Energy optimization tool for mild hybrid vehicles with thermal constraints

Tomas Tamilinas
Chitranjan Singh

LUND
UNIVERSITY

Department of Automatic Control

Academic supervisor, Lund University: Pontus Giselsson.
Academic co-supervisor, Chalmers University of Technology: Nikolce Murgovski.
Supervisor, Volvo Car Group: Mitra Pourabdollah.
Supervisor, Volvo Car Group: Martin Sivertsson.

Written at Volvo Car Group.
Thesis written in collaboration with a student from KTH.

MSc Thesis TFRT-6083
ISSN 0280–5316

Department of Automatic Control
Lund University
Box 118
SE-221 00 LUND
Sweden

Printed in Sweden by Tryckeriet i E-huset.
Lund 2019

Abstract

The current global scenario is such where impact on the environment is becoming a rising concern. Global automotive manufacturers have focused more towards hybrid and electric vehicles as both more aware customers and governmental legislation have begun demanding higher emission standards. One of the many ways that Volvo Car Group approaches this trend is by mild hybridization which is by assisting the combustion engine by a small electric motor and a battery pack. A smart energy management strategy is needed in order to get the most out of the benefits that hybrid electric vehicles offer. The main objective of this strategy is to utilize the electrical energy on-board in such a manner that the overall efficiency of the hybrid powertrain becomes as high as possible. The current implementation is such that the decision for using the on-board battery is non-predictive. This results in a sub-optimal utilization of the hybrid powertrain. In this thesis, a predictive energy optimization tool is developed to maximize the utility of hybridization and the practical implementation of this tool is investigated. The optimization considers both the capacity as well as the thermal load constraints of the battery. The developed optimization tool uses information about the route ahead together with convex optimization to produce optimal reference trajectories of the battery states. These trajectories are used in a real-time controller to determine the battery use by controlling the adjoint states in the Equivalent Consumption Minimization Strategy equation. This optimization tool is validated and compared with the baseline controller in a simulation environment based on Simulink. When perfect information about the road ahead is known, the average reduction in fuel consumption is 0.99% relative the baseline controller. Several issues occurring in the real implementation are explored, such as the limited computational speed and the length of the route ahead that can be predicted. For this reason the information input to the optimization tool is segmented and the resulting performance is investigated. For a 30 second segmentation of the future route information, the average saving in fuel consumption is 0.13% relative to the baseline controller. It is shown that the main factor limiting the amount of savings in fuel consumption is the introduction of the thermal load constraints on the battery.

Acknowledgements

I would like to thank Volvo Car Group for offering the thesis work, and also the whole team at which I resided in as well. Special thanks to Martin, Mitra and Nikolce for their tutoring and help along the thesis, and also the coolest guy at Volvo, Marcin.

The other thesis workers, including my work-partner for this project, present at the same location throughout the working period, made this journey much more enjoyable. Thank you to: Chitranjan, Ankur, Ansh, Karen and Wilhelm.

And also a special thanks to AriZona drink company, they make fantastic drinks. And also a thanks to the peanut supplier Alesto, their peanuts have been the prime source of calories during the thesis period.

Last but not least, I would like to thank my family, Lolita and Janusz, and the phone calls we have been having every Sunday.

To mom.

Tomas
Gothenburg, June 2019

Contents

Division of labor

- **Introduction**
  Authors: Tomas Tamilinas, Chitranjan Singh
- **Hybrid electric vehicle**
  Authors: Chitranjan Singh, Tomas Tamilinas
- **Vehicle modelling**
  Authors: Tomas Tamilinas, Chitranjan Singh
- **Optimization algorithms**
  Authors: Tomas Tamilinas
  Development: Tomas Tamilinas, Chitranjan Singh
- **Energy Management System**
  Authors: Tomas Tamilinas, Chitranjan Singh
  Development: Chitranjan Singh, Tomas Tamilinas
- **Trajectory tracking**
  Authors: Tomas Tamilinas, Chitranjan Singh
  Development: Chitranjan Singh, Tomas Tamilinas
- **Drive cycle segmentation**
  Authors: Tomas Tamilinas
  Development: Tomas Tamilinas, Chitranjan Singh
- **Drive cycle merging**
  Authors: Tomas Tamilinas, Chitranjan Singh
  Development: Tomas Tamilinas
- **EOCM state estimation learning**
  Authors: Chitranjan Singh, Tomas Tamilinas
  Development MATLAB: Tomas Tamilinas
  Development advanced software: Chitranjan Singh

- **Receding horizon**
  Authors: Tomas Tamilinas, Chitranjan Singh
  Development MATLAB: Tomas Tamilinas
  Development advanced software: Chitranjan Singh
- **Results, analysis and conclusion**
  Authors: Tomas Tamilinas, Chitranjan Singh

List of Figures

List of Tables

Notations

The list describes the annotations that are used in the document.

