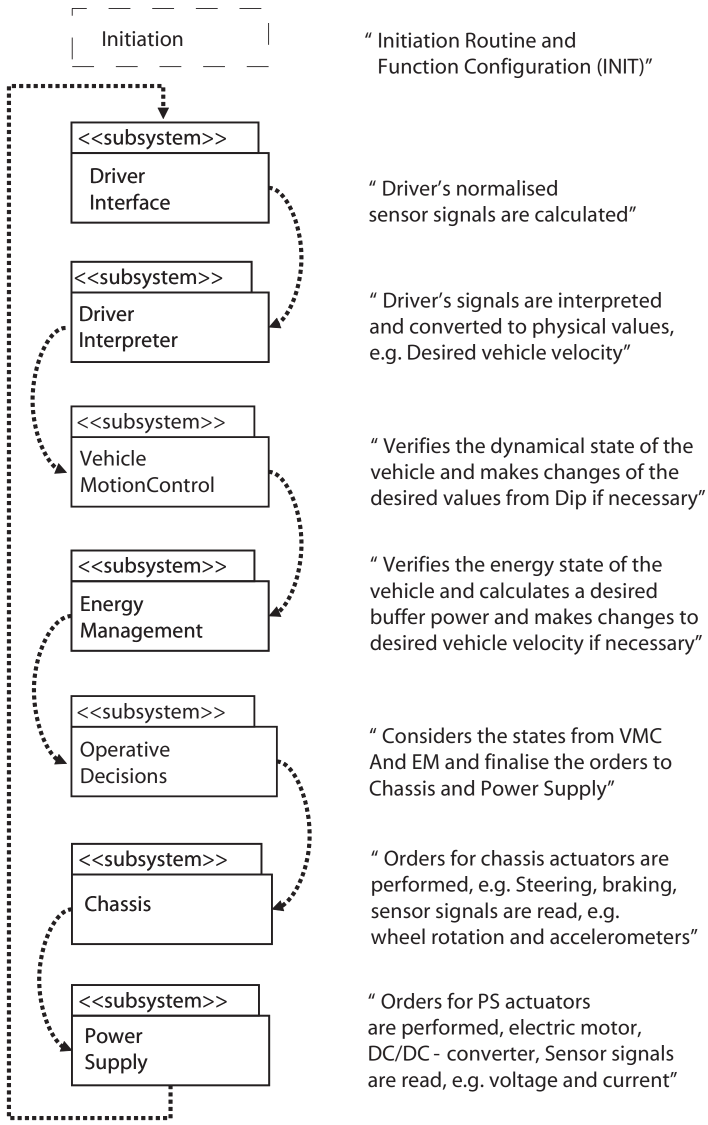


Fig. 10. Program loop used in the Vehicle System Control.

be re-written into dimensionless form by so called PI-parameters, then the solution to the differential equations stays the same if the PI-parameters for the two systems are the same, [16].

This allows scale model vehicles to be used instead of full scale vehicles for investigate the effect of vehicle dynamics and control architecture. Here, in this article, the vehicle's lateral dynamics have been decoupled from the longitudinal propulsion.

A scale model testbed has earlier been used for studying vehicle lateral dynamics and control by Brennan et.al. with good results, [17] and [18]. A linear bicycle model was used for describing the lateral dynamics. The Society of Automotive Engineers standard coordinate system convention is used with z-axis pointing down into the road. The state-space model is

$$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{u} \tag{8}$$

where

$$\mathbf{A} = \begin{bmatrix} -\frac{C_{\alpha f}+C_{\alpha r}}{m x_{vel}} & -x_{vel} - \frac{L_f C_{\alpha f} - L_r C_{\alpha r}}{m x_{vel}} \\ -\frac{L_f C_{\alpha f} - L_r C_{\alpha r}}{I_z x_{vel}} & -\frac{{L_f}^2 C_{\alpha f} + {L_r}^2 C_{\alpha r}}{I_z x_{vel}} \end{bmatrix},$$

$$\mathbf{B} = \begin{bmatrix} -\frac{C_{\alpha f}}{m} & 0 \\ -\frac{L_f C_{\alpha f}}{I_z} & -\frac{T_k}{I f_z R_w} \end{bmatrix},$$

$\mathbf{x} = \begin{bmatrix} y_{vel} & \dot{\psi} \end{bmatrix}^T$, $\mathbf{u} = \begin{bmatrix} \delta_f & \Delta\tau \end{bmatrix}$. The y_{vel} is the lateral velocity and $\dot{\psi}$ is the yaw rate. The δ_f and $\Delta\tau$ are the front steering angle and the differential steering torque input.

