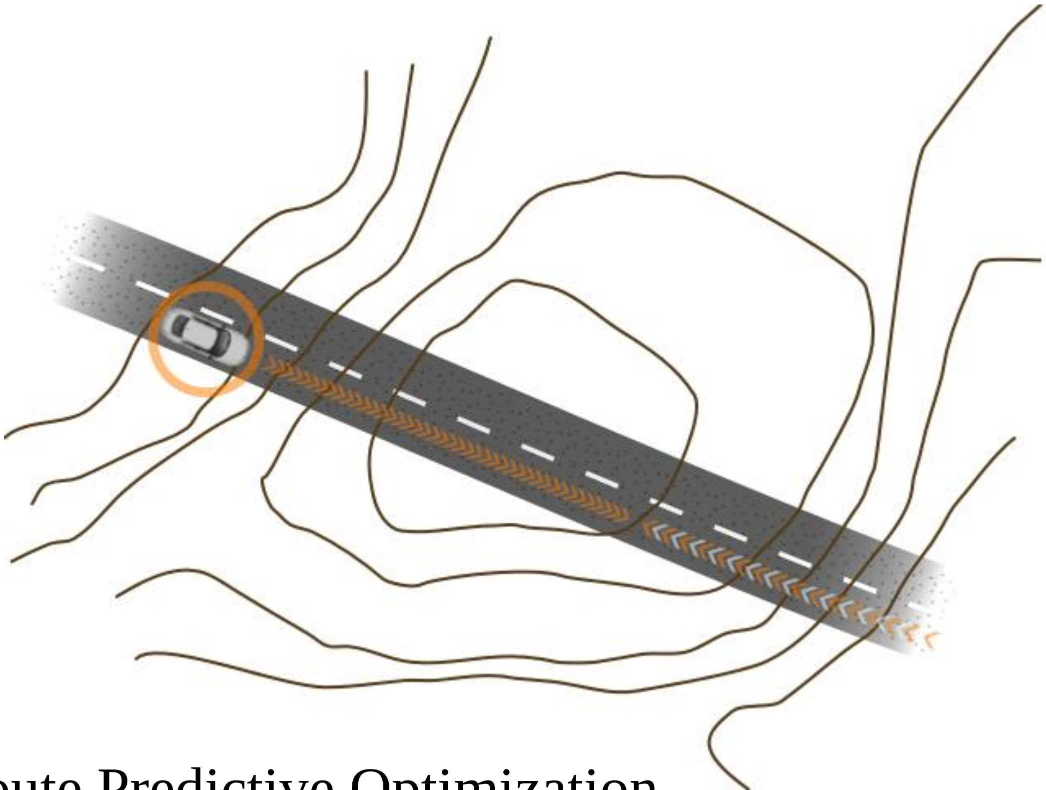
CHALMERS



Route Predictive Optimization of State-of-Charge Reference Signal

Extension to a Hybrid Control Unit

Master of Science Thesis in the Master Degree Programme's: Intelligent System Design and Systems, Control & Mechatronics

OLOF APPELKVIST
PHILIP GEBEL

Department of Signals and Systems
Division of Control
CHALMERS UNIVERSITY OF TECHNOLOGY
Göteborg, Sweden, 2010

Master's Thesis No. EX077/2010

Abstract

A lot of research has been spent on investigating if knowledge of the upcoming route can improve fuel economy in hybrid electric vehicles. With navigational systems becoming increasingly integrated in vehicles, new and exciting possibilities emerge. Within the project, summarized in this report, a functioning vehicle model was developed, which was connected to an existing hybrid control system. The scope of the project was to find a strategy to exploit navigational information to predict the upcoming road attributes, and to analyze how different choices would influence the fuel consumption. A system with a position dependant horizon was created and an algorithm to influence the existing control system was developed. Results show that it is possible to decrease fuel consumption by planning the use of the electric motor based on upcoming road segments that hold energy.

Keywords: Hybrid Electric Vehicle, Look Ahead, Route Prediction, State-of-Charge reference signal

Preface

This report summarizes what was found during a 30hp master thesis project, carried out by the authors during the spring/summer of 2010. It is the final part of the Master of Science programmes in Intelligent System Design and Systems, Control & Mechatronics, at Chalmers University of technology in Göteborg, Sweden. The project was executed in the hybrid controls group, at SAAB Powertrain AB in Trollhättan, Sweden.

Acknowledgement

We would like to thank our supervisor and examiner at Chalmers, Prof. Jonas Fredriksson, at the department of Mechatronics and our main contacts at SAAB Powertrain AB, Lars Johansson and Magnus Berndtsson for a welcoming and encouraging support. We have also been guided by Dr. Lars Johannesson, department of control at Chalmers as well as many helpful employees at SAAB Powertrain AB, especially Dr. John Lord, Magnus Källvik, Linus Andersson and Mattias Hansson. Thank you all!

Olof Appelkvist and Philip Gebel
Göteborg, September 2010

Contents

List of Figures

List of Tables

1 Introduction

Hybrid electric vehicles (HEV) have left the stage of being prototypes and experimental visions of the future, to become viable options for customers at the large automobile market and available from a growing number of manufactures, e.g. Toyota, Honda and Ford. In a HEV there exist, as the term implies, at least one electric propulsion system apart from other power source(s), usually an internal combustion engine (ICE). There exist different options to combine the electric motor (EM) and the ICE, and in this report a parallel hybrid with electric rear axis has been studied.

In order to exploit both (or all) propulsion units in the car, a control system analyses, among other things, the driver demanded torque, present output to the wheels, the State-of-Charge (SoC) of the batteries, and calculates the fuel optimal torque split factor between the different systems. In present applications these controls only subject the current time instance when generating the split factor, so called online-optimization.

A number of studies have been made in order to investigate if it is possible to reduce fuel consumption when information about the route is given. Such information could include length, speed profile and topography of the route, information that could be extracted from the route planning system. A more thorough review of found literature in the field can be found in chapter 2.3.

The paper that became most relevant for this project is [1], where an optimized reference signal for SoC is proposed, based on calculations of future energy demand given road topography, expected vehicle velocity and the position of the vehicle. The most important factor for choosing this approach was the given condition (more described in chapter 3) with a functioning control system developed at SAAB Powertrain AB.

1.1 Purpose

The main purpose of the thesis project was to investigate if current hybrid control system could be improved with an extending algorithm that processes and analyzes future route information, gathered from a GPS/navigation-system. An exemplifying algorithm should be developed and simulations should be analyzed with fuel consumption in focus.

Further the project should aim at investigate different kinds of strategies, found in academia and examine what kind of information from a GPS/navigation-system is needed.

1.2 Objective

A number of goals have been identified to be able to fulfil the purposes of project. The following list is an overview to recognize the different outcomes of the thesis project work.

- Conduct a literature study on the subject of on-board route information assistance for hybrid electric vehicle control.
- Model the designated vehicle, with the existing control algorithm in MatLab/Simulink.
- Define the necessary additional information needed to extend the existing online-optimization.
- Extend the current control algorithm with a look-ahead algorithm.
- Use the model environment to calculate fuel consumption and make a comparison against the present control strategy, given specified drive-cycles.
- Compare the results from the model environment with the fuel saving potential expected from the literature study.

1.3 Limitations

The main restriction in the project was that present control algorithm should be kept intact. Only minor modifications were allowed in order to adapt it to new drivecycles and to make it respond to the developed route information algorithm. Parametric configurations were also allowed to make the developed plant model simulate correct.

The study is limited to only cover the designated vehicle design, further described in section 3.3.

An early expressed guideline for the project was to keep developments as simple as possible.

The assumed GPS/route planning system were treated as inputs to the system only. No models for this system were developed, with the exception of extracting information from a recorded drivecycle with topography and travelled distance. The given longitudinal and latitudinal position of the car were assumed to be the same as the travelled distance, though only presented in one dimension. A consequence of using the data in this manner was the assumption the road being straight in the X-Y-plane. Since information about the altitude existed, the present and future positions of the vehicle were given in two dimensions, X and Z, but this assumption could be generalized to three dimension if $position = [X, 0, Z]$. This implied that no deviation, noise or latency were assumed for the positioning system, or that these were already captured in the recorded data.

The development, tests and verifications were preceded in desktop simulation environment only, using MatLab/Simulink. No further implementations in real vehicles or more advanced drive simulators has been covered.

In-depth details about the control system has not been covered in the report, due to secrecy agreements.

1.4 Method

In order to gain insight into the technology of hybrid vehicles and look-ahead strategies, a literature study was performed. The authors had to get familiarized with the simulation environment used at SAAB Powertrain AB. A broad perspective of the subject was used in the beginning, which narrowed down to a model concept. Later conclusions with wider perspectives were stressed.

The general strategy when creating models was based on iteration, simple models were made and when functionality was satisfying, the complexity increased in order to reach satisfying results.

In an early phase of the project evaluation and decision steps were made in a planning report to be able to overlook all important progressions of the project.

2 Background Theory

2.1 Hybrid Electric Vehicle

In contrast to most common vehicles propelled by one mover, e.g. a gasoline engine or a diesel engine, the hybrid electric vehicle (HEV) uses two or more power sources in combination with appropriate power converters. There are other types of hybrid vehicles than HEV, using e.g. pneumatics or hydraulics, but the combination of electric energy storage and internal combustion engine (ICE), is usually what is meant by the term hybrid vehicle.

Hybrid electric vehicles can be configured in different ways and with different choices regarding the ICE and EM, but generally there exist one ICE capable of delivering most of the requested torque to wheels, from the driver. In figure 2.1the parallel hybrid configuration is showed. The EM is dimensioned to be able to assist the ICE in accelerations and even drive the car when conditions allow, hence allowing the ICE to shut down. The same applies for idling phases where the auxillary power demand can be covered by the battery, again allowing the ICE to shut off.

There also exists an electric energy buffer, e.g. electrochemical batteries, which can deliver energy to the EM but also store energy delivered from the EM. The EM thus works as a generator, making it possible to recuperate energy if possible or necessary. In some HEVs there exists more than one EM where the functionalities are more specific towards propulsion and generating, where the different EMs can be of different types and with different electric specifications. The torque distribution between the engine and the electric motor is determined by a hybrid control unit (HCU), see section 3.1.

The combined configuration brings the possibility to e.g. improve fuel economy or increase overall power. The use of battery energy storage together with electric motor(s) allows the use of a smaller combustion engine operating under more fuel-efficient conditions. [4]

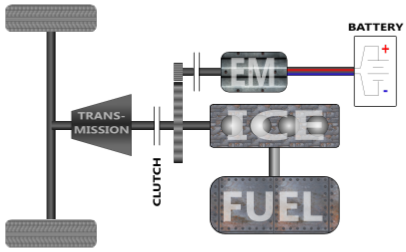


Figure 2.1: Schematic view of parallel hybrid electric vehicle

2.2 GPS and route planning system

Global Positioning System (GPS) is a satellite based positioning system to find longitude, latitude and altitude on the earth's surface. The technology was originally developed by the American military to aid missiles, aircrafts etc. but has evolved to become important for civil use as well. [13, 3]

There exist 24 satellites in the GPS- network covering the entire planet in six fixed orbits. Every satellite transmits encrypted signals telling when the signal has left the satellite and from which one of them. The receiver calculates the traveling time of the signal by comparing the encoded time-stamp with its own. The time of the receiver can been adjusted by other signals sent from the satellites to always be accurate. When distances, i.e. signal transfer time, to many satellites is compared the position can be triangulated with good precision. [13, 3]

Most common GPS-receivers only use satellites directly to give an estimate, with typical precision of 10-15m. There exist extended correction systems, i.e. differential GPS , with land based transmitters used e.g. for aviation and marine with an accuracy of typically 5m [3]. The satellite-signal can in many ways be distorted, blocked or corrupted by e.g. weather conditions, buildings or vegetations and one way to come around this problem is the standard assisted GPS (A-GPS) [3, 15]. Here, the positioning is aided by wireless network technology as cellular phone base stations. This improves the reception quality, lower power consumption and increase information exchange ability [15].

2.3 Look-ahead strategies

Many studies have been done to find strategies towards more optimal control of the hybrid system in HEV's using information about future road conditions. One co-occurring condition is the horizon, found in [1], or look-ahead of the vehicle, in [18] and [7]. where the future road section is analyzed with available information about the section up to a specific point, either in time or distance. The information could be extracted from e.g. an electronic map together with current position given by a GPS-system.

In [1] the horizon is used to extract segments of the road where the State-of-Charge cannot explicitly be controlled, i.e. when the power train is either in recuperation mode or boosting mode. This is parts of the road with either negative road slope or probable deceleration in velocity, e.g. decreasing speed or stop points. The optimization can also be based on all estimated upcoming route positions where the power demand is achieved over the entire trip [18]. This will form a more complete solution, but also demand more computational power.

It is often desirable to find a control strategy that directly can take advantage of the forecasted information and adjust the control parameters to fulfill upcoming route demands. Here equivalent energy strategies are commonly used, i.e. convert electric and fuel energy to comparable form and hereby create a minimization-problem for the fuel energy used [4]. When designing the control system with route prediction in mind, advantages have been found, compared to traditional control strategies. [9], [7].

In order to find the truly optimal power split or the minimal fuel consumption over a designated route, dynamic programming, DP, can be used. Here the entire trip is required in advance and the calculations are demanding. In [8] it is suggested an approximate algorithm (ADP) to the dynamic programming to relax the computational burden. DP is preferable for comparable studies, when the analysis can be done offline and compared to the result of an online control strategy. [8], [1].

The predictive information can also be used to create inputs to already existing control systems. In [10] and [1] a preferred reference trajectory for the SoC of the battery is calculated, based on data on upcoming road. This will influence a HCU, designed to follow a SoC-reference given as input.