Abbreviations

DDP	Deterministic dynamic programming
DP	Dynamic programming
ECM	Engine Control Module
ECMS	Equivalent Consumption Minimization Strategy
EM	Electric motor
EMS	Energy Management System
EOCM	Battery overcurrent thermal state
HEV	Hybrid Electric Vehicle
ICE	Internal Combustion Engine
ISG	Integrated Starter Generator
KERS	Kinetic Energy Recovery System
OCM	Overcurrent monitor
PHEV	Plug-in Hybrid Electric Vehicle
RMS	Root mean square
SoC	State of Charge
SoE	State of Energy
VCG	Volvo Car Group

ZOH	Zero-order hold

Battery model parameters and variables

E_{batt}	Total battery energy capacity
$EOCM_{max}$	Maximum value of the EOCM state
I_{batt}	Battery output current
$P_{batt,loss}$	Battery power losses
$P_{batt,max}$	Maximum discharging battery power
$P_{batt,min}$	Maximum charging battery power
P_{batt}	Battery power
P_{ech}	Vehicle electrochemical power output
R_i	Battery inner resistance
$SoC_{horizon}$	Receding horizon state of charge trajectory
SoC_{max}	Maximum value of the battery state of charge
SoC_{min}	Minimum value of the battery state of charge
$SoC_{neutral}$	Neutral charge of the battery
$SoC_{tgt,horizon}$	Terminal state of charge target for receding horizon optimization
U_{batt}	Battery output voltage
U_{oc}	Battery inner voltage

Greek characters

α	Correction term for EOCM state underestimation correction
β	Weight between using the new estimation of α and the previous guess
δ	EOCM state slack variable used in receding horizon
η	Integrated starter generator belt efficiency
λ	Equivalence factor for electrochemical power
μ	Equivalence factor for battery current
ω	Angular velocity of powertrain output shaft

ρ_{air}	Air density
θ	Vehicle inclination angle
ϑ	Weight between using old and new receding horizon trajectory
ζ	Compensation term for converting terminal state of charge error into fuel consumption

Load and actuation parameters and variables

ICE_{on}	Discrete variable dictating whether the combustion engine is turned on or off
ISG_{on}	Discrete variable dictating whether the electric motor is turned on or in standby mode
P_{aux}	Power delivered to auxiliary system of the vehicle
$P_{beltloss}$	Belt losses in the powertrain
P_{brake}	Braking power from the friction brake system
P_f	Power from fossil fuel energy source
$P_{ICE,loss}$	Power loss in the internal combustion engine
$P_{ICE,max}$	Maximum power output of the internal combustion engine
$P_{ICE,mech}$	Mechanical power output by the internal combustion engine
$P_{ICE,min}$	Minimum power output of the internal combustion engine
P_{ICE}	Power output by the internal combustion engine
$P_{ISG,loss}$	Power loss in the electric motor
$P_{ISG,max}$	Maximum power output of the electric motor
$P_{ISG,mech}$	Mechanical power output by the electric motor
$P_{ISG,min}$	Minimum power output of the electric motor
P_{ISG}	Power provided by the integrated starter motor
$P_{mean,horizon}$	Mean power demand during the considered horizon
P_{req}	Required power output at wheels

Vehicle model parameters and variables

A_v	Frontal area of vehicle
c_d	Vehicle coefficient of drag
c_{r0}	Rolling coefficient 0
c_{r1}	Rolling coefficient 1
F_{aero}	Air resistance force
F_{grad}	Gradient resistance force
F_{rr}	Roll resistance force
g	Gravitational constant
J_w	Sum of moment of inertia for all internal components
m_v	Mass of the vehicle
r_w	Vehicle wheel radius
T_{req}	Required torque output at wheels
T_{TR}	Required traction resistance torque
v	Vehicle velocity
v_{next}	Vehicle velocity in next considered segment

1

Introduction

1.1 Background

The automotive market has been established for over a hundred years and the overall products developed and sold by automotive companies have been gradually but steadily changing over time. Internal combustion engines (ICEs) have been the cornerstone for the automotive industry until recently. Globally, the transport industry accounts for about 25% of the world's energy consumption and light-duty vehicles used for passenger transportation account for the biggest part of the overall transportation energy consumption [EIA, 2015]. The new era in the automotive industry, motivated by the heightened environmental focus as well as the growing fossil fuel scarcity, has caused the companies in the market to invest in new technologies. Customers of the automotive companies as well as governmental legislation set increasingly stringent requirements on new vehicles in terms of fuel consumption and emission. One possible solution to satisfy these criteria is by using electric energy. This allows the replacement of fossil fuels as the primary source of energy, and also the possibility for higher efficiency in the powertrain.