Equation 8 assumes constant longitudinal velocity, x_{vel}. The vehicle's lateral position y, is then a function primarily dependent on the following parameters

$$y = (m, I_z, x_{vel}, L_f, L_r, T_w, R_w, C_{\alpha f}, C_{\alpha r}) \tag{9}$$

Where m is the vehicle mass, I_z yaw inertia, L_f length from front axle to centre of gravity, L_r length from rear axle to centre of gravity, T_k track of the vehicle, R_w wheel radius, $C_{\alpha f}$ Cornering stiffnes front, and $C_{\alpha r}$ Cornering stiffnes rear.

The derivation of the PI parameters are explained in [17] and [18]. A summary of the PI groups is as follows

$$\Pi_1 = \frac{L_f}{L}, \Pi_2 = \frac{L_r}{L}, \Pi_3 = \frac{C_{\alpha f}}{m \cdot x_{vel}^2}, \Pi_4 = \frac{C_{\alpha f}}{m \cdot x_{vel}^2}, \Pi_5 = \frac{I_z}{m \cdot L^2} \tag{10}$$

The first and second PI group relates to vehicle dimensions and also mass balance. The third and fourth are coupled to front and rear axle cornering stiffness who have the longitudinal velocity present. The fifth PI parameter is about the yaw inertia.

The dynamic similitude comparison of the current configuration of the SMC showed that it is close to a sports car, a summary is given in Table 2, further details will be found in [23].

The characteristic polynomial for equation 8 is as follows

$$s^2+\left(\frac{C_{\alpha f}+C_{\alpha r}}{m x_{vel}}+\frac{{L_f}^2 C_{\alpha f}+{L_r}^2 C_{\alpha r}}{I_z x_{vel}}\right)s+\frac{C_{\alpha f}C_{\alpha r}(L_f+L_r)^2}{m I_z {x_{vel}}^2}-\frac{L_f C_{\alpha f}-L_r C_{\alpha r}}{I_z}=0. \tag{11}$$

If one nondimensionalise equation 11 by using the PI parameters the characteristic equation can be rewritten as

$$s^{*2}+\left((\Pi_3+\Pi_4)+\frac{1}{\Pi_5}\left(\Pi_1^2\Pi_3+\Pi_2^2\Pi_5\right)\right)s^*+\frac{1}{\Pi_5}\left(\Pi_3\Pi_4-\Pi_1\Pi_3+\Pi_2\Pi_4\right)=0 \tag{12}$$

Due to that the PI-parameters match fairly well the normalized pole locations between the two systems will be similar.

	SMC	Sports Car
Constant x_{vel} (m/s)	4	25
Π_1	0.66	0.5
Π_2	0.34	0.5
Π_3^a	0.13	0.35
Π_4^a	0.25	0.35
Π_5	0.13	0.26
Poles	-0.76, -0.63	-0.981, -0.59

Table 2
Dynamic similitude comparison. [a]These parameters can be matched by varying the velocity x_{vel}.

Another aspect is the longitudinal propulsion of the SMC. What type of full size configuration can the SMC be compared with? Especially when the electric motor is scaled to full size? Two longitudinal performance parameters, maximum velocity and maximum acceleration of the vehicle, are considered here. These two parameters will give a fairly good idea what type of configuration the full size vehicle would be.

Here, a study of the longitudinal dynamics of the vehicle is made, by setting up following differential equation for the longitudinal acceleration x_{acc}

$$F_{tr} - \sum F_{resistance} = m \cdot x_{acc} \tag{13}$$

Where F_{tr} is the driving force generated, and $F_{resistance}$ is the resistance forces.

$$\sum F_{resistance} = F_{air} + F_{roll} + F_{slope} \tag{14}$$

Where the resistance force contains three forces: drag resistance $F_{air} = 0.5C_d \cdot A \cdot \rho x_{vel}^2$ with C_d drag coefficient, A frontal area of the vehicle, and ρ air density. Roll resistance $F_{roll} = f_{roll} \cdot m \cdot g$ with f_{roll} rolling resistance coefficient. Finally the slope resistance force $F_{slope} = m \cdot g \cdot sin(\alpha)$ with slope angle α.