2.4 Related areas

The horizon with upcoming altitude and estimated speed, only needs to rely on GPS and electronic map and the information extracted from the latter is static over time. One way to create real time update to this information is to use a traffic information system, [8] With systems in the car capable of receiving this kind of information, the expected velocity can be corrected to traffic conditions, such as queues, accidents or bridge openings. Also red light crossings or railway crossing could be forecasted if the information is accurate. [12]

One large error when predicting the driving of a vehicle is how the driver actually responds to the driving environment. In [17] an energy management system has been evolved depending on driving patterns of commuting drivers. Repeated driving scenarios can be used to find specific attributes for individual drivers.

Crucial, and a large expense, to the HEV is the battery. It is thereby interesting to limit actions that are harmful to the battery. In [5], a strategy to estimate the battery usage or state of health (SOH) is developed. In [6], a prediction control is extended with battery durability.

2.5 MatLab/Simulink

MatLab/Simulink is a high-level computational software developed by *the MathWorks, Inc* widely used both in the industry and in the academic world. MatLab is well suited for numerical calculations, programming, visualizations and general algorithm development. Included in the MatLab package is a number of toolboxes that adds special features such as statistical analysis, linear optimization etc.

Simulink is a graphical interface to MatLab based on block diagrams that allows for the creation of time-varying model based simulations, with ability to include Matlab code with the Embedded Matlab block [11].

3 Initial Arena

This chapter describes the initial conditions of the project, what models were available and what the desired vehicle specifications were.

3.1 Hybrid Control Unit

An essential part of a hybrid vehicle is the hybrid control unit, HCU. A complete hybrid control software system was made available from SAAB Powertrain AB, which in this project is treated as a black-box control unit; see figure 3.1. It was mostly left unchanged with the exception of certain parameters tuned to fit the vehicle specifications. The decision process in the HCU is based around minimization criteria in order to reduce the total losses of the system, but at the same time meet the driver´s requirements.

Current engine- and motor speeds are fed from a plant model to the HCU, in every sample, and decides what amount of torque the corresponding actuator should deliver. This is done in an optimization loop that consider possible torque amounts for a given actuator-speed and losses in the actuator for each available amount of torque. The output from the system is the most fuel optimal amount of torque for each actuator that does not violate the individual constraints and at the same time generates the least amount of overall losses possible.

An important feature of the HCU is its ability to follow a state-of-charge reference. When the actual SoC differs from the reference the HCU will penelize behaviours leading to increasing deviation and favour the opposit. This has a direct effect on the behaviour of the propulsion system, if the SoC reference signal is e.g lower than the actual SoC, the HCU will favour using the electric motor and vice versa.

The State-of-Charge reference signal is assumed to be constant in its original form, however it is possible to control it with the vehicle velocity as control input. This has not been considered in this project.

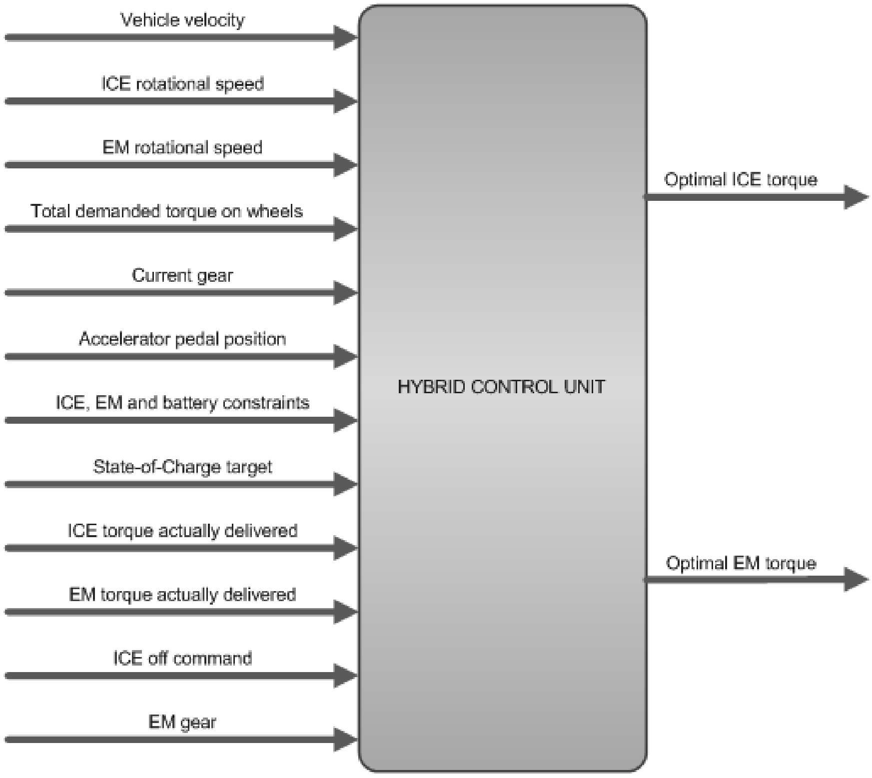


Figure 3.1: A schematic view of the Hybrid Control Unit

3.2 Drivecycles

Typically drivecycles consist of a velocity profile that may be manipulated in order to achieve acceleration and total distance. Standardized drivecycles are widely used for calculations of fuel economy, emissions etc. Common drivecycles include the Federal Test Procedure (FTP-75) used in the United States and the Extra Urban Driving Cycle (EUDC) used in Europe [4], see figure 3.2 for an example.

One drivecycle was made available from Chalmers University which is based upon real measured data from a route in Göteborg, Sweden, see figure 3.3. The drivecycle is measured with a sampling time of 1s and contains the necessary topographic profile that is needed for the developed algorithm to function. [8]

In order to test the functionality of the algorithm, two drivecycles with constant vehicle velocity were developed, see figures 3.4 and 3.5. Test drivecycle 1 features the same topographic profile as the real, measured drivecycle. Test drivecycle 2 features the same topographic profile as in the measured drivecycle but scaled to be more aggresive.

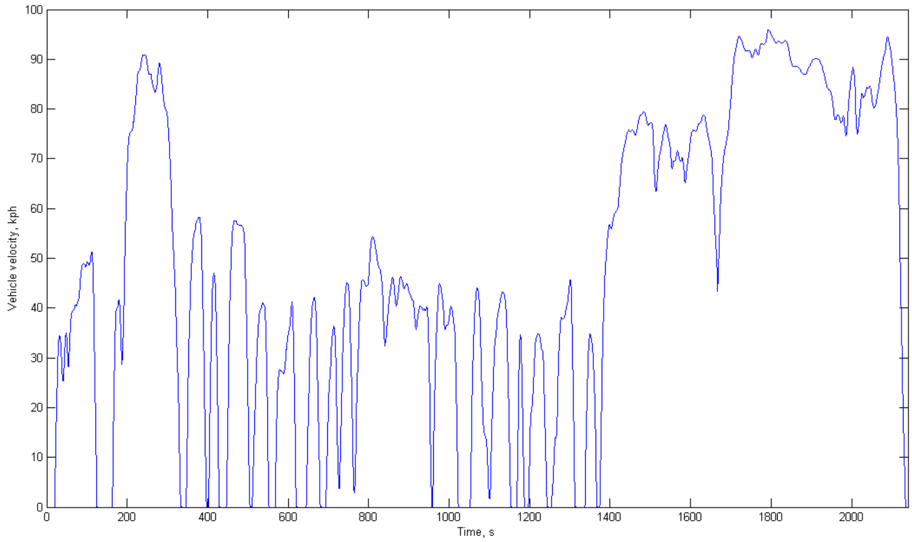


Figure 3.2: FTP drive Cycle

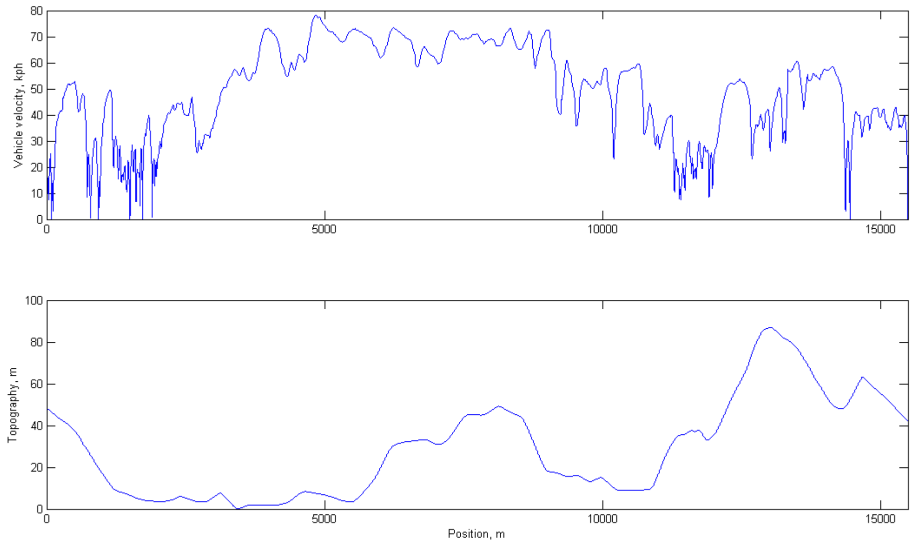


Figure 3.3: Measured drivecycle

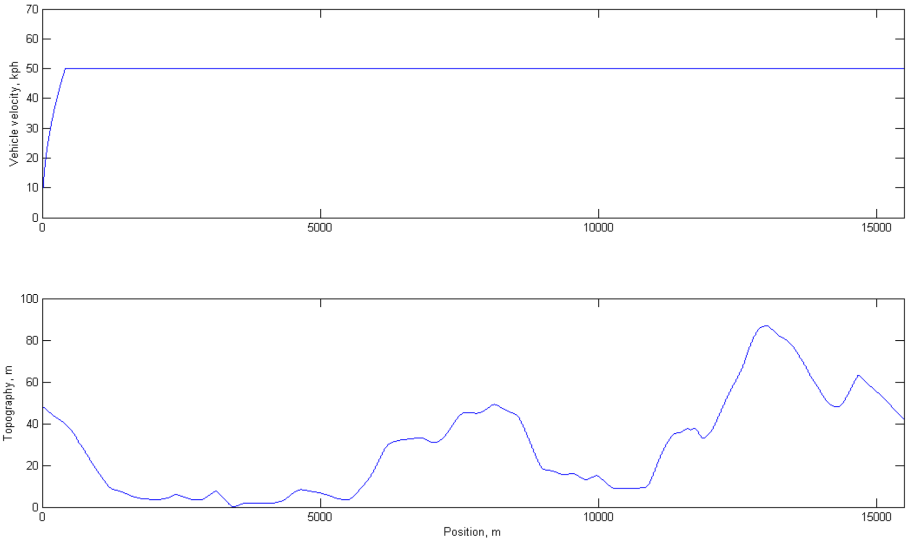


Figure 3.4: Test drivecycle 1 with topography

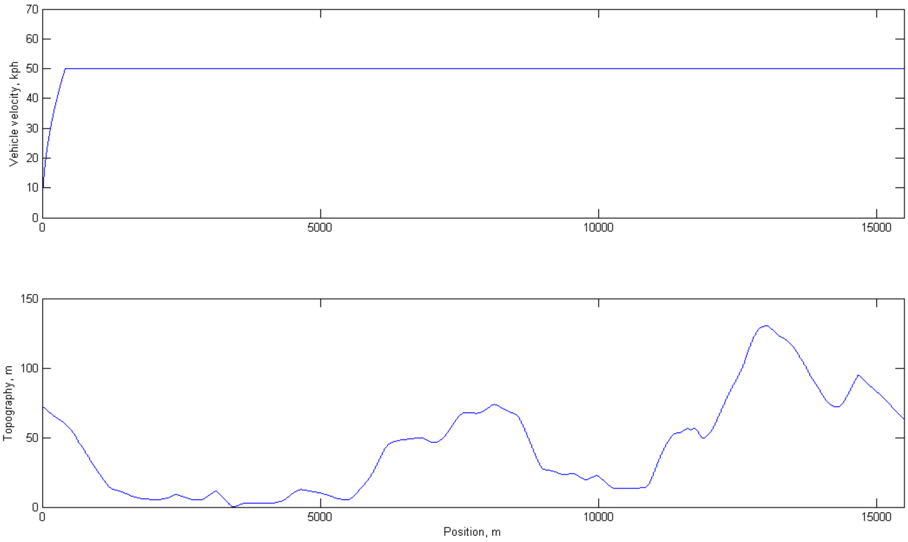


Figure 3.5: Test drivecycle 2 with topography

3.3 Vehicle Specifications

The aforementioned hybrid control unit is generic, developed to be integreted in different vehicles with a wide range of specifications. For this project a SAAB 9-3 Sedan was chosen. Following specifications where used both for tuning the HCU and create a plant (vehicle) model. The latter is a physical representation of the vehicle and simulates how the car responds to the driving cycle and the HCU.

The prime mover of the system was chosen to be a commercially available conventional 2 liter diesel engine. The vehicle features an electric rear wheel axis with a permanent magnet electric motor capable of delivering a continuous power of approximately 12 kW. Due to the choice of rear axis, the vehicle velocity is limited to 120 km/h since no disconnect clutch is to be modelled. The transmission was chosen as a manual gearbox with automatic gear shifts depending on the vehicle velocity. The vehicle will feature a simple battery model, that will mainly act as an energy buffer; the power specification will not limit the electric motor. Three different battery sizes were decided upon, 0.6 kWh, 1.2 kWh and 2.4kWh, and 300 V for all sizes.