There are multiple issues with moving towards complete electric powertrains. One major factor is that the energy density of gasoline and diesel is extremely high compared to the available methods of electric energy storage. Another issue with pure electric vehicles is the recharging duration. These two major factors, amongst others, such as manufacturing cost and added weight, makes the ICE still an attractive option.

The hybrid electric vehicle (HEV) is a step towards fully electrified vehicles. HEVs are vehicles that combine the prevalent technology of ICEs with electric motors (EMs) and batteries. The benefits of the ICE, such as the long range of travel without the need to refuel, the well-covered infrastructure of fuel stations and the possibility to refuel quickly, are retained while the use of an EM aims to reduce overall emissions of the vehicle by exploiting its higher efficiency and the added possibility of recuperating kinetic energy during braking periods.

Volvo Car Group (VCG) seeks to tackle the issue of hybrid vehicles by using technology inspired by Formula One, the Kinetic Energy Recovery System (KERS).

The KERS is a system for recovering a moving vehicle's kinetic energy under braking and storing this energy in an on-board battery. The stored electrochemical energy is subsequently used for short-term propulsion assist. In modern VCG cars an Integrated Starter Generator (ISG) unit is installed which is linked directly to the driveshaft. The ISG is an addition to the starter motor. By including a 48V battery pack, the system can reduce fuel consumption by recuperating energy through regenerative braking and adding a degree of power assist to the engine [Volvo Car Group, 2011].

This configuration gives the customer the benefits of a mild HEV while avoiding most of the aforementioned downsides of the electric vehicle as both the battery and the electric motor are limited in size. However, there is one issue which arises with this setup. The requested power demand from the battery needs to be controlled appropriately due to the capacity and thermal constraints of the battery.

Due to more than one power source in the vehicle, as with all hybrid vehicles, an energy management system (EMS) is essential. The objective of the EMS can be formulated as following: determine the optimal power-split between the different power sources in order to minimize fuel consumption at each time instance. The strategy needs to factor in good driving behaviour and responsiveness.

1.2 Problem formulation

There are many different EMS strategies and the controller can be designed in many ways. The battery packs to be included in VCGs vehicles equipped with the KERS technology are limited in capacity, and the efficiency of the vehicles is therefore highly dependent on where and how the electric energy is used. The aforementioned thermal loads also need to be taken into consideration.

There are different types of HEVs, such as the plug-in hybrid electric vehicle (PHEV) which contain a battery pack that can be recharged from an external power source. As the considered KERS vehicle does not have this option, it must remain charge neutral by the end of any considered drive cycle.

The current EMS for vehicles with the KERS system at VCG is a rule-based logic. The current strategy uses no information about the future route or the current thermal state and thus it will not be optimal in utilizing the electrical energy. An example case of sub-optimality is a vehicle approaching a downhill section which is large enough to refill the on-board battery entirely, from its lowest allowable SoC to its highest SoC. The current EMS can result in a moderate use of the battery, resulting in large thermal loads, and also with a high SoC right before the start of the long downhill section. This results in sub-optimal recuperation during the subsequent downhill section as the amount of energy that can be regenerated is limited by the battery's upper SoC limit, as well as the recharging rate may be limited due to the battery's thermal limits. By using information about the route ahead, such as the predicted speed and inclination angle of the vehicle, the battery could be depleted

before the downhill and then be kept from being utilized so as to reduce the thermal state before the downhill starts. This maximizes the possible recuperation of energy during the downhill section as high currents may be used due to the cooled battery, as well as a large capacity can be recuperated due to the initially drained battery. Information available from, for example, modern navigation systems, could therefore be used to predict the power requirements and use the available electric energy accordingly in order to reduce fuel consumption and emissions.

The objective of this thesis is to design and implement an EMS that, when given vehicle and route information, computes how the electrical energy from the battery should be used in an optimal fashion to minimize fuel consumption. The route information is to be sparsed to reduce the overall computational burden. The problem is characterized by the challenges due to the length of the considered routes, the limited capacity of the battery in the KERS system, and the thermal considerations for battery performance. Moreover, the limited computational power of the processors in a vehicle requires a method that is not computationally demanding and yet provides tangible improvements. The resulting strategy is to be benchmarked against the theoretical optima as well as the current energy management solution.

The output information from the EMS are to be trajectories which describe how the battery should be used. In addition to this, a real-time algorithm is to be developed that decides the actual power-split in order to follow the calculated optimal trajectories.

The overall idea is that that information about drive cycles are sent to a central server by vehicles currently driving. When and how often the data is sent is determined by a segmentation algorithm. The sent data is stored in the central server and merged with data from other vehicles previously driven on same roads. Once a new vehicle enters a certain drive cycle, a request is sent to the central server and there a calculation is done based on the stored information about that particular drive cycle. The calculated trajectories are sent to the vehicle, and in the vehicle the real-time trajectory tracking is performed.