If we now take the powertrain into consideration and neglect the inertia we have

$$T_{tr} = T_{em} \cdot i_1 \cdot i_2 \cdot i_3 \tag{15}$$

Where T_{tr} is the driving torque at the wheels and T_{em} is the torque generated by the electric motor, and i_i are the gear ratio's in the drive train.

If the inertia of the wheels are also neglected one can set up the following relationship $T_{tr} = F_{tr} \cdot R_w$ and assume that all the driving torque is transformed to driving force F_{tr}. Then we can write the driving force as

$$F_{tr} = \frac{T_{tr}}{R_w} = \frac{T_{em} \cdot i_1 \cdot i_2 \cdot i_3}{R_w} \tag{16}$$

With the following state $x = x_{vel}(t)$ the system can be described by the following non-linear differential equation

$$\frac{dx}{dt} = \frac{-0.5C_d \cdot A \cdot \rho}{m}x^2 + \frac{T_{em} \cdot i_1 \cdot i_2 \cdot i_3}{R_w \cdot m} - f_{roll} \cdot g - g \cdot \sin(\alpha)$$
$$T_{em} = Min\left(T_{em,cont}, \frac{P_{em,cont}}{\omega(t)}\right)$$
$$\omega(t) \in \Big[0 \ \omega_{max}\Big]$$

where the rotational speed $\omega(t)$ of the electric motor is the input signal. The continous torque $T_{em,cont}$, and continous power $P_{em,cont}$ of the electric motor are used to define its characteristics.

The vehicle's state x that affect the vehicle's longitudinal motion is then a function primarily dependent of following n=5 parameters

$$x = (m, T_{em,cont}, P_{em,cont}, A, \omega_{max}) \tag{17}$$

Where the unitless constants are neglected C_d, i_i, f_{roll}, and α.

The following fundamental k=3 parameters are selected m, A, and ω_{max}. To non-dimensionalise the differential equations one needs to define j=n-k=5-3=2 PI-groups.

The first PI-group is defined by

$$\begin{aligned}
&\Pi_1 = m^{\alpha_1} A^{\beta_1} \omega_{max}^{\gamma_1} T_{em,cont} = [I] \\
&\Pi_1 = N^{\alpha_1} s^{2\alpha_1} m^{-\alpha_1} m^{2\beta_1} s^{-\gamma_1} Nm = [I] \\
&[N] \qquad \alpha_1 + 1 = 0 \\
&[m] \qquad -\alpha_1 + 2\beta_1 + 1 = 0 \\
&[s] \qquad 2\alpha_1 + \gamma_1 = 0
\end{aligned} \tag{18}$$

Which gives

$$\Pi_1 = \frac{T_{em,cont}}{m \cdot A \cdot \omega_{max}^2} \tag{19}$$

In similar manner the second PI group is defined as

$$\Pi_2 = \frac{P_{em,cont}}{m \cdot A \cdot \omega_{max}^3} \tag{20}$$

The PI-parameters for the SMC is

$$\Pi_1 = \frac{0.32}{16 \cdot 0.0804 \cdot 314.159^2} = 2.52 \cdot 10^{-6}$$
$$\Pi_2 = \frac{80}{16 \cdot 0.0804 \cdot 314.159^3} = 2.00 \cdot 10^{-6} \tag{21}$$

By assuming that the full scale vehicle has the same PI-parameters one can derive the continous torque and power of the electric motor. The full scale vehicle has the mass m=1300 kg, frontal area A=1.7 m^2, and that the electric motor has the same max rotational speed ω_{max}=314.159 rad/s. Using equation 19 gives a continuous torque of $T_{em,cont}$=550 Nm and by using equation 20 gives the continuous power of $P_{em,cont}$=137 kW.