4 Modeled and re-parameterized systems

Several models had to be developed in order for the HCU to function properly. This chapter gives an insight of the different subsystems, their role and how they are modelled. The first section covers the simulation flow in order to understand how signals are created.

4.1 Simulation flow and layout

The method that has been chosen is the Quasistatic approach which can be briefly described as an inverse simulation model [1], [8]. The main idea is to use already recorded or standardized drivecycles as the first input to the system (see section 3.2), where the velocity profile and topographic profile is assumed to be known over the complete time period.

With this information available, the required torque and wheel rotation can be calculated in every time-step, as:

$$w_{wheel} = \frac{v}{r_{wheel}} \tag{4.1}$$

$$T_{wheel} = F_{traction} \cdot r_{wheel} \tag{4.2}$$

where

$$F_{traction} = ma + F_{air} + F_{roll} + F_{gravity} \tag{4.3}$$

T_{wheel} is assumed to be 0 when $v = 0$

The inverse simulation approach eliminates the need for a dynamic driver model; the vehicle is instead constarined to only follow the drivecycle. Another method would be to include a driver model that has the velocity profile as a reference and tries to follow the given set-point, similar to a PID-regulator. However, chosen the inverse simulation model eliminates any errors or uncertainties that can influence the results. There are of course potential downfalls to the chosen approach. The resulting fuel consumption might for example be lower due to the absence of driver disturbances.

Another potential issue due to the inverse simulation model could be that it is not sure if the engine, electric motor and control system could cope with the drive-cycle in use. The engine and motor each have upper and lower limits in terms of speed and torque. If the drivecycle demands a higher/lower output torque/speed than what both the engine and motor can deliver, signal saturation will occur. However, the simulation will not stop. This means that the demanded power output must always be checked towards the produced power and make sure that they are equal.

4.2 Plant model

The purpose for developing a plant model is to create the signals needed for the HCU to function as intended, and to process the signals generated by the HCU. Refer to figure 4.1 for a schematic figure.

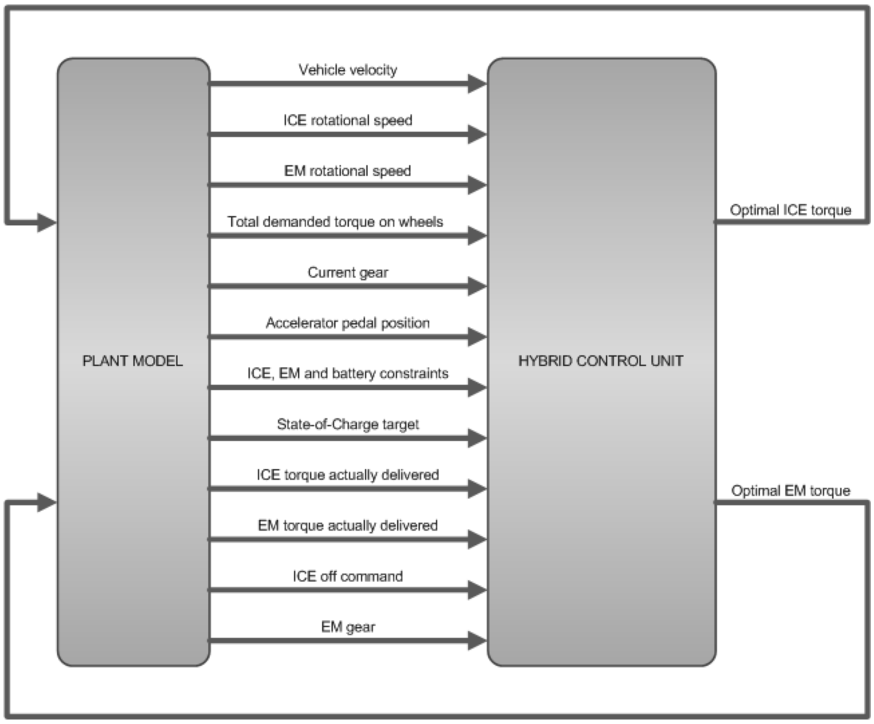


Figure 4.1: The HCU coupled with the plant model

The vehicle model that has been the basis for all simulations in this thesis is a parallel hybrid vehicle where the electric motor (EM) is coupled with the rear wheel axle, see figure 4.2. This means that the engine and electric motor are completely decoupled from each other, and allows for the electric motor to propel the vehicle when the engine is turned off during certain conditions. However, the electric motor cannot act as a conventional generator when the vehicle is at a stand-still since the electric motor is decoupled from the transmission shaft.

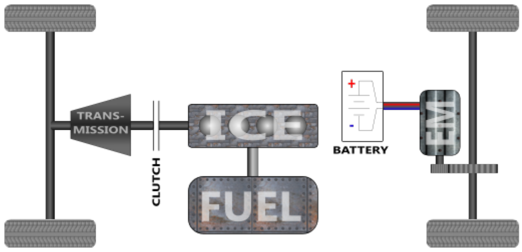


Figure 4.2: Architecture of the plant model used in simulations

4.2.1 Chassis model

The chassis model is regarded as the first system in the simulation flow where the torque and wheel rotational velocity is determined. Inputs to the system are the vehicle velocity, v_{kph}, acceleration, a, and slope of road,α.

Instead of using conventional energy calculations for aerodynamic drag force and rolling friction force, a quadratic formula based upon real driving measurements is used. The total force acting on the vehicle is described by:

$$F_{tot} = m \cdot a + F_1 + F_2 \cdot v_{kph} + F_3 \cdot v_{kph}^2 + m \cdot g \cdot sin\left(\alpha\right) \tag{4.4}$$

The resulting output is the total wheel torque request by the driver (in this case, the drivecycle) which is given by:

$$T_{wheel} = F_{tot} \cdot r_{wheel} \tag{4.5}$$

4.2.2 Gearbox model

The vehicle is assumed to have a transmission with six gears and one final drive. Gear changes occur when the vehicle reaches a certain velocity and the model is implemented using relay logics. The transmission system is realized very simple and is assumed of having an constant efficiency of 95%. The vehicle speed, v_{kph}, is the only input to this system. The outputs are the current gear and gear ratio. Figure 4.3 shows gear changes on the FTP drivecycle, see figure 3.2.

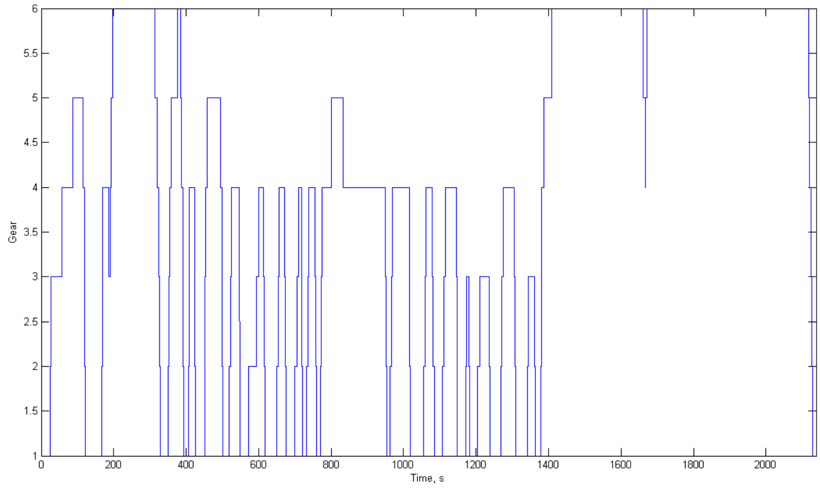


Figure 4.3: Gear changes on FTP cycle

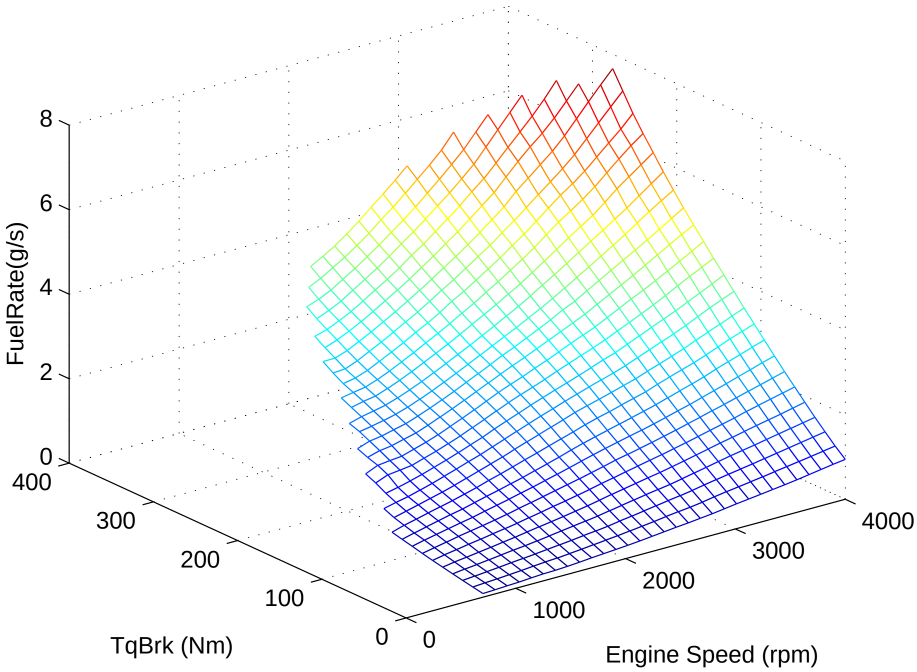


Figure 4.4: Engine fuel consumption map

4.2.3 Engine

Inputs to the engine block are the engine speed that is derived from the gearbox model, and the torque request that is determined in the HCU. The outputs are the actual delivered torque and the momentary fuel consumption. The momentary fuel consumption is given through a lookup function, see figure 4.4. No internal losses have been considered.

The engine operating speed is between 700 and 4100 rpm and the available output torque range is 0 – 355 Nm depending on the engine speed.

4.2.4 Electric motor

Inputs to the system are the current electric motor rotational speed and the torque request determined by the HCU. Outputs from the model are the resulting current (positive or negative), resulting power, resulting motor losses and the motor efficiency.

The motor is of permanent magnet type and is connected to the rear axle (see figure 4.2) with gear ratio 11.

4.2.5 Battery

A generic battery model has been used to simulate an energy buffer. Inputs to this system are the charge/discharge current of the electric motor. The State-of-Charge is given as output. The resulting State-of-Charge (SoC) is determined in this model as the ratio between the electric charge that can be delivered from the battery Q, to the nominal battery capacity Q_0

$$SoC\,(t) = \frac{Q\,(t)}{Q_0\,(t)} \tag{4.6}$$

In order to preserve battery lifetime and to protect against over/undercharging, the State-of-Charge is only allowed to operate in between a given window [14], the State-of-Charge window is set to $60\% - 80\%$.

4.2.6 Auto start / stop

A simple event-based engine-off system has been developed to mimic a start/stop-behaviour. This model allows the vehicle to shut off the engine when certain criterias are valid, thus allowing the vehicle to run purely on electric power. The auto start/stop model shuts the engine off when the following criterias are fullfilled:

- The vehicle velocity is below a certain threshold
- The SoC is above a lower limit
- The electric motor can produce enough power to meet the power demand on the wheels.

If the engine is turned on, an additional criteria was created to force the engine to stay on for at least 10s in order to avoid too many start/stops of the engine in a short time-period.

4.2.7 Pedal positions

Although no driver model is present, the HCU still requests the accelerator pedal-position as input. Instead of creating a complete driver model, a simple pedal-position block has been modeled. It is strictly based on a pedal-position map that has the vehicles velocity v_{kph} and the total wheel torque request as input and produces the corresponding accelerator pedal-position (0-100%) as output.

No brake pedal has been modeled as it is assumed that all the negative wheel torque that the engine and electric motor cannot handle is covered by regular friction brakes.

4.3 Predictive SoC-reference signal generator

Modern cars often feature a navigation system, described in section 2, onboard and it is assumed that the vehicle used in this project is equipped with a standard navigation system, that is assumed to be connected to the HCU. The information is analyzed in different steps in order to generate a position dependant SoC-reference signal. This information contains future expected velocity (based on speed limit signs), a complete altitude trajectory and also start and stop positions for segments where recuperation may take place. See figure 4.5 for an overview of the information exchange.

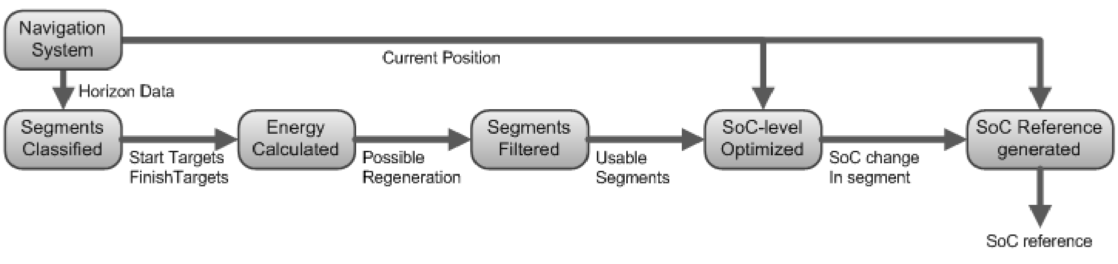


Figure 4.5: Schematic description of the prediction system

The expected output from the reference signal generator is a position dependent state-of-charge reference trajectory that is based on the expected SoC changes within the recuperation segments and the original SoC reference signal. The overall design goal has been to lower the SoC reference signal before an important regeneration segment to maximize the amount of energy regenerated, without violating the constraints of the battery. By lowering the SoC reference, the HCU will favour operations using the EM and penalize the use of the ICE in order to reach the SoC target, thus decreasing the fuel consumption. This is done by creating a deviation from the original SoC reference when the information from the navigation system indicates that energy can be regenerated in an upcoming road segment. This deviation is intended to be zero when no energy can be regenerated; hence the goal is always to make the generated SoC reference signal close to the original SoC reference at the end of the horizon.