1.3 Related research

Previous work has been done within VCG tackling similar issues. The optimization tool developed in this thesis is based on the results from a previous Master's Thesis as well as from the research and development made by the personnel at VCG working on the same project. The papers listed below are related to similar projects.

- Route Based Optimal Control Strategy for Plug-In Hybrid Electric Vehicles. Almgren, Johan and Elingsbo, Gustav. [Almgren and Elingsbo, 2017]
- Efficient Route-based Optimal Energy Management for Hybrid Electric Vehicles. Berntsson, Simon and Andreasson, Mattias. [Berntsson and Andreasson, 2018]

- Adaptive Control of a Hybrid Powertrain with Map-based ECMS. Sivertsson, Martin and Sundström, Christofer and Eriksson, Lars. [Sivertsson et al., 2011]

1.4 Thesis objective

The goals of this master's thesis are listed below.

- Extend upon the current optimization algorithm, reduce its complexity and integrate the battery thermal state into the optimization calculation.
- Create a real-time algorithm for tracking the trajectories produced from the optimization algorithm in order to create a complete optimization tool. The controller is to be robust and should work on any drive cycle while also considering driveability factors such as high frequency switching of actuators.
- Create an algorithm for sparsing drive cycle input data to the optimization tool.
- Investigate how information from multiple drives on same route can be merged together for future use.
- Measure the drive cycle prediction horizon length influence on fuel consumption.
- Analyze the inflicted limitations on fuel consumption savings due to hardware limits.

1.5 Thesis limitations

The algorithm is originated in MATLAB and the developed algorithms are tested within the MATLAB environment. Advanced simulation software based on Simulink is available for more realistic results. Some results, but not all, are presented in this environment.

- When considering driveability performance, only frequent switching between the ICE and the EM due to aggressive controlling is considered. Other factors, such as acceleration behavior and the throttle response, are neglected.
- Standstills during drive cycles are neglected and are discussed instead.
- The auxiliary load of the vehicle is assumed to be constant and is discussed instead.

1.6 Outline

This thesis report contains the chapters listed below.

- **Hybrid electric vehicle** - This short section includes a general overview of the hybrid the main configurations of existing HEVs and also a description of the VCG KERS powertrain layout.
- **Vehicle modelling** - This section explains how the ICE, ISG and on-board battery are modelled and described mathematically. The gear selection and how additional discrete variables are decided is explained as well.
- **Optimization algorithms** - The algorithm types used in EMS are given a general description and their main traits are brought to attention. The used optimization algorithms in this thesis are explained in more detail.
- **Energy Management System** - Here the structure of the used EMS is shown and different approaches and developments of it are compared.
- **Trajectory tracking** - In this chapter the real-time controller used for following the SoC and EOCM state trajectories is introduced.
- **Drive cycle segmentation** - The used drive cycles and the different segmentation algorithms are presented. Measured data from drive cycles is post-processed for correcting errors.
- **Drive cycle merging** - Different approaches to how the data from multiple drives on the same drive cycle can be merged are explained.
- **EOCM state estimation learning** - An approach to correcting the errors in estimating the EOCM state is shown here.
- **Receding horizon** - The issue of and the solution to not being able to see the entire drive cycle is presented in this chapter.
- **Results** - Chapter containing the results for: different optimization algorithm layouts, segmentation algorithms, global optimization algorithm and real-time tracker, receding horizon influence, merging of trajectories, EOCM state estimation learning and hardware limitations. This chapter also includes discussions surrounding the found results.
- **Conclusion and future work** - The conclusion of the thesis work and suggestions for what could be further developed.

2

Hybrid electric vehicle

2.1 A brief introduction to the hybrid electric vehicle

Hybrid electric vehicles, or HEVS, utilize more than one unique energy reservoir. The most popular combination of energy sources in HEVs, which is also the one used in the studied vehicle, is chemical energy stored in the fuel tank (fossil fuels) and electrochemical energy stored in the battery (electricity). As there are two different sources of energy on-board the vehicle, there needs to be a method for finding the optimal way of using these two. The real-time optimization problem can be stated as the following: at each time instance, determine the power split between the ICE and the EM that together equal the total demanded power output. Mathematically, the statement corresponds to finding, at each t, $P_{ICE}(t)$ and $P_{ISG}(t)$ that satisfies the total power required (P_{req}) in (2.1), while also minimizing some cost function.

$$P_{req}(t) = P_{ICE}(t) + P_{ISG}(t) \tag{2.1}$$

Studied vehicle configuration

There are three main possible configurations of HEV powertrains, these are:

- the parallel-hybrid drivetrain,
- the series-hybrid drivetrain, and
- the combined-hybrid drivetrain.