Now one can calculate an approximate maximum vehicle speed $x_{vel,max}$ by studying the continuous output power and how much resistance forces it can overcome on a horisontal surface

$$P_{em,cont} = (F_{roll} + F_{air}) \cdot x_{vel,max} = \left(f_{roll} \cdot m \cdot g + 0.5C_d \cdot A \cdot \rho \cdot x_{vel,max}^2\right) \cdot x_{vel,max}. \tag{22}$$

A maximum vehicle velocity $x_{vel,max}$=267 km/h can be calculated by equation 22 with following assumptions f_{roll}=0.012, g=9.81 m/s^2, C_d=0.3, and air density ρ=1.202 kg/m^3. However if the same fixed gear ratio $(i_1 \cdot i_2 \cdot i_3)$ is used in the full scale vehicle, the maximum velocity will be limited by the maximum rotational speed of the electric motor in this case the maximum vehicle velocity would be 81 km/h. But most likely some kind of gearbox e.g. continuously variable transmission would be used to optimise the use of the electric motor.

The maximum acceleration for the vehicle can be estimated by the following equation

$$x_{acc,max} = \frac{T_{em,cont} \cdot i_1 \cdot i_2 \cdot i_3}{R_w \cdot m} \tag{23}$$

with a wheel radius R_w=0.3 m, gives $x_{acc,max}$=5.88 m/s^2. If the max acceleration can be kept constant to 100 km/h this speed would be reached in 4.7 s.

6 Test Run of SMC

The SMC was tested on how it performs with the implemented functions within VSC. See section 2.2 for further details about the used algorithms within VMC, EM, and OD.

The steering, acceleration, and braking works sufficiently and relatively fast. The traction controller within VMC works when the car is subjected to slippery surfaces. If the traction controller within VMC is closed down skidding is hard to avoid on slippery surfaces, due to high the traction force generated by the electric motor. Skidding is not a problem when driving on asphalt, see further in Section 6.1.

The maximum target speed is of 4 m/s second was achieved during the test runs. Also the maximum target acceleration 1.5 m/s^2 from 0 to 3 m/s was obtained. These targets were used during the design phase of the car.

The operative time of the vehicle is approximately 30 to 50 min. Depending on how the vehicle is driven. This time agrees quite well with the simulated time of 30 min.

The logging from a Drive Cycle Test was performed and is further discussed in Section 6.2.

6.1 Traction Control Testing

A simple traction controller is located within VMC function, see Section 2.2.2. A max acceleration test was performed on a surface with low friction, close to ice conditions. During the acceleration the VMC signals that a critical state is achieved, and VMC reduces the desired velocity from DIp. Two test runs were performed. One with the VMC function activated, and the other was with the VMC function deactivated during the whole acceleration, see Figure 11. The continuous lines shows the front wheel rotational speed and the dashed shows the rear wheel rotational speed. The implemented traction controller within VMC improves the acceleration performance, see Figure 11. The time to reach the longitudinal velocity of 1.5 m/s or 25 rad/s in the front wheels was 2.65 s without the controller, and 2.2 s with the controller activated.

The simple traction control implemented in VMC and the state controller within OD worked overall as desired. However, due to the fact that different rotational speed sensors, with different accuracy, were used for front and rear wheels introduced a numerical error on the slip estimation which affected the results of the implemented traction controller compared to the simulated

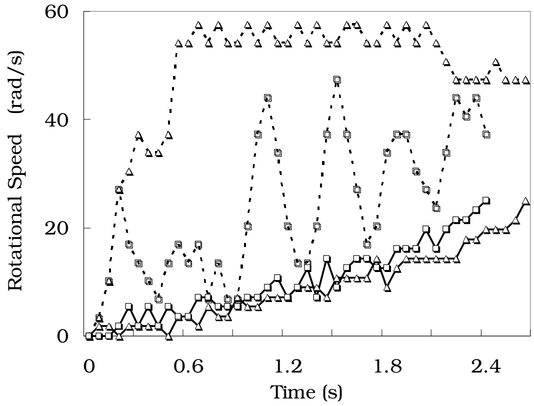


Fig. 11. Front (continuous line) and rear (dashed line) wheels, with VMC function activated (squares) and without (triangles).

results. Further details can be found in [26].