4.3.1 Horizon from the navigation system

The infotainment-system in the car is assumed to hold 3D maps with good precision together with a GPS system. As mentioned in section 1.3, the signal is here simplified to only carry one-dimensional positions together with elevation. This data is extracted from the same vectors that describes the drivecycle, see section 3.2, which feeds the plant. This assumed perfect estimation of the position, no errors in the map structure and ability to send information to the hybrid control with no restrictions. Together with the aforementioned data, maximum allowed speed of the road is given in every position. The simulated GPS/map-system transmits the vector:

$$horizon(simulationStep) = \begin{bmatrix} altitude \\ speedLimit \end{bmatrix}_{position_{horizon}(simulationStep)} \tag{4.7}$$

When the car is started the system is triggered to collect the horizon data given by $position_{horizon} < length_{horizon}$, while standing still, $position_{vehicle} = 0$. This fast forward search is simply a loop where $position_{horizon}$is iterated with fix step length in the interval$\{0, length_{horizon}\}$:

$$while\ (k < length_{horizon}) \begin{cases} position_{horizon} = k + length_{resolution} \\ k = k + 1 \end{cases} \tag{4.8}$$

The generated instances of $horizon$ is stored in a buffer, with size $\frac{length_{horizon}}{length_{resolution}}$, which is depleted in every simulation step, until the buffer is empty. After this phase the postion of the horizon will instead be given by:

$$position_{horizon} = position_{vehicle} + length_{horizon} \tag{4.9}$$

Since it was wanted not to lose any data between the initial position and the initial horizon position, the fast forward search simulates the vehicle travel the road in shortened time. The step-length-parameter defines the motion steps in distance that should be used. Smaller steps take longer time to complete and a consideration must be done between completing this phase without losing real-time data and at the same time not lose too much information due to too large steps.

4.3.2 Segmentation

The first analysis of the signal, $horizon(simulationStep)$, is a logic segmentation. Segments are set to sections of the road that have negative slope and maintained or decreseasing allowed speed, e.g. where it is most possible that recuperation of energy will take place. It was assumed that an increasing allowed speed would cause an actual increase in vehicle velocity, thus not give opportunity of recuperation. These rules are summerized as:

$$\begin{array}{rr} (altitude(simulationStep) < altitude(simulationStep-1)) & \wedge \\ (speedLimit(simulationStep) \leq speedLimit(simulationStep-1)) & \Rightarrow \\ \Rightarrow segmentStartFlag & \end{array} \tag{4.10}$$

$$\begin{array}{rr} (altitude(simulationStep) \geq altitude(simulationStep-1)) & \wedge \\ (speedLimit(simulationStep) > speedLimit(simulationStep-1)) & \Rightarrow \\ \Rightarrow segmentStopFlag & \end{array} \tag{4.11}$$

The segments is only generated depending on assumed travel speed and without any acceleration phases. Naturally this will cause an error compared to the actual recuperation taking place. The car could for instance be forced to make unpredicted stops, caused by traffic conditions, which would first generate a large deceleration and later an acceleration. Also, the vehicle, with a driver, will not constantly follow the speed limits given by the map. The actual speed will first deviate from the predicted speed and second, accelarations not predictable will occur.

Refer to the upper graph in figure 6.1, where identified regeneration segments clearly can be seen.

4.3.3 Energy calculations of recuperation segments in horizon

Assuming that all incoming segment within the scope of the horizon contain some energy that can be used to passively charge the battery, the segments´ energies must be calculated. Each segment i, is characterized by its length in meters, assumed travelling velocity and mean slope. Where the slope is calculated by:

$$\alpha = \arctan\left(\frac{\triangle altitude_i}{\triangle distance_i}\right) \tag{4.12}$$

The average required force F_i acting on the vehicle in segment i is calculated from the same energy calculations used in section 4.2.1. Note the subscript in the below equation. This is a consequence of not taking the electric path constraints in consideration at this point.

$$F_{i,unlimited} = F_1 + F_{(2)} \cdot v_{kph} + F_{(3)} \cdot v_{kph}^2 + m \cdot g \cdot \sin(\alpha) \tag{4.13}$$

A substantial difference is the lack of the term $F = m \cdot a$ since the assumed vehicle velocity in segment i is given as a constant value. The acceleration term is thus zero.

Since the electric path of the vehicle has constraints to how much energy it can handle, a limiting function has been introduced as follows:

$$F_i = max\left(F_{i,unlimited}, \frac{P_{min}}{v \cdot \eta_{recuperation}}\right)$$

Where P_{min} is the maximum amount of the negative power the electric path can handle at the predicted electric motor rotational speed, and $\eta_{recuperation}$ is the average efficiency of the electric path.

The total estimated amount of energy in segment i, is then calculated by:

$$E_{recuperation,i} = \begin{cases} -F_i \cdot \triangle distance_i & F_i < 0 \\ 0 & else \end{cases} \tag{4.14}$$

The extra criterion that $E_{recuperation,i}$ should be zero if $F_i >= 0$ is a consequence of not taking boosting phases etc. into consideration when creating the SoC-reference.

Finally the estimated SoC change that may occur in segment i, is calculated from:

$$\triangle SoC_i = \frac{E_{recuperation,i}}{Q_0 \cdot V_0 (SoC_{set})} \tag{4.15}$$

Where Q_0 is the nominal battery capacity and V_0 is the nominal voltage of the battery. Note that the estimated SoC changes in segment i vary depending on battery specifications.

4.3.4 Generation of state-of-charge reference trajectory

With future segment data given, including the estimated SoC changes in each segment, the position dependent State-of-Charge reference can be synthesized. The objective is to find optimal start values of the State-of-Charge for the recuperation segments so that State-of-Charge boundaries are never exceeded. Since the estimated SoC-changes in the recuperation segments are known, it is the trajectory from the end of recuperation segment i, to the start of recuperation segment $i+1$ that are to be found.

The requirements of the calculated State-of-Charge reference are

- The state-of-charge reference signal must stay within the State-of-Charge windows boundaries $[SoC_{max}, SoC_{min}]$ consistently.
- At the end of the horizon, the State-of-Charge should stay within a target interval $[SoC_{set} - \delta SoC_{set},\ SoC_{set} + \delta SoC_{set}]$. This is a consequence of not knowing if the end of the horizon is the end of the driving mission.

The notation *free* is used in order to mark the resulting segment that is between two known regeneration segments.

The problem is formulated as the following minimization problem:

$$\min_{x} \left(\sum_{i=1}^{N} \frac{\triangle SoC_{free,i}}{\triangle dist_{free,i}} \right) \tag{4.16}$$

Where N is the number of segments that the optimization routine receives as input.

$\triangle SoC_{free,i}$ is the unknown difference in State-of-Charge from the end of a known segment to the unknown beginning of the new segment. Hence, $SoC_{,i}$is the unknown variable that is to be found:

$$\triangle SoC_{free,i} = SoC_i - (SoC_{i-1} + \triangle SoC_{i-1}) \tag{4.17}$$

Where $SoC_0 =$current SoC,$\triangle SoC_0 = 0$ and$\triangle dist_{free,i}$ is the distance between recuperation segment i and recuperation segment $i-1$.

If the vehicle is inside a recuperation segment while the optimization is started, the current recuperation segment is not taken into consideration. This implies two cases

1. If the vehicle is inside a recuperation segment that has previously been part of an optimization routine, the State-of-Charge trajectory to the end of that particular segment is assumed to be known. Hence the current optimization routine only takes recuperation segments ahead of the current segment into consideration.
2. If the vehicle is inside a recuperation segment that has not been included in a previous optimization routine, no information about estimated SoC change is known and the segment as a whole is discarded. This may happen if the horizon is short and the end position of the segment is not known until the vehicle is already inside the segment.

The resulting, unknown, vector x is given as

$x = [SoC_1, \ldots, SoC_N, \epsilon_1, \ldots \epsilon_N]^T$

where ϵ are slack variables introduced to make the solution feasible.

The problem is formulated as a linear programming problem

$$\min_x \sum_i^N c_i x_i \tag{4.18}$$

so that

$$Ax \leq b \tag{4.19}$$

where the inequalities are given by

$$SoC_i \geq SoC_{ref,min} - \epsilon_i \tag{4.20}$$

$$SoC_i + \triangle SoC_i \leq SoC_{ref,max} + \epsilon_i \tag{4.21}$$

$$SoC_N + \triangle SoC_N \geq SoC_{set} - \delta SoC_{set} - \epsilon_N \tag{4.22}$$

$$SoC_N + \triangle SoC_N \leq SoC_{set} + \delta SoC_{set} + \epsilon_N \tag{4.23}$$

where SoC_{set} is the desired State-of-Charge target at the end of the horizon length and $\delta SoC_{set}$is the maximum allowed deviation from SoC_{set}

Finally the affine SoC reference is created by interpolating between the optimized SoC values with the current vehicle position as input.

For further reading about linear programming, the reader is encouraged to refer to [16].

5 Verification of developed modeles

5.1 Plant model

The first step in the simulation chain is the plant-model that ultimately powers the preceding models. If the plant-model does not produce signals that are valid, the consequent result will not be valid. An important output that is produced by the plant-model is the driver demanded torque on wheels, see section 4.2.1. This signal is the foundation for how the control algorithm should distribute the torque among the two actuators (EM and ICE). In order to verify this signal, a comparison was made with a more comprehensive simulation model developed by SAAB Powertrain AB. The comparison was based on a simulation were both models were running the same drivecycle (FTP). Figure 5.1 shows the comparison, where the solid line is the developed plant model and the dotted is from the more sophisticated simulation model.

Note that the signal differs due to the simplicity of the plant model, however the two signals follow each other in size and time. The high and low extremes in the comparative simulation run is from gear changes not present in the simplified plant model.

Since the goal of the project has been to compare fuel consumption between vehicles with and without a predictive State-of-Charge reference signal generator, the plant model must produce as accurate results as possible in terms of delivered engine torque and momentary fuel consumption. Figure 5.2 shows the result, where the red graphs indicate the developed plant model and the blue shows the comparative model.

As can be noted, the signals follow eachother in size and time, however some deviations have been noted. Due to set down-shifts in the developed plant model, the engine speed differs as the vehicle velocity decreases. Down-shifts are not present in the comparative model. Another major deviation can be seen in the upper graph of figure 5.2, where the comparative model order fuel cut-off at a number of times. The momentary fuel consumption of the comparative model is thus zero, where the developed plant model has a non-zero fuel consumption. This difference accounts for approximately 57.5 grams of fuel, equal to about 0.02 l/10km. Also, the comparative model features spin-losses which the developed plant model does

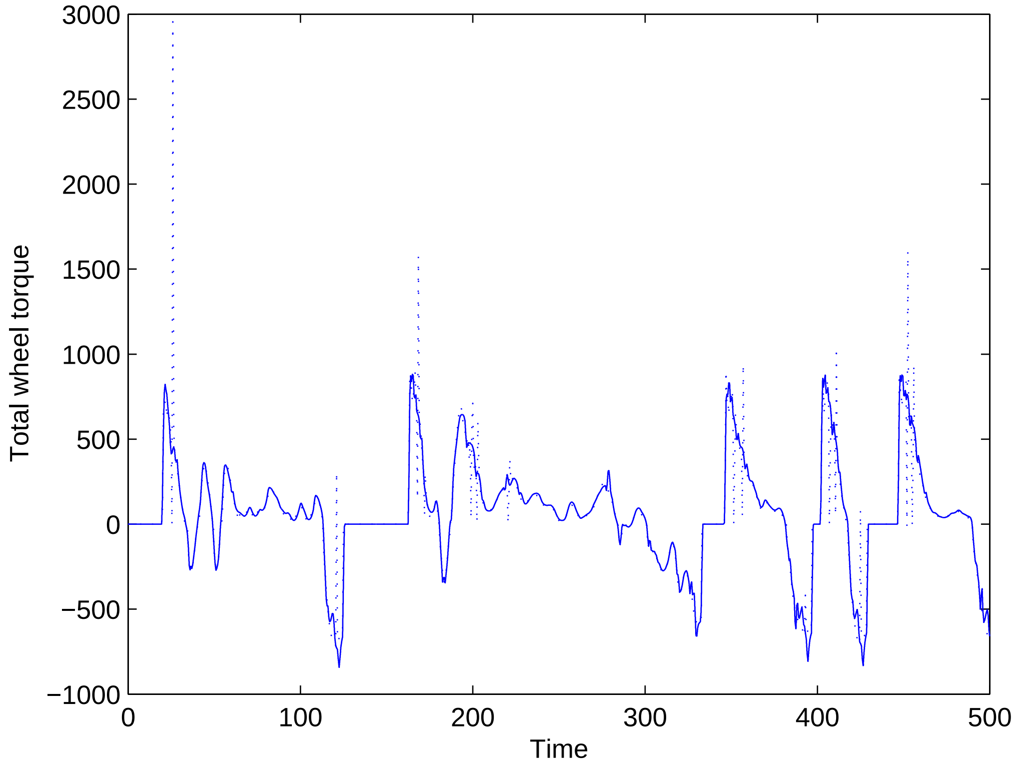


Figure 5.1: Performance of the plant model

not, meaning that a somewhat lower fuel consumption of the plant model is expected.