The parallel-hybrid topology works on the concept that the primary and the secondary prime movers, i.e the ICE and the EM, can provide power simultaneously therefore the maximum torque provided as well as the maximum power provided exceeds the maximum limit of the individual prime movers. The ratio between the output power from the two prime movers can also be combined freely as long as the driver demand (output power at wheels) is satisfied [Guzzella, 2007].

The series topology works by providing the primary and secondary prime movers in series. As opposed to the parallel-hybrid layout, in a series-hybrid layout

only the EM propels the vehicle as the energy stored in a battery can be charged either by regenerative braking or by the ICE through a generator. The idea with this layout is that as the ICE speed is decoupled from the output driveshaft, it can always operate at its most efficient regions. Since the efficiency of the generator, battery and the EM are relatively high, the goal with this layout is to reach lower overall fuel consumption and emission output [Guzzella, 2007].

The combined-hybrid drivetrain takes advantage of the two previous layouts. There are two electric motors on-board, one used for propulsion and one acting as a generator. This layout gives a large freedom of using the different actuators as they are not directly coupled together, but by a power-split device which usually consists of a planetary gear set which helps in operating the EM and the ICE at different speeds [Guzzella, 2007].

The studied KERS vehicle is a parallel configuration mild hybrid with an ISG acting as the EM, the ISG is designed and sized to offer better fuel economy than conventional vehicles. As the layout is of the parallel configuration, the EM can be used in order to push the operating point of the ICE into the more efficient regions. The assist can occur in both directions. If the power output at the wheels and the gear selected is such that the ICE can operate in a more efficient range if it were to produce less power, the EM can provide this extra power so that the ICE can operate more efficiently. This can be done in the opposite direction, in that case the EM acts as a generator and the excessive power produced from the ICE is stored in the battery.

The ISG can provide a fraction of the ICE power. Therefore, the compensation from the ISG is limited. Additionally, the EM can be used as a generator by either using regenerative braking or the ICE to charge the on-board battery. The EM can also operate together with the ICE in order to increase the maximum power output of the vehicle.

3

Vehicle modelling

3.1 Vehicle dynamics

The primary use of the vehicle dynamics model is to calculate the required power demand according to the target speed, altitude and acceleration demands. The power required needs to be enough to overcome the road load and let the vehicle reach the target speed in the allocated time. The calculations are done for each segment of the drive cycle, where each segment can be seen as a zero-order hold (ZOH) of the velocity, inclination and power demand. The power required according to the drive cycle information is calculated as (3.1-3.7) where m_v is the mass of the vehicle, r_w the vehicle wheel radius, J_w the sum of moment of inertia for all internal components in the vehicle, v_{next} the vehicle velocity in the next considered segment, v the current vehicle velocity, Δt the current segment duration, ρ_{air} the air density, c_d the coefficient of drag, A_v the frontal area of the vehicle, g the gravity constant, c_{r0} and c_{r1} rolling coefficients and θ the inclination angle of the vehicle.

$$F_{aero} = \frac{1}{2}\rho_{air} c_d A_v v^2 \tag{3.1}$$

$$F_{roll} = m_v g(c_{r0} + c_{r1} v)\cos\theta \tag{3.2}$$

$$F_{grad} = m_v g \sin\theta \tag{3.3}$$

$$T_{TR} = (F_{aero} + F_{roll} + F_{grad}) r_w \tag{3.4}$$

$$a = \frac{v_{next} - v}{\Delta t} \tag{3.5}$$

$$T_{req} = (m_v r_w + \frac{J_w}{r_w})a + T_{TR} \tag{3.6}$$

$$P_{req} = T_{req}\frac{v}{r_w} \tag{3.7}$$

3.2 Internal combustion engine

The losses in the ICE are a function of engine torque and speed, which can be modelled as piecewise affine in order to reduce the computational demand by the optimization tool as well as to help in modelling the vehicle components as convex functions. The affine functions are described as (3.8), where k represents the set of linear equations for a specific angular velocity of the ICE, ω. The linearization and the values of the constants A and B are calculated by the Least mean squares method and is done by personnel at VCG following the theory from [Murgovski et al., 2012].

$$P^k_{ICE,loss}(\omega) = A^k(\omega)P_{ICE,mech} + B^k(\omega) \tag{3.8}$$

Since the power loss as a function of the power output is convex and approximated linearly, the actual loss in the component is found as the maximum value of all the linear candidates. As can be seen in Figure 3.1, the found value of the loss differs slightly from the real value, this deviation is due to the limited amount of affine functions used in the approximation. The function to find the matching power loss is shown in (3.9) below.

$$P_{ICE,loss}(\omega) = \max_k (A^k(\omega)P_{ICE,mech} + B^k(\omega)) \tag{3.9}$$