6.2 Drive Cycle Testing

The SMC was evaluated during a drive cycle test, driven indoor on concrete. This tested the simple energy management algorithm that is located within EM function, see Section 2.2.3.

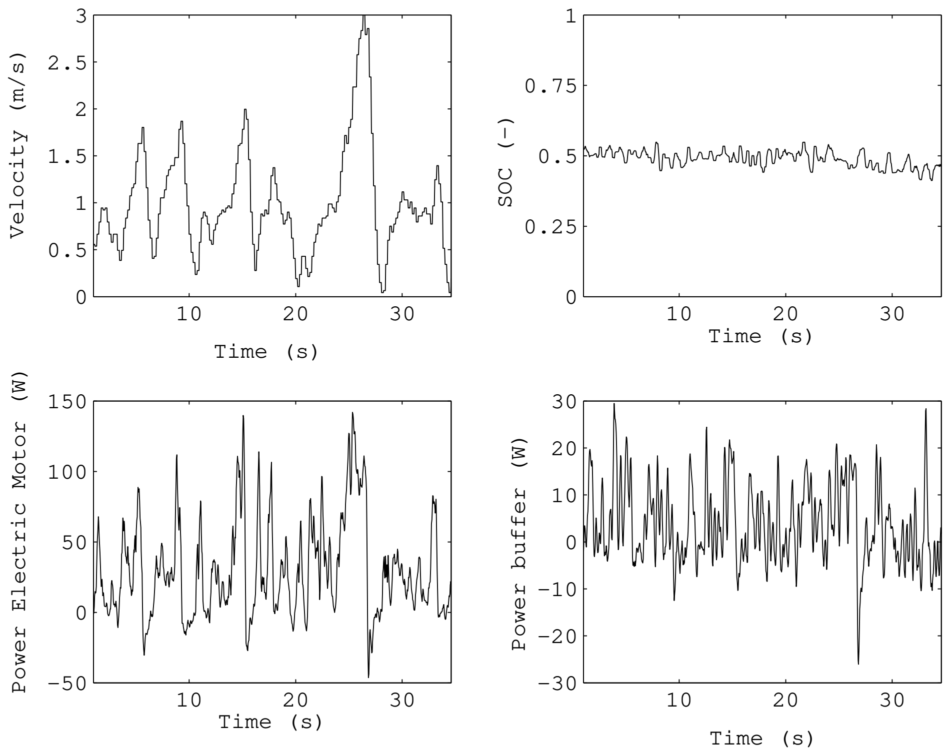


Fig. 12. In door drive cycle test with the SMC.

The drive cycle is shown in the upper left plot in Figure 12. The SOC is shown in the upper right plot in Figure 12. Finally the power demand on the electric motor and buffer power are shown in the lower left and right plot respectively in Figure 12.

The SOC has only small changes due to that low buffer power is used and the total available buffer energy is high compared to a single acceleration. One can also see that electric motor has a negative power demand during deceleration especially during the highest deceleration at time 26.8 s. The regenerated energy is stored in the buffer. Another interesting observation was that the used DCDC converter can only handle power flows of maximum 40 W out from the buffer, but several times higher into the buffer. This is due to implemented software restrictions by the supplier of the used DC/DC converter.

7 Summary and Future Work

This paper describes a Reusable VSC that was implemented and tested in a SMC. The current hardware configuration of the SMC include a battery as a PPU, supercapacitors as a buffer, DC/DC converter, and a electric motor that can also be used as a generator. The car is RC-controlled and has the RC-receiver connected to the DSP card. The car is front steered, and mechanical brakes are applied on the front wheels. This is managed by two electric servos. The car is rear wheel driven.

The suggested VSC code was easy to use with the current hardware configuration of the SMC. It has also has been proven to be easy to add and change functionality within the different functions such as DIp, VMC, EM, and OD. Also when the hardware was changed, e.g. wheel rotation sensors, only the affected subsystem, in this case Ch needed to be changed.

Logged data from the test run shows that the PS and EM of the SMC works as intended. The test run also verifies that the computer model of the SMC reasonable well explains the signals.