The resulting fuel consumption from the two simulation runs can be seen in table 5.1.

Figure 5.3 shows the plant model with the electric motor connected. Torque and speed of the two actuators are shown, where the electric motor correspond to red graphs. The behaviour of the auto start/stop function can clearly be seen. The resulting fuel consumption of the plant model running in hybrid mode was 0.37 l/10km.

Plant model	Comparative model
0.48	0.49

Table 5.1: Resulting fuel consumption of the developed plant model and the comparison (l/10km)

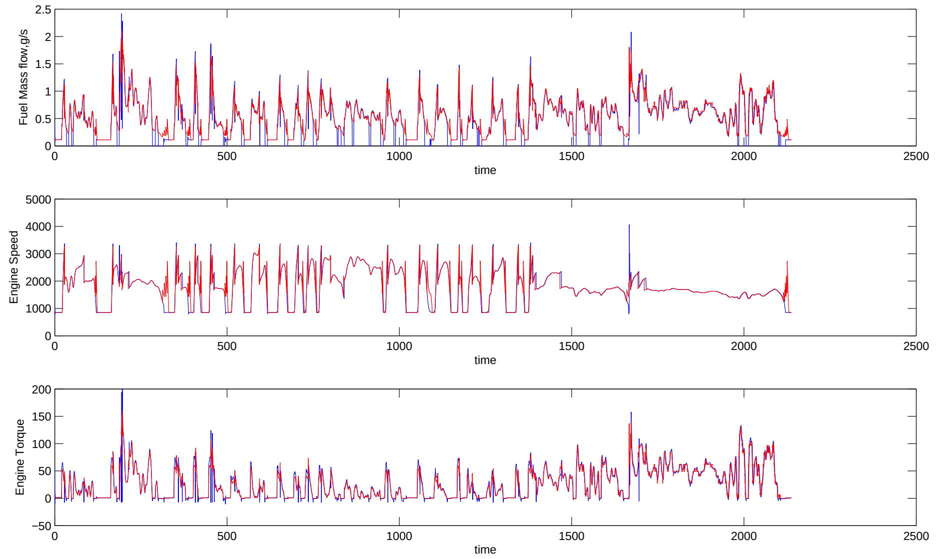


Figure 5.2: Comparison of momentary fuel consumption, engine speed and engine torque

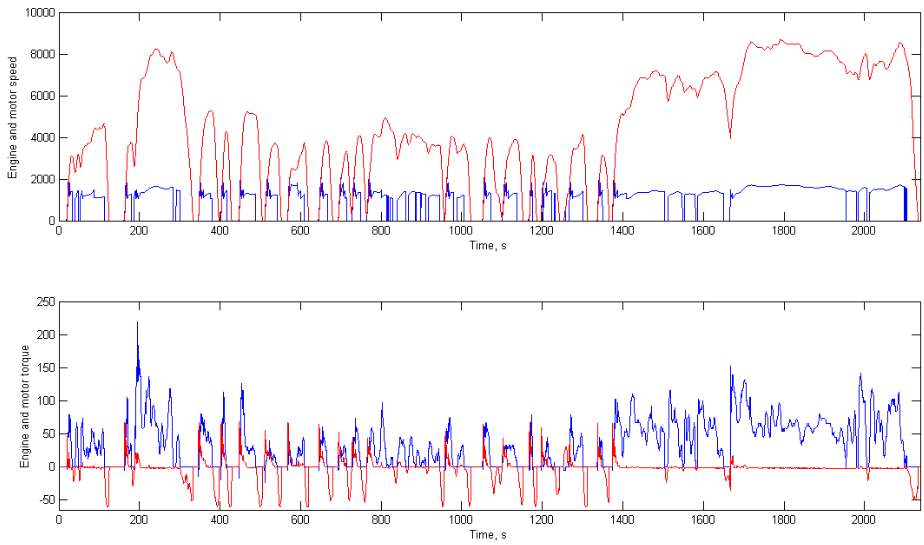


Figure 5.3: Plant model running in hybrid mode on FTP cycle

5.1.1 Horizon

The generated trajectories for position and altitude profile in the horizon offset should only differ in time from current valid values, since the signal from the assumed navigation system and the real positions are generated from the same drivecycle vectors, as described in section 3.2. A direct comparison of the predicted road section and the actual driven road was done visual in the end of the drivecycle simulation. The signals from the two time domains where plotted against distance instead, to verify that same result had been achieved.

Due to the fast forward search in the beginning of the horizon generator, the signal displays with less resolution in this section. The resolution is predefined before the simulation starts, as described in section 4.3.1.

5.1.2 Verification of reference signal generator

The State-of-Charge reference signal generator is verified by making sure that the constraints of the system are never violated. If the resulting reference signal is above/below the allowed SoC window, the reference is not correct. The reference should also follow the segments in length, meaning that a starting SoC value must be at the same position as the starting segment. A final check is made to verify that the end of the reference signal is within the allowed target.

6 Result collection

The collected results are divided into two main sections. In the first part (6.1) more simplified driving scenarios, described in section 3.2, are used in order to observe the performance of the prediction system given circumstances matching the design parameters. The second section describes how the system respond to real driving, using measured drivecycle data.

Each subsection begins with a short analysis explaining what is seen in the collected data. Blue lines in graphs shows results where a hybrid vehicle is equipped with the developed algorithm and red interrupted lines represent the hybrid vehicle without the developed algorithm. The filled areas in the top section of the graphs correspond to useable segments; segments represented by blue interrupted lines indicate not useable segments. Fuel consumption (FC) and comparison is given in tables after the analysis of each section.

6.1 System response

The results in this section are divided between the two constructed drivecycles. Comparisons between three battery capacities and the two horizon lengths are shown.

6.1.1 Analysis and summary of Test drivecycle 1

As can be seen in the figures below, the reference signal is lowered before an important recuperation segment and is raised during the segment to ensure that as much energy as possible can be regenerated. This is especially noticeable when having a complete knowledge about the drivecycle in advance; see figures 6.1, 6.2 and 6.3. The fuel consumption is reduced in all cases, however the size of the reduction depends on the horizon length and battery size where a larger battery gives a lower fuel reduction.

When using a short horizon, the same amount of segments is not found. They are found mostly one at a time, treating that segment as the last segment. This is the reason for the reference signal only being lowered in order to be the same as the original reference at the end of the segment. Using a short horizon reduces the fuel savings in all cases.

6.1.1.1 Test drivecycle 1 with full length horizon

Torque distribution for this drivecycle can be found in Appendix A.

Battery capacity:	0.6	1.2	2.4	kWh
FC w/o. algorithm	0.276	0.266	0.264	l/10km
FC w. algorithm	0.263	0.261	0.262	l/10km
Difference	-4.71	-1.88	-0.76	%

Table 6.1: Fuel consumption comparison, Test drivecycle 1 with full length horizon.

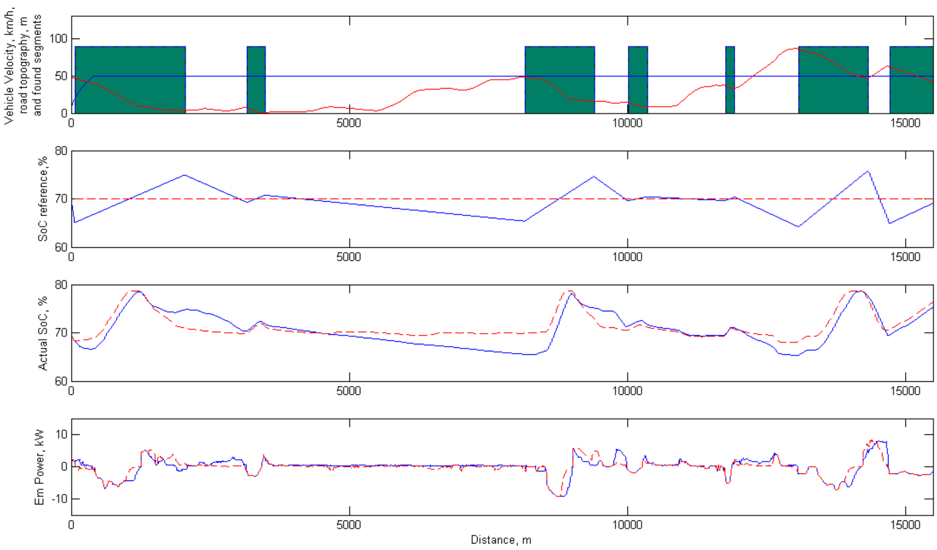


Figure 6.1: Comparison between simulation runs with and without algorithm, test cycle 1, 0.6 kWh, full length horizon.

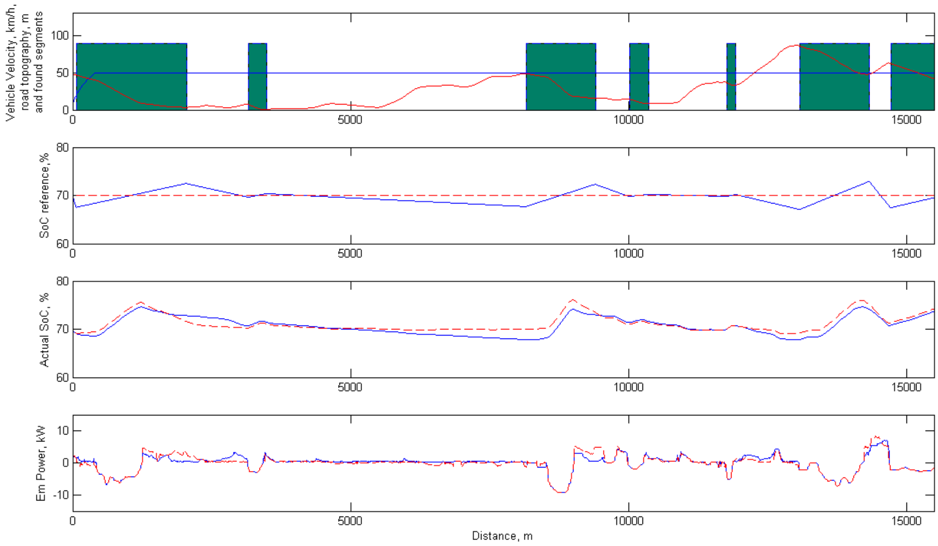


Figure 6.2: Comparison between simulation runs with and without algorithm, test drivecycle 1, 1.2 kWh, full length horizon.

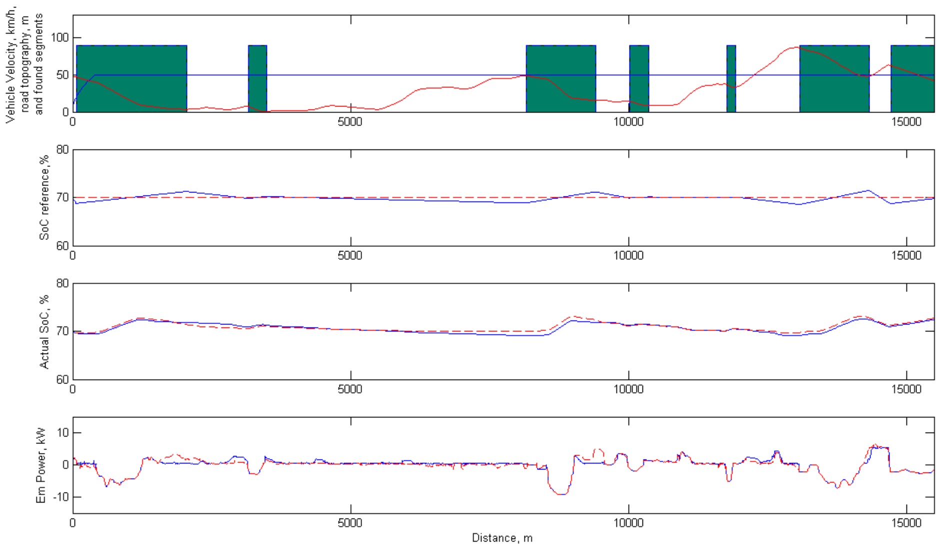


Figure 6.3: Comparison between simulation runs with and without algorithm, test drivecycle 1, 2.4 kWh, full length horizon.

6.1.1.2 Test drivecycle 1 with horizon length 1050 meters

Battery capacity:	0.6	1.2	2.4	kWh
FC w/o. algorithm	0.276	0.266	0.264	l/10km
FC w. algorithm	0.275	0.266	0.263	l/10km
Difference	-0.36	0	-0.38	%

Table 6.2: Fuel consumption comparison, test drivecycle 1 with horizon length 1050 m.

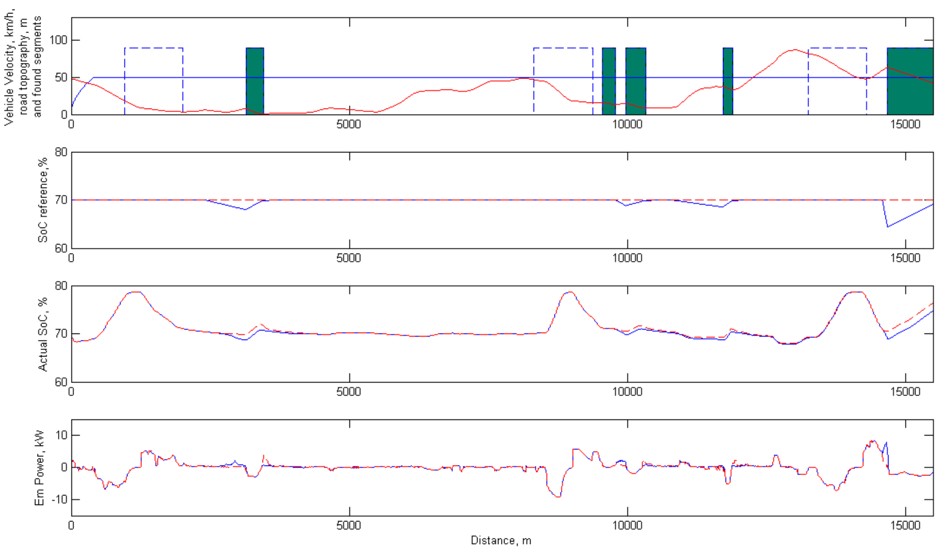


Figure 6.4: Comparison between simulation runs with and without algorithm, test drivecycle 1, 0.6 kWh, short horizon.