The relation can be stated as

$$P_{ICE,loss} = f(P_{ICE,mech}, \omega). \tag{3.10}$$

The output of the ICE is limited in both directions as the negative power limit is a result of the engine braking characteristics. The limits are formulated as (3.11).

$$P_{ICE,min}(\omega) \leq P_{ICE,mech} \leq P_{ICE,max}(\omega) \tag{3.11}$$

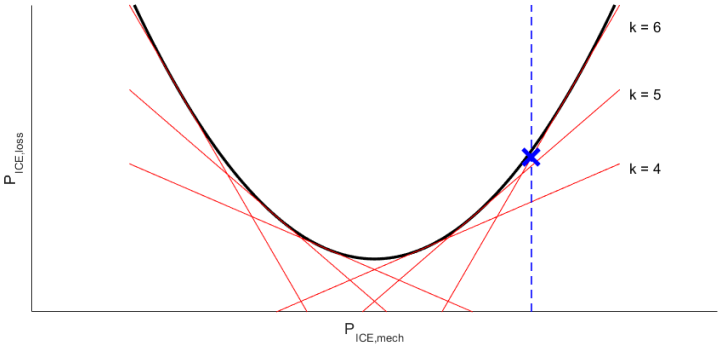

Figure 3.1 Linear approximation of the ICE power losses with $k = [1,6]$. The maximum of all the affine functions at the current $P_{ICE,mech}$ (dashed line) in this case is $k = 6$.

3.3 Integrated starter generator

The function for the power losses of the ISG based on the output power is modelled and calculated in the same manner as for the ICE in the previous section due to the same convex characteristics in the losses. The calculation of the power losses for the ISG is given by (3.12), the linearization is done as before, by personnel at VCG following the theory from [Murgovski et al., 2012]. The effect of linearization and the uncertainty of this approximation is unknown.

$$P_{ISG,loss}(\omega) = \max_k (A^k(\omega)P_{ISG,mech} + B^k(\omega)) \tag{3.12}$$

The relation can, as for the ICE, be stated as (3.13)

$$P_{ISG,loss} = f(P_{ISG,mech}, \omega). \tag{3.13}$$

The losses due to the transmission belt depend on the direction of power transmission between the ISG and the output shaft. The belt efficiency is denoted by η.

$$P_{beltloss} = \begin{cases} P_{ISG,mech}(1-\eta), \text{if } P_{ISG,mech} \geq 0. \\ P_{ISG,mech}(1-\frac{1}{\eta}), \text{if } P_{ISG,mech} < 0. \end{cases} \tag{3.14}$$

Discrete ISG losses

The ISG power losses are not solely dependent on the required ISG power output. There is one additional factor that affects the ISG power loss and that is the discrete variable that dictates whether the ISG is turned on or in standby mode. When the power output demand from the ISG is zero, the ISG EM still consumes power as it

is still connected to the powertrain. By decoupling the ISG and going to stand by mode, the power losses become significantly lower than the losses occurring from the ISG being turned on but not providing any power. This characteristic of the ISG can be seen in Figure 3.2. The switch between the modes happens quickly and is considered lossless.

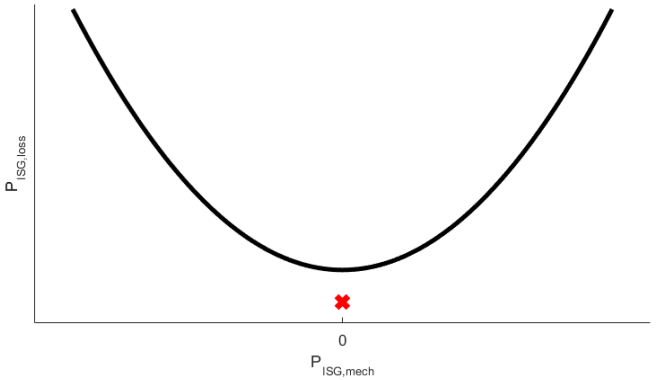

Figure 3.2 ISG losses as a function of ISG power output. When the ISG is in standby mode the power losses are lower than when it is turned on at $P_{ISG,mech} = 0$.

The ISG, like the ICE, has both a positive and a negative power limitation. These are due to the maximum output and recharging capabilities of the EM. These limitations are given as (3.15).

$$P_{ISG,min}(\omega) \leq P_{ISG,mech} \leq P_{ISG,max}(\omega) \tag{3.15}$$

3.4 Neglected effects and losses

As can be seen in the power requirement calculation (3.1-3.7), the effects of inertia in the powertrain components are neglected. Furthermore, friction losses in subcomponents in the vehicle are also neglected and thus the calculated expected power demand throughout a drive cycle should be underestimating the values for a real vehicle. The overall underestimation of the power demand is, however, assumed to be relatively small as the power demanded to overcome road loads and to achieve desired acceleration of the vehicle is comparably larger.