According to the dimensional analysis the SMC corresponds to a sports car with a top speed of 267 km/h and maximum acceleration of 5.88 m/s^2.

There are several future opportunities for the SMC. Following objectives are suggested:

- Implement a real FC stack will be used as PPU. The PPU will have a local controller to supervise the FC stack.
- Use dimensional analysis to study behaviour of normal passenger cars,

SUV's, pick up's, and trucks by adjusting the PI parameters of the SMC.
- Change the hardware configuration of the Ch and PS and study the modularity within the suggested VSC Code.
- Compare different control structures and how they affect the overall performance of the vehicle.
- Test different algorithms within VMC and EM and study the behaviour of car.

References

[1] J. C. Burba, Consideration of Control System Architecture in Electric and Hybrid Electric Vehicles, Electric Vehicle Symposium 18, Berlin, 2001.

[2] M. Gäfvert, Topics in Modeling, Control, and Implementation in Automotive Systems, Dep. of Automatic Control, Lund Institute of Technology, Sweden, 2003.

[3] E. Coelingh, P. Chaumette, and Mats Andersson. Open-Interface Definitions for Automotive Systems, Application to a Brake by Wire System, SAE 2002-01-0267.

[4] A. M. Phillips, Functional Decomposition in a Vehicle Control Systems, American Control Conference, Anchorage, 2002.

[5] L. Laine and J. Andreasson, Modelling of Generic Hybrid Electric Vehicles, In Proceedings of the 3^{rd} International Modelica Conference, Linkping, Sweden, November 2003.

[6] L. Laine and J. Andreasson, Generic Control Architecture applied to a Hybrid Electric Sports Utility Vehicle, in Proceedings of the Electric Vehicle Symposium, (EVS 20), Los Angeles, USA, November 2003.

[7] J. Andreasson, L. Laine, and J. Fredriksson, Evaluation of a Generic Vehicle Control Architecture, in Proceedings of the 30^{th} Congress of the International Federation of Automotive Engineering Societies, (FISITA'30), Barcelona, Spain, May 2004.

[8] L. Laine and J. Andreasson, Reusable Functional Partitioning of Tractive Force Actuators Applied on a Parallel Hybrid Electric Vehicle, in Proceedings of the 7^{th} International Symposium on Advanced Vehicle Control, (AVEC'04), Arnhem, Netherlands, August 2004.

[9] P. Hoblet, R.T.Jr. O'Brien, Piepmeier J.A., Scale-Model Vehicle Analysis for the design of a Steering Controller, Proceedings of the 35th Southeastern Symposium on System Theory, 2003.

[10] M. Shinoa, M. Nagaib, Independent wheel torque control of small-scale electric vehicle for handling and stability improvement, JSAE Review 24, 2003.

[11] R. S. Woodley, L. Acar, Neural Network Based Control for a Backward Maneuvering Trailer Truck, Proc. Of the 37th IEEE Conference on Decision and Control, Florida, USA, 1998.

[12] J. Pereira, J. B. Bowles, A Comparison of PID and Fuzzy Control of a Model Car, Electrical and Computer Engineering, University of South Carolina, Columbia, USA, 1993.

[13] K. Nishimori, S. Hirikawa, K. Fujimura, H. Tokutaka, S. Kishida, H. Hiraga, N. Ishihara, Automatic Tuning Method of Membership Functions in Simulations of Driving Control of a Model Car, Proc. Of International Joint Conference on Neural Networks, 1993.

[14] S.-J., Chang, C.-W. Cheng, T.-H. S. Li, Design and Implementation of Fuzzy Garage-Parking Control for a PC-based Model Car, National Cheng Kung University, Taiwan, 1996.

[15] SPRU024e User Manual for TMS320LF2407A Evaluation Module, Texas Instruments Inc., 2004.

[16] E. Buckingham, On Physical similar systems: Illustrations of the use of dimensional equations, Physical Rev, vol. 4, pp. 345-276, 1914.

[17] S. Brennan, A. Alleyne, The Illinois Rodway Simulator: A Mechatronic Testbed for vehicle Dynamics and Control, IEEE/ASME Transactions on Mechatronics, vol. 5, no. 4, 2000.