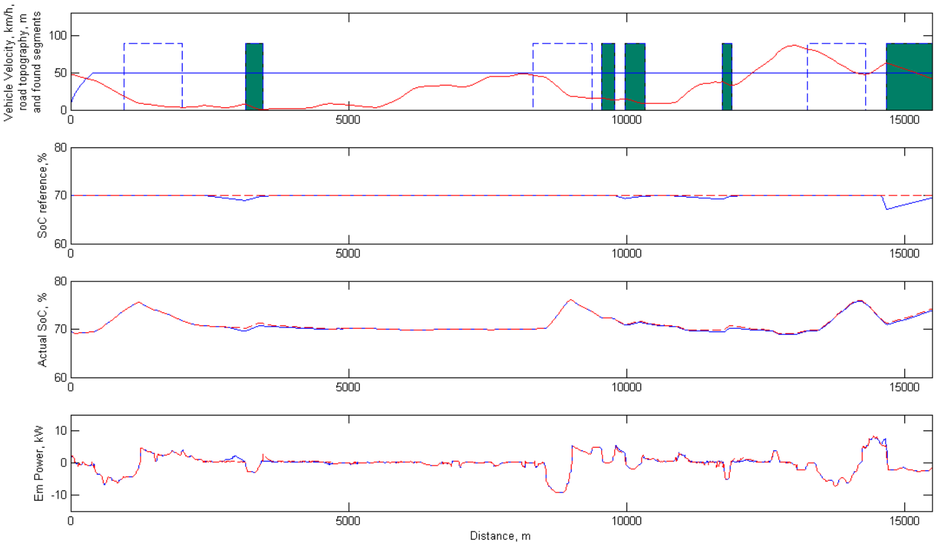


Figure 6.5: Comparison between simulation runs with and without algorithm, test drivecycle 1, 1.2 kWh, short horizon.

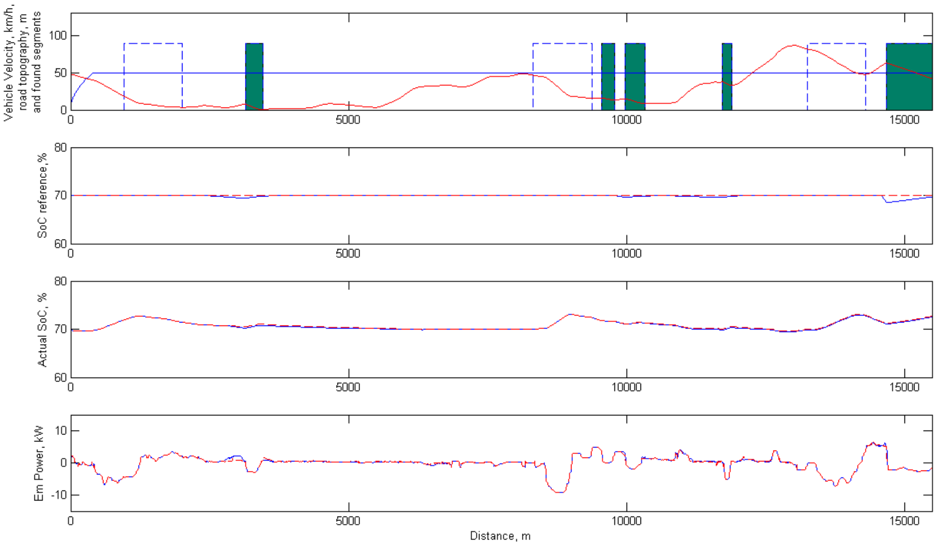


Figure 6.6: Comparison between simulation runs with and without algorithm, test drivecycle 1, 2.4 kWh, short horizon.

6.1.2 Analysis and summary of test drivecycle 2

As can be seen in table 6.3, the overall fuel consumtion is slightly higher due to the increasing slopes. The energy calculations showed that more energy was held in the segments, causing the reference signal to have a more distinct behavior. New usable segments were found since the gravitational force overcame the aerodynamic force and the rolling friction, see figures 6.7, 6.8 and6.9.

Since more energy was held in the segments, the actual SoC reached the upper battery limit in some cases when having a constant SoC reference, resulting in energy lost to the friction brakes. The algorithm successfully lowered the SoC reference before these segments resulting in less energy waste.

6.1.2.1 Test drivecycle 2 with full length horizon

Battery capacity:	0.6	1.2	2.4	kWh
FC w/o. algorithm	0.392	0.340	0.307	l/10km
FC w. algorithm	0.344	0.307	0.296	l/10km
Difference	-12.2	-9.71	-3.58	%

Table 6.3: Fuel consumption comparison, test drivecycle 2 with full length horizon.

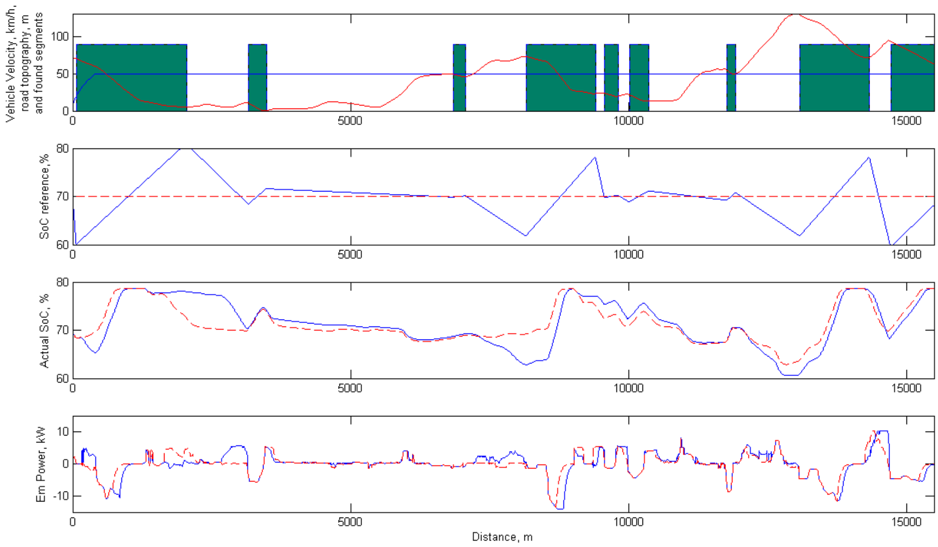


Figure 6.7: Comparison between simulation runs with and without algorithm, test drivecycle 2, 0.6kWh, full length horizon.

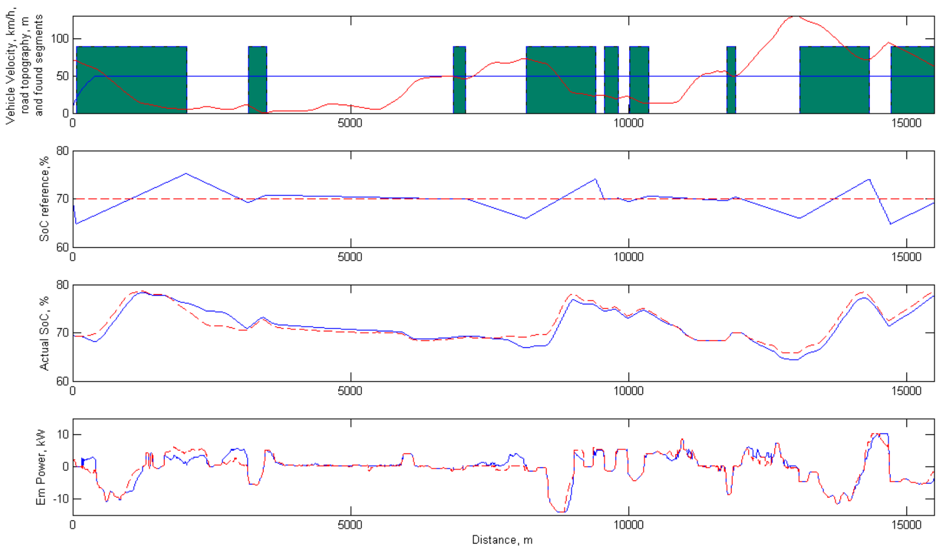


Figure 6.8: Comparison between simulation runs with and without algorithm, test drivecycle 2, 1.2 kWh, full length horizon.

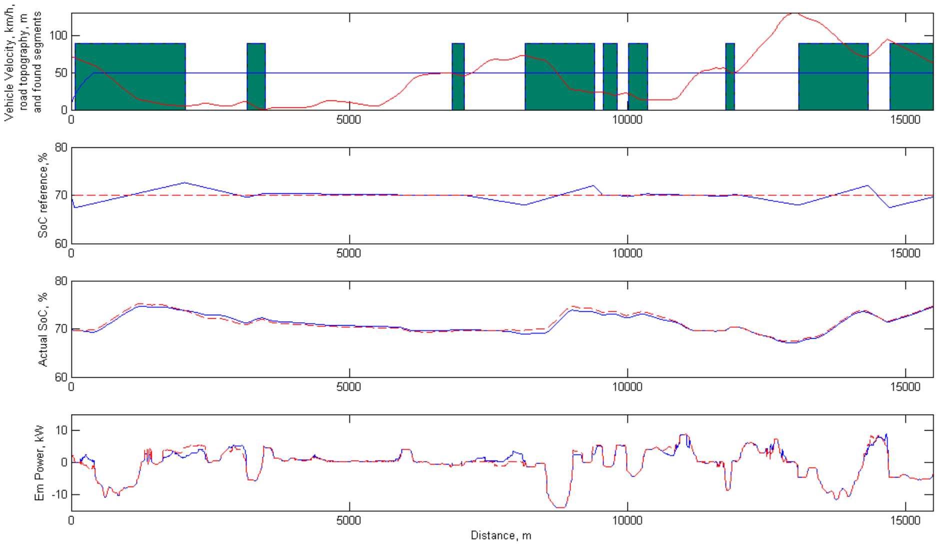


Figure 6.9: Comparison between simulation runs with and without algorithm, test drivecycle 2, 2.4 kWh, full length horizon.

6.1.2.2 Test drivecycle 2 with horizon length 1050 meters

Battery capacity:	0.6	1.2	2.4	kWh
FC w/o. algorithm	0.392	0.340	0.307	l/10km
FC w. algorithm	0.391	0.340	0.307	l/10km
Difference	-0.26	-0.09	-0.07	%

Table 6.4: Fuel consumption comparison, test drivecycle 2 horizon length 1050 m.

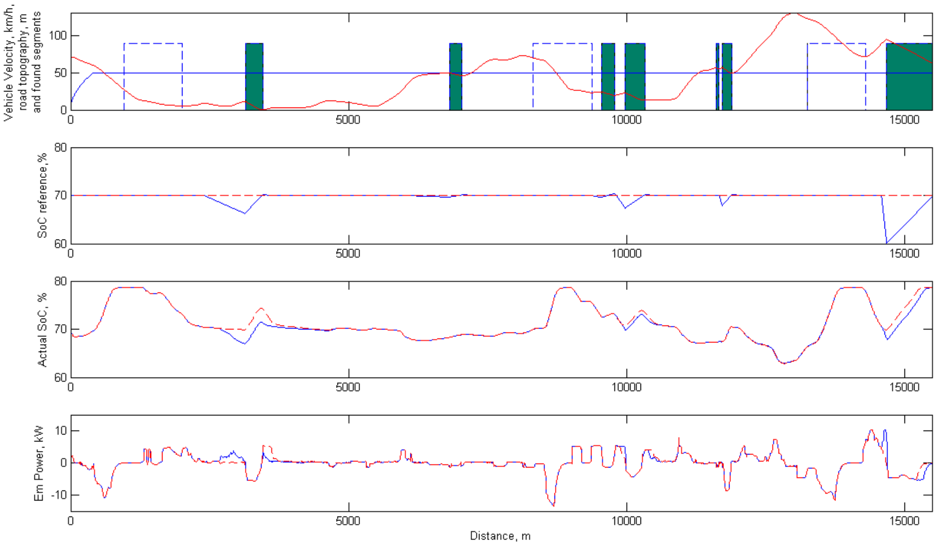


Figure 6.10: Comparison between simulation runs with and without algorithm, test drivecycle 2, 0.6 kWh, short horizon.