The discrepancy in power calculations is discussed at the end of the report and a suggestion for a simple compensation is made.

3.5 Battery model

The battery used in the vehicle is modelled as a Thévenin's equivalent circuit, as seen in Figure 3.3. The open-circuit voltage (U_{oc}) and the inner resistance (R_i) of the battery are known and are approximated as piecewise affine functions of the battery SoC in order to reduce modelling complexity. In this thesis U_{oc} and R_i are constant in the operating region of the battery. The data for the linear approximations is calculated by the Least mean squares method.

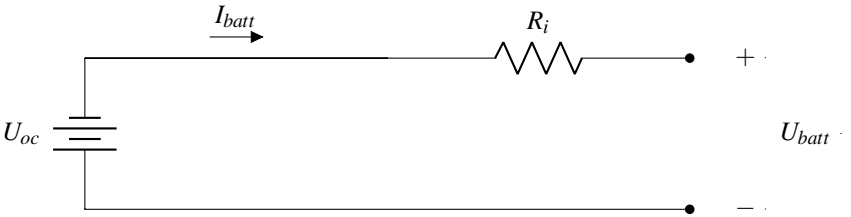

Figure 3.3 Used Thévenin's equivalent circuit model of the on-board battery.

The output current from the battery needs to be calculated as a function of the used power in order to obtain the energy balance. By using Kirschoff's law on the battery model, the output voltage from the battery can be calculated as (3.16).

$$U_{batt} = U_{oc} - I_{batt} R_i \tag{3.16}$$

The used battery power can now be calculated as

$$P_{batt} = I_{batt}(U_{oc} - I_{batt} R_i). \tag{3.17}$$

Solving the equation (3.17) for the current gives the output current from the battery as a function of the used power, this is given by (3.18).

$$I_{batt} = \frac{U_{oc} - \sqrt{U_{oc}^2 - 4R_i P_{batt}}}{2R_i} \tag{3.18}$$

The electrochemical power in the vehicle is defined as

$$P_{ech} = P_{batt} + P_{batt,loss}, \tag{3.19}$$

where $P_{batt,loss}$ is defined as

$$P_{batt,loss} = R_i I_{batt}^2. \tag{3.20}$$

The output power from the battery is limited by the hardware. The maximum absolute value of the power is dependent on the sign (different capacity for charging and recharging), see (3.21).

$$P_{batt,min} \leq P_{batt} \leq P_{batt,max} \tag{3.21}$$

The battery power demand is given by the sum of the demand from auxiliary systems on-board and both the ISG power demand and its power losses. The expression is given by

$$P_{ech} = P_{aux} + P_{ISG,mech} + P_{ISG,loss} + P_{batt,loss}, \tag{3.22}$$

where the battery power loss, $P_{batt,loss}$, occurs due to the inner resistance. The auxiliary power in the vehicle, P_{aux}, is assumed to be constant.

Since U_{oc} is assumed constant in the operating region of the battery, the energy balance in the battery is given by (3.23)

$$\Delta SoC = \frac{-I_{batt}U_{oc}}{E_{batt}}\Delta t, \tag{3.23}$$

where E_{batt} is the battery's total energy capacity.

Battery thermal load model

The battery used in the vehicle not only has SoC and power limits, but also thermal limitations. The thermal load is represented as another state. The state is involved in the overcurrent monitoring system and the state will be written as EOCM where OCM stands for overcurrent monitoring. The state can be described as (3.24) where f_1 and f_2 are functions given by the battery supplier. The next evolution of the EOCM state in the considered discrete time step is dependent on the state itself and the utilized current in the previous step.

$$EOCM_{k+1} = f_1(EOCM_k) + f_2(I^2_{batt,k}) \tag{3.24}$$

The relation in (3.24) can be interpreted as a simple thermal state equation with a dissipative part, f_1, depending on the current state, and a heating part, f_2, due to currents in the battery.

The thermal limitations for the battery are defined by an upper threshold value of the EOCM state. This constraint is given by

$$EOCM \leq EOCM_{max}. \tag{3.25}$$

The state of EOCM can not be negative as the value 0 represents a "completely cooled" battery. The model (3.24) is not questioned nor verified in this thesis, it is assumed to be correct and the set threshold on this state is to be strictly followed.

3.6 Auxiliary power

The Auxiliary power of the vehicle, P_{aux}, in this thesis is assumed to be constant.

3.7 Gear selection

The selection of gear for the studied vehicle's transmission is determined by a map and is outside the scope of this thesis.

4

Optimization algorithms

4.1 Overview

In all types of hybrid vehicles there needs to be a controller that determines how the power demand is to be supplied from the combination of multiple power sources on-board. The main objective of this controller is to minimize the overall energy use while also considering driveability factors and hardware constraints.