[18] S. Brennan, A. Alleyne, Using a Scale Testbed: Controller Design and Evaluation, IEEE Control Systems Magazine, vol. 21, June 2001.

[19] FG Modellsport, www.fg-modellsport-gmbh.de, Winterbach, Germany, 2004.

[20] J. Hellgren, L. Laine, J. Sjöberg, M. Rönnberg, D. Gomes, and A. Abudings, Systematic Design and Development of a Hybrid Electric Scale Model Car, Technical Report, Dept. of Machine and Vehicle Systems, Chalmers University of Technology, Sweden, July 2004.

[21] J. Hellgren, Conceptual Powertrain Design and Energy Management of Hybrid Electric Vehicles using Evolutionary Algorithms and Sugeno Fuzzy Logic, Submitted to IEEE Transaction on Vehicular Technology, 2004.

[22] M. Rönnberg, Implementation of a Control System for a Scaled Series Hybrid Electric Vehicle, Master thesis report, Dep. of Machine Vehicle Systems, Chalmers University of Technology, Sweden, 2004.

[23] D. E. B. Gomes, Design and Development of a Hybrid Electric Scale Model Car, master thesis report, Dep. of Machine Vehicle Systems, Chalmers University of Technology, Sweden, 2004.

[24] C. K. Song, Michael Uchanski, and J. K. Hedrick, Vehicle Speed Estimation Using Accelerometer and Wheel Speed Measurements, SAE 2002-01-2229, 2002.

[25] H. Gomaa, Designing Concurrent, Distributed, and Real-Time Applications with UML, Pearson Education Corporate, 2001.

[26] P. Labenda, Development, Implementation and Testing of a Traction Control Algorithm for Hybrid Electric Scale Model Car, Dep. of Signals and Systems, Chalmers University of Technology, Sweden, 2005.

A Appendix -Nomenclature and Tables

Abbreviations

A	Actuator
Aux	Auxiliary Systems
Ch	Chassis
DIf	Driver Interface
DIp	Driver Interpreter
DSP	Digital Signal Processor
em	electric motor
EM	Energy Management
HEV	Hybrid Electric Vehicle
Mc	Mechanical Connection
OD	Operative Decisions
PPU	Primary Power Unit
PS	Power Supply
RC	Remote Control
S	Sensor
SMC	Scale Model Car
SOC	State of Charge
VMC	Vehicle Motion Control
VSC	Vehicle System Control
WU	Wheel Unit

Notations

A	Frontal Area [m^2]
$brake$	brake level [0,1]
i	Gear ratio
i_x^{act}	Actual current in components x=em, buff, and conv [A]
Π_i	PI parameter [-]
m	Vehicle mass [kg]
F_{air}	Air resistance force [N]
$F_{resistance}$	Resistance force [N]
F_{roll}	Rolling resistance force [N]
F_{slope}	Slope resistance force [N]
F_{tr}	Traction force on the wheels [N]
P_{buff}	Buffer power [W]
P_{PPU}	Primary Power Unit power [W]
P_{em}	Power electric motor [W]
$P_{em,cont}$	Continous power electric motor [W]
ω_{front}	Speed front wheels [rad/s]
ω_{rear}	Speed rear wheels [rad/s]
ω	Speed electric motor [rad/s]
λ_{front}	Slip front wheels [0,1]
λ_{rear}	Slip rear Wheels [0,1]
lat	Lateral motion [0,1]
$long$	Longitudinal motion [0,1]
T_{em}	Torque electric motor [Nm]
$T_{em,cont}$	Continous torque electric motor [Nm]
T_{tr}	Traction torque on the wheels [Nm]
SOC	State of Charge of Buffer [0,1]
SOC_{ref}	State of Charge reference of Buffer [0,1]
$state_i$	State of VMC or EM [0/1]
$steer$	Steering angle [deg]
u_x^{act}	Actual voltage in components x=em, buff, and conv [V]
x_{vel}	Longitudinal vehicle velocity [m/s]
x_{acc}	Longitudinal vehicle acceleration [m/s^2]
y_{vel}	Lateral vehicle velocity [m/s]
y_{acc}	Lateral vehicle acceleration [m/s^2]

Vehicle mass, m	16 kg
c.o.g. to front axle, L_f	0.3499 m
c.o.g. to rear axle, L_r	0.1813 m
Wheel radius, R_w	0.06 m
Track width, T_w	0.120 m
Cornering stiffness front, $C_{\alpha f}$	96 N/rad
Cornering stiffness rear, $C_{\alpha r}$	187 N/rad
Yaw inertia, I_z	0.6 kgm^2
Air drag Coeff. C_d	0.28
Frontal Area A	0.0804 m^2

Table A.1
Summary of vehicle dynamics parameters.