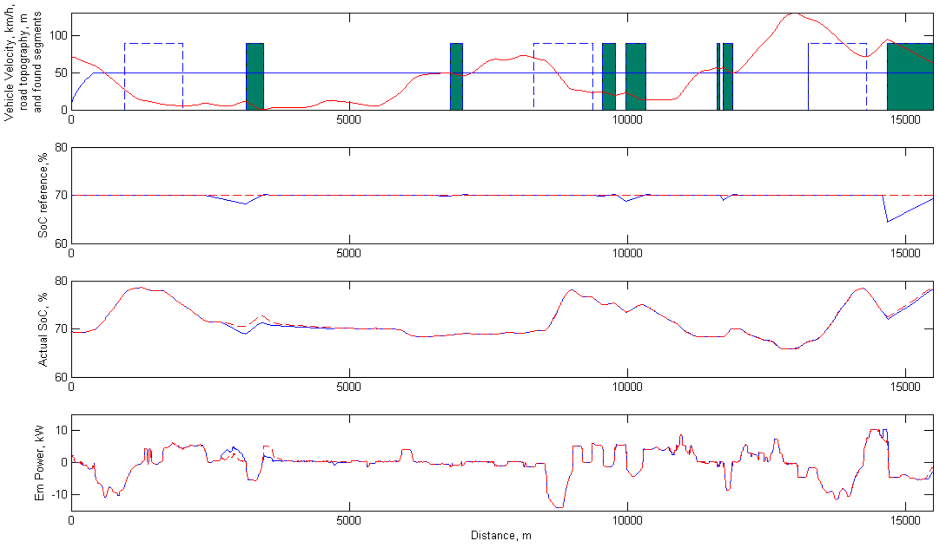


Figure 6.11: Comparison between simulation runs with and without algorithm, test drivecycle 2, 1.2 kWh, short horizon.

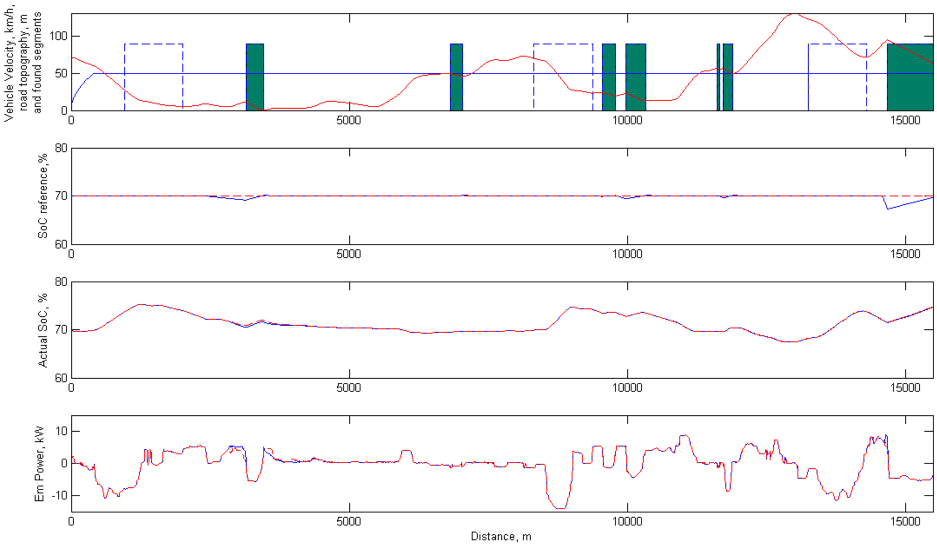


Figure 6.12: Comparison between simulation runs with and without algorithm, test drivecycle 2, 2.4 kWh, short horizon.

6.2 Test with measured drivecycle

This section shows the performance of the developed algorithm if the vehicle are driven in a more realistic manner, using the measured data (see section 3.2). Comparisons between three battery capacities and the two horizon lengths are shown. As in previous sections there is a short analysis. Blue lines in graphs shows results where a hybrid vehicle is equipped with the developed algorithm and red interrupted lines represent the hybrid vehicle without the developed algorithm. The filled areas in the top section of the graphs correspond to useable segments; segments represented by blue interrupted lines indicate not useable segments. Fuel consumption (FC) and comparison is given in tables after the analysis of each section.

6.2.1 Analysis and summary of the measured drivecycle

This test investigated how robust the algorithm was when the driver did not follow the speed limits. For a long horizon, the algorithm performed well since the larger segments with a lot of energy was found. However, the short horizon did actually give a slightly higher fuel consumption. This is probably due to the driver not following the speed limit and requesting a different energy profile than what was expected. The actual SoC could in these cases have been lowered to much in comparison to what was obtained, resulting in a higher torque demand for the engine to force the actual SoC back to the reference signal.

6.2.1.1 Measured drivecycle with full length horizon

Battery capacity:	0.6	1.2	2.4	kWh
FC w/o. algorithm	0.419	0.397	0.381	l/10km
FC w. algorithm	0.400	0.389	0.352	l/10km
Difference	-4.53	-2.02	-7.61	%

Table 6.5: Fuel consumption comparison, measured drivecycle with full length horizon.

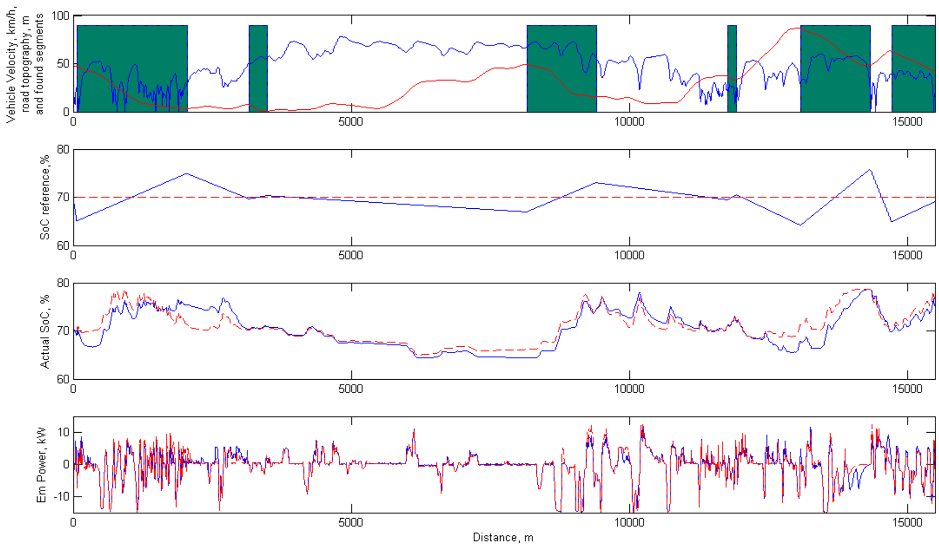


Figure 6.13: Comparison between simulation runs with and without algorithm, measured drivecycle, 0.6 kWh, full length horizon.

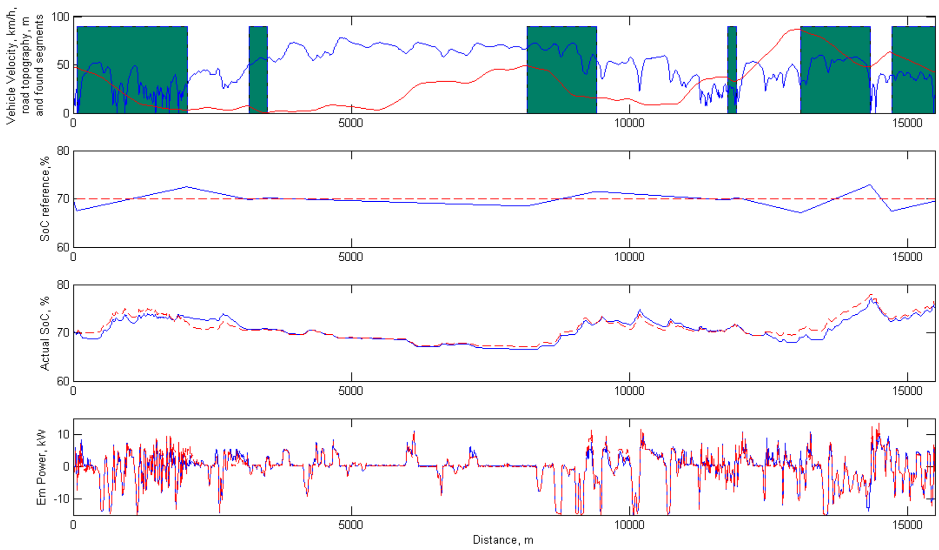


Figure 6.14: Comparison between simulation runs with and without algorithm, measured drivecycle, 1.2 kWh, full length horizon.

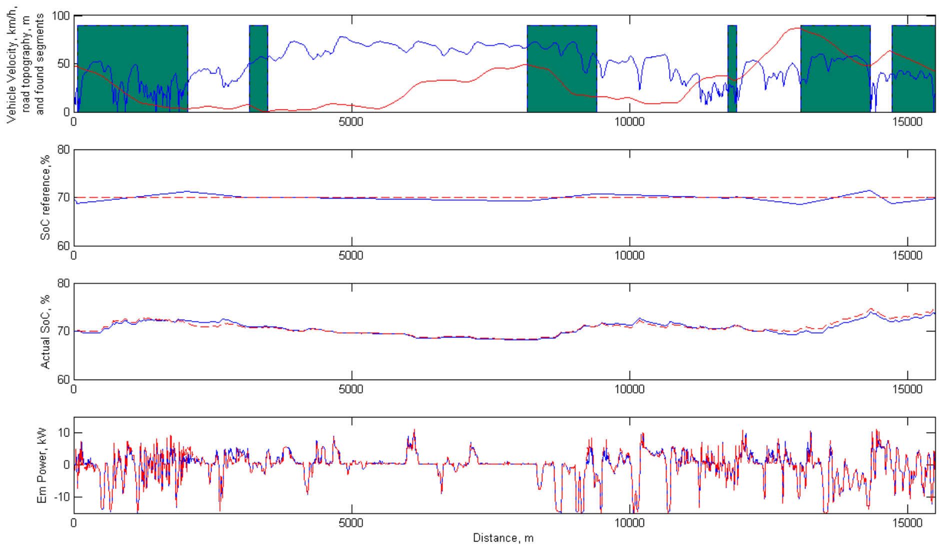


Figure 6.15: Comparison between simulation runs with and without algorithm, measured drivecycle, 2.4 kWh, full length horizon.

6.2.1.2 Measured drivecycle with horizon length 1050 meters

Battery capacity:	0.6	1.2	2.4	kWh
FC w/o. algorithm	0.419	0.397	0.381	l/10km
FC w. algorithm	0.419	0.398	0.383	l/10km
Difference	0	+0.18	+0.74	%

Table 6.6: Fuel consumption comparison, measured drivecycle with horizon length 1050 m.

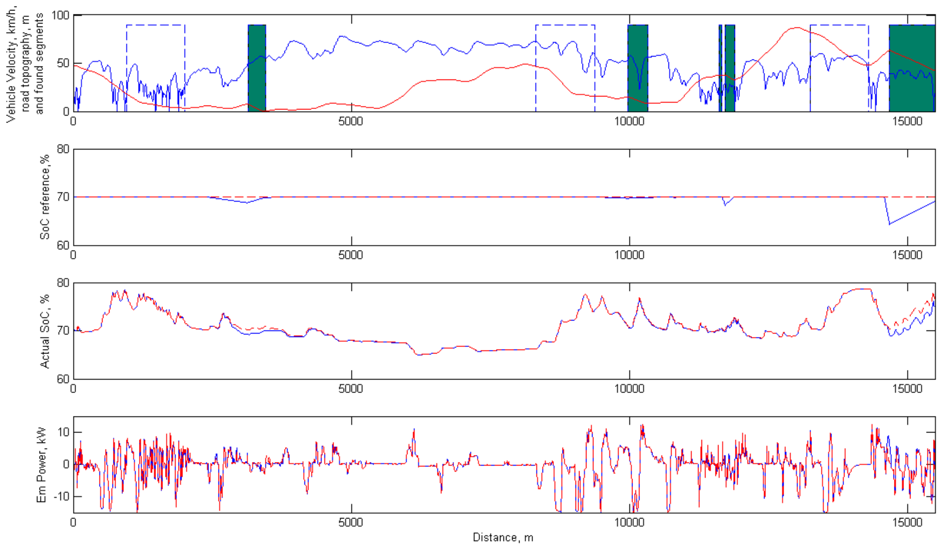


Figure 6.16: Comparison between simulation runs with and without algorithm, measured drivecycle, 0.6 kWh, short horizon.

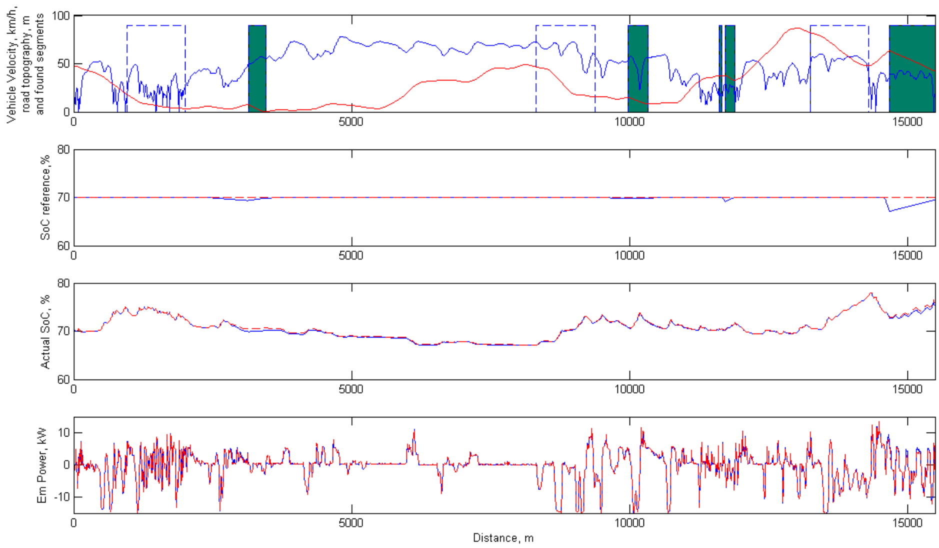


Figure 6.17: Comparison between simulation runs with and without algorithm, measured drivecycle, 1.2 kWh, short horizon.