Electric Motor	
Manufacturer	Östergrens elemotor
Model	BLDC3, L4495704
Mass	1 kg
Efficiency	0.6-0.8
Max Power P_{max}	230 W
Continous Power P_{Cont}	80 W
Max Torque T_{max}	0.98 Nm
Continous Torque T_{Cont}	0.32 Nm
Torque Constant, T_k	0.05 Nm/A
Max speed, Ω_{max}	3000 rpm
Resistance, R	0.6 Ohm
Induction, L	1.6 mH
Voltage	24 V
Length, Diameter	11, 8 cm
driveline	
Gear ratio of cogged belt	2
Gear ratio between cogged belt axle and differential	2.087
Gear ratio differential	1

Table A.2
Characteristics of the electric motor and gear ratio of drive line.

Manufacturer Model	Maxwell BCAP0350A01
Mass	0.057 kg
Efficiency	0.9-0.99
Max continuous current	40 A
Capacitance	350 F
Energy	1100 J
Rated, Surge Voltage	2.5, 3.8 V
Resistance	3.2 mΩ
Length, Diameter	6.15, 3.3 cm

Table A.3
Characteristics of one super capacitor.

Manufacturer Model	ZAPI 4Q
Mass	1 kg
Efficiency	0.85
Max current	70 A
Continous current	30 A
Input Voltage	24 V
Output Voltage	0-24 V
Length, Width, Height	12, 14, 5 cm

Table A.4
Characteristics of the DC/DC converter.

Signal:	Description:	Values/Unit:
Driver Interface		
long	Sensed longitudinal motion	[-0.3,1]
lat	Sensed lateral motion	[-1,1]
brake	Sensed reg.brake or mech.brake	[0,1]
Chassis		
x_{vel}	Actual long. velocity	[m/s]
x_{acc}	Actual long. acceleration	[m/s^2]
y_{vel}	Actual lat. velocity	[m/s]
y_{acc}	Actual lat. acceleration	[m/s^2]
ω_{front}	Actual speed front WUs	[rad/s]
ω_{rear}	Actual speed rear WUs	[rad/s]
λ_{front}	Actual slip front WUs	[-]
λ_{rear}	Actual slip rear WUs	[-]
Power Supply		
F_{tr}	Actual traction force	[N]
P_{em}	Actual power Elect. Mach.	[W]
P_{buff}	Actual buffer power	[W]
SOC	State of Charge	[0-1]
fuel	Amount of fuel	[0-1]

Table A.5
Interface signals from functional level 2.

Signal:	Description:	Values/Unit:
Driver Interpreter		
x_{vel}	Desired long. velocity	[m/s]
$steer$	Desired steer angle	[degrees]
$brake$	Desired brake action	[0,1]
Vehicle Motion Control		
x_{vel}	Desired longitudinal velocity	[m/s]
$steer$	Desired steer angle	[degrees]
$brake$	Desired brake action	[0,1]
$state$	Dynamical state 0=ok, 1=nok	[0/1]
Energy Management		
x_{vel}	Desired long. velocity	[m/s]
P_{ppu}	Desired power from PPU	[W]
P_{buff}	Desired power from Buffer	[W]
F_{tr}	Desired traction force	[N]
state	Energy state 0=ok, 1=nok	[0/1]
Operative Decisions		
steer	Requested steer angle	[degrees]
x_{vel}	Requested longitudinal velocity	[m/s]
P_{ppu}	Requested power from PPU	[W]
P_{buff}	Requested power from Buffer	[W]
F_{tr}	Requested traction force	[N]
$brake$	Requested brake action	[0,1]
$brake_{mode}$	Regenerative brake or not	[1/0]

Table A.6
Interface signals from functional level 1.