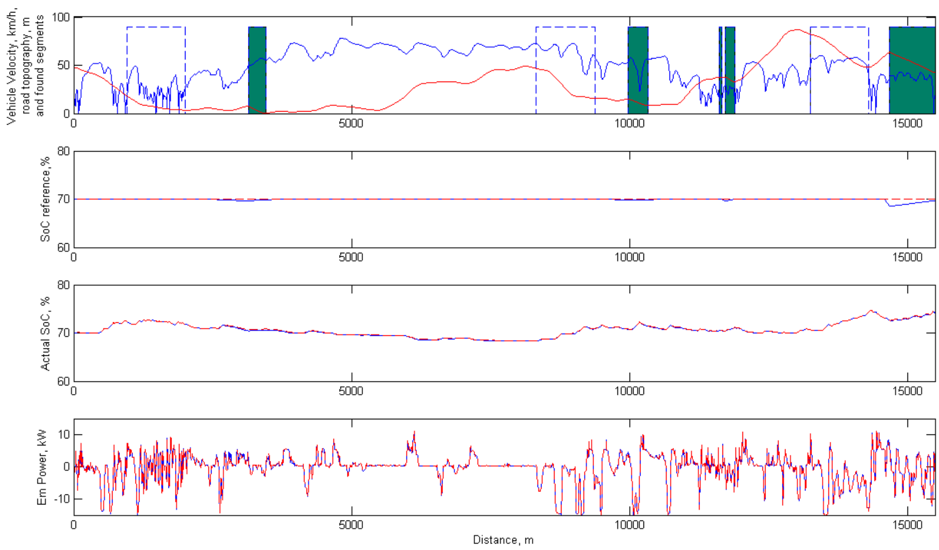


Figure 6.18: Comparison between simulation runs with and without algorithm, measured drivecycle, 2.4 kWh, short horizon.

7 Discussion

7.1 Results

Generally, the developed algorithm functions as intended with the only inputs being future altitude and estimated mean vehicle velocity. The fuel consumption is lowered for all cases when having complete knowledge about the route. Since the algorithm creates a SoC reference which intends to lower the actual SoC before a regeneration segment, the use of the EM is more extensive than without the algorithm. Inside the segment, the SoC reference is raised making the HCU favour using the electric motor as a generator. In drivecycles with an aggresive topographic profile, e.g test drivecycle 2, and having constant SoC-refrence the upper battery constraint is violated more often meaning that more energy will be wasted in the friction brakes. Using the algorithm, the vehicle is allowed to regenerate more energy.

Fuel reductions are generally larger for smaller batteries, corresponding well with [18]. A deviation can be seen for the larger battery on the measured drivecycle with complete knowledge about the route, see table 6.5 and figure 6.15. In this case the larger battery achieves a considerably larger fuel reduction than the smaller batteries. This might be a result of the simple start/stop logic described in section 4.2.6, where the engine is forced to turn on if the state-of-charge falls below a treshold. For the larger battery this threshold is never violated, allowing the vehicle to run in electric mode more often. As can be seen in figure 6.13 the state-of-charge is occasionally forced below the threshold, turning the engine on. Modifying the start/stop criterias would potentially give different results.

A shorter horizon does not give fuel reductions of the same size as when having a complete knowledge of the route, compare tables 6.1 and 6.2 for example. This can be a consequence of different factors. First, since the horizon is short only short segments can be seen within the scope. If the end of a larger segment is discovered while the vehicle is still inside the segment, that segment is discarded, see section 4.3.4. This implies that all the potential energy that is held in the larger segment cannot be estimated and not taken into consideration. Second, if a segment is found inside the short horizon the vehicle might be very close to the segment while still requesting a large SoC reduction. The vehicle might not be able to process the proposed SoC reduction within the short timeframe and thus not using the electric motor as much as desired. As can be seen in section 6.2.1.2, a small increase in fuel consumption was obtained. This could be a consequence of not estimating the energy content within segments correct. The SoC might have been lowered too much, which could have forced the ICE to charge the battery instead of obtaining enough free energy in the segment.

An important note is that even though three battery sizes were used, only one electric motor where modeled and used in the simulations. This can be a limiting and distorting factor when looking at the results. The electric motor was originally sized to match the smallest battery size, 0.6kWh. A more reasonably comparison between the batteries could have been to use larger electric motors in conjunction with the larger battery sizes. A larger electric motor would potentially been able to solely propel the vehicle more often and during longer periods of time when having a larger battery which might have given larger fuel reductions for the larger batteries. However, due to time limitations this was not possible to investigate during the course of the project.

As mentioned in section 3.1, the original state-of-charge reference was assumed to be a constant value. More accurate would have been to have a velocity dependant state-of-charge reference where the SoC would have been lowered during high speeds with the assumption that energy could be regenerated when bringing the vehicle to a stand-still. This would probably not have had any effects when simulating on the two test cycles, see figures 3.4 and 3.5, since the velocity is constant. On the measured drivecycle it could however influenced the results.

One of the design goals of the algorithm was to bring back the State-of-Charge reference to the original reference signal, hence creating only a deviation from the original reference. This behavior becomes clear when using a short horizon, where only one or a few segments are found. Using the full horizon length this behavior can be seen at the end of the driving mission.

An important note is that the results might change depending on the HCU ability to follow a given reference signal. Even though the algorithm works satisfactory, it is still the HCU that in the end determines the distribution of torque between the ICE and the EM. If the HCU is tuned to penalize deviations from the SoC reference differently, the results would differ.

7.2 Possible improvements

In order to improve the robustness of the algorithm, a better prediction is needed for the expected vehicle velocity. The largest error in this assumption is that a driver will seldom drive according to the set speed limit. This could be a combination of speed limits, stop-signs, road-crossings, railway-crossings etc. Also, the ability to use traffic information [12], driver analysis [17] and some way measure the probability of the prediction could be beneficial. If the prediction is likely based on wrong assumptions the calculated SoC reference could be weighted and limited.

The way segmentation is implemented in this project could result in large discrepancy from actual available number of segments. This is due to the fixed horizon length which always give the same amount of data to process, if set to short the algorithm cannot distinguish segments longer than the horizon length. Another approach towards this could be to set the number of segments fix and proceed the search until the number is satisfied. Since the uncertainty of the predicted segments increase with distance this approach should be used with care.

The algorithm (see section 4.3.2) only produces segments if a negative slope is found, thus no segments will be found on flat road sections. One more rule could be added to set segments of tabular ground where the allowed speed decreases. Also, the assumption that recuperation segments ends if the allowed speed is increased is very rough. A better approach could be to measure how large the increase in velocity is and start a new segment again when the actual velocity is assumed to have reached the estimated new velocity.

7.3 Further research and ideas for other Master Thesis's

Study how the information should be exchanged from the GPS/route planning system and the hybrid optimization system. How does the transfer protocols used in the car affect the ability to create a detailed prediction?

Where should the prediction be done, in the GPS/planning system, the hybrid optimization system or standalone only feeding the hybrid optimization system with one signal?

Create a probabilistic route prediction system where attributes as road crossings, traffic information and legal set speed are analyzed. This could generate a better estimation of upcoming driving behaviour.

References

1. Ambühl D. and Guzzella, L. (2009) ”Predictive Reference Signal Generator fo Hybrid Electric Vehicles”, *IEEE Transactions on Vehicular Technology, Vol. 58, No. 9.* DOI: 10.1109/TVT.2009.2027709
2. Beck, R., Bollig, A. and Abel, D. (2007) ”Comparison of Two Real-Time Predictive Strategies for the Optimal Energy Management of a Hybrid Electric Vehicle”, *Oil & Gas Science and Technology - Rev. IFP, Vol. 62, No. 4, pp 635-643.* Institut français du pétrole. DOI: 10.2516/ogst:2007038
3. *Garmin.* http://www8.garmin.com/aboutGPS/ (2010-09-09).
4. Guzzella, L. and Sciarretta, A. (2007) *Vehicle Propulsion Systems: Introduction to Modeling and Optimization. 2nd ed.* Berlin: Springer. ISBN: 9783540746911
5. Haifeng, D., Xuezhe, W. and Zechang, S. (2009) ”A new SOH prediction concept for the power lithium-ion battery used on HEVs”. *Vehicle Power and Propulsion Conference, VPPC ’09.* DOI: 10.1109/VPPC.2009.5289654
6. Hajimiri, M.H. and Salmasi, F.R. (2006) ”A Fuzzy Energy Management Strategy for Series Hybrid Electric Vehicle with Predictive Control and Durability Extension of the Battery”. *Conference on Electric and Hybrid Vehicles, ICEHV ’06.* DOI: 10.1109/ICEHV.2006.352279
7. Hellström, E. (2010) *Look-ahead Control of Heavy Vehicles.* Ph.d. thesis: Dissertations No 1315, Department of Electrical Engineering, Linköping University, Institute of Technology, Linköping, Sweden. ISBN: 9789173933896
8. Johannesson, L. (2009) *Predictive Control of Hybrid Electric Vehicles on Prescribed Routes.* Ph.d. thesis: Ny Serie 2944, Department of Signals & Systems, Chalmers University of Technology, Göteborg, Sweden. ISBN: 9789173852630
9. Katsargyri, G.-E. et al. (2009) ”Optimally controlling Hybrid Electric Vehicles using path forecasting”. *American Control Conference, ACC ’09.* DOI: 10.1109/ACC.2009.5160504
10. Kessels, J.T.B.A., van den Bosch, P.P.J. (2007) ”Electronic Horizon: Energy Management using Telematics Information”. *Vehicle Power and Propulsion Conference, VPPC 2007.* DOI: 10.1109/VPPC.2007.4544190
11. *Mathworks.* http://www.mathworks.com/products/ (2010-08-04).

12. Montazeri-Gh and M., Asadi, M. (2008) ”Application of Vehicle Telematic System in Fuzzy-Based HEV Control”. *3rd International Conference on Information and Communication Technologies: From Theory to Applications.* ICTTA 2008. DOI: 10.1109/ICTTA.2008.4529962

13. GPS. *Nationalencyklopedin web.* http://www.ne.se/lang/gps/ (2010-09-09).

14. Piller, S., Perring, M. and Jossen, A. (2001) ”Methods for state-of-charge determination and their applications”, *Journal of power sources, Vol. 96, pp 113*-120. Center for Solar Energy and Hydrogen Research, Ulm, Germany.

15. Richton B et al. (2001) ”Assisted GPS for Wireless Phone Location - Technology and Standards”. *Next Generation Wireless Networks*, editor: Sirin Tekinay, Dordrecht, Kluwer Academic Publishers Group. ISBN: 0792372409

16. Vanderbei, R. (2008) *Linear Programming, foundations and extentions. 3rd edition* New York: Springer. ISBN: 9780387743882

17. Yokoi, Y. et al. (2004) ”Driving pattern prediction for an energy management system of hybrid electric vehicles in a specific driving course”. *30th Annual Conference of Industrial Electronics Society, IECON 2004.* DOI: 10.1109/IECON.2004.1431842

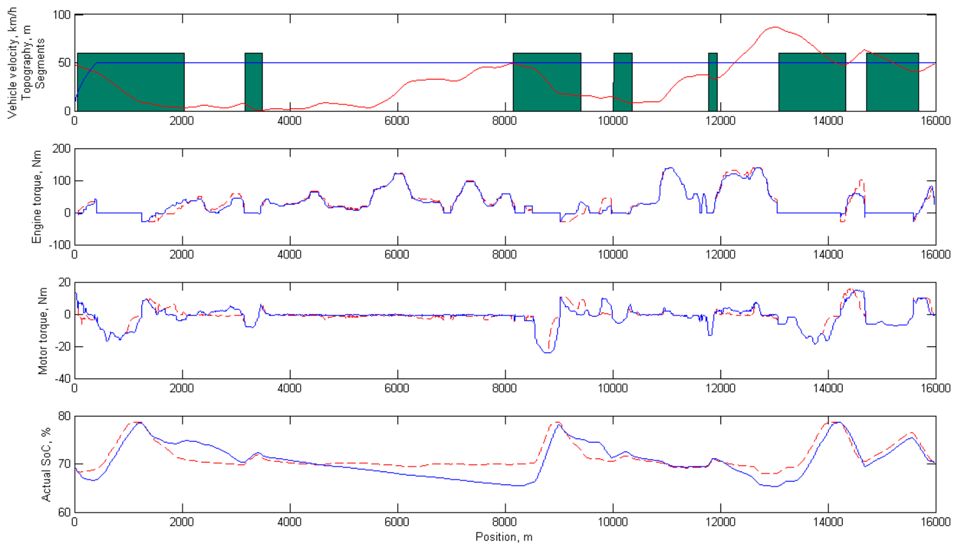
18. Zhang, C. et. al. (2010) ”Role of Terrain Preview in Energy Management of Hybrid Electric Vehicles”, *IEEE Transactions on Vehicular Technology, vol. 59, no 3.* DOI: 10.1109/TVT.2009.2038707

Bibliography

- Andersson, J. (2001) *Route Adaptation of Charge Strategies for Hybrid Vehicles.* Ph.d. thesis: Ny Serie 1779, Department of Machine and Vehicle Systems, Chalmers University of Technology, Göteborg, Sweden. ISBN: 9172910968
- Sciarretta A., Back, M. and Guzzella, L. (2004) ”Optimal Control of Parallel Hybrid Electric Vehicles”, *Transactions on Control Systems Technology, vol 12, pp 352-363.* DOI: 10.1109/TCST.2004.824312

Appendix A

Torque distribution, test drivecycle 1



Torque distribution, test drivecycle 1, 0.6 kWh, full horizon length