In this section an overview is given of how different optimization methods for the power-split between two prime movers are classified and compared. Furthermore, the three optimization algorithms used in the thesis: dynamic programming, Equivalent Consumption Minimization Strategy and convex optimization, are described.

4.2 Optimization methods

There is a vast amount of research done on designing algorithms that have the objective to minimize fuel consumption in HEVs by deciding when and how much the energy from different energy stores on-board should be used. A common way of categorizing the different algorithms is by grouping them according to their approaches [Zhang et al., 2015]. This categorization is shown in Figure 4.1.

The rule-based energy management strategy is characterized by using a set of rules to determine the power-split in order to meet the requirements of the driver while operating within the constraints of the hardware limitations [Alagarsamy and Moulik, 2018]. Although the rule-based EMS cannot, with a realistic set of rules, obtain the global optimal solution for minimizing fuel consumption, it is still relevant due to easy implementation in practice. Additionally, the hardware requirement is low as the controller does not demand significant calculations due to only using a lookup table [Alagarsamy and Moulik, 2018]. Nearly all of mass-produced hybrid electric vehicles use rule-based EMS [Guzzella, 2007].

In optimization-based EMS the goal of the controller is to minimize a cost function that includes the vehicle fuel consumption, emissions, and more. As this thesis

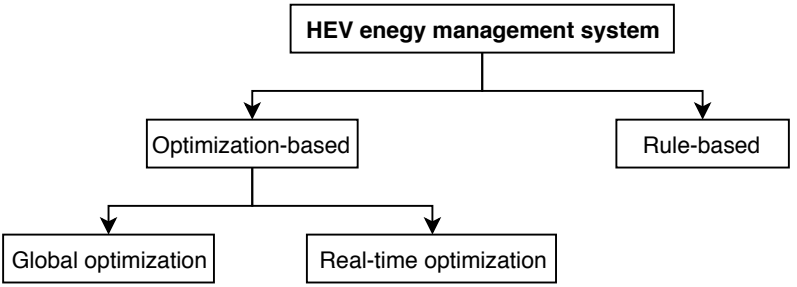

Figure 4.1 EMS categorization based on the approaches taken.

only focuses on the fuel consumption of the vehicle, it is assumed that the cost function contains only the fuel consumption. What makes the optimization-based strategies more efficient than the rule-based strategies is that an optimal power-split can be found for any combination of the input. The goal of these strategies is to determine the power-split while also considering, directly in the optimization, the fuel consumption and the component thresholds [Alagarsamy and Moulik, 2018]. The optimization-based EMS contains two subcategories, real-time and global optimization strategies.

In the real-time optimization EMS the cost function is dependent mainly on the present state of the system parameters, as it can also include history, making this a causal controller. Due to the instantaneous optimization nature of the real-time optimization EMS, the resulting power-split is globally sub-optimal. Even though the optimal power-split is found for each instant throughout a drive cycle, the optimal use of the battery over the entire drive cycle is not obtained as the algorithm has no chance to prepare and adapt to future events [Alagarsamy and Moulik, 2018].

In global optimization EMS the optimization for one instant is replaced by a global optimization which requires knowledge of the entire drive cycle. This helps in determining the use of the different energy sources according to future events [Alagarsamy and Moulik, 2018]. The global optimization EMS is non-causal but can be combined with real-time optimization algorithms in order to obtain an optimal solution to the power-split. The theoretically lowest possible fuel consumption can be obtained since the algorithm both plans and prepares for future events, and optimizes the instantaneous power-split. Due to computational complexity of these algorithms, these methods are not easily implementable in practice [Wirasingha and Emadi, 2010].

4.3 Deterministic dynamic programming

Dynamic programming (DP) is a common technique used for solving discrete-time optimal control problems. DP has the advantage of always finding the optimal con-

trol input evolution that minimizes a cost function [Zhang et al., 2015]. When the cost function does not contain any random and unknown disturbances the technique is usually called deterministic dynamic programming (DDP). The limitation of this algorithm is that it suffers from the curse of dimensionality. The advantage remains that every system irrespective of being convex or not can be optimized. These characteristics of DP result in it being preferred for complex systems containing complex states but not for systems with a high number of states. [Guzzella, 2007; Zhang et al., 2015]

4.4 Equivalent Consumption Minimization Strategy

The Equivalent Consumption Minimization Strategy (ECMS) is a common type of real-time optimization algorithm that is widely used in the area of EMS. The strategy is derived from the Pontryagin's minimum principle and the resulting optimization problem is an instantaneous minimization of a Hamiltonian H. For hybrid vehicles utilizing both fossil fuels and electric energy, the Hamiltonian is generally defined as (4.1) where P_f is the power from a fossil fuel source, λ the equivalence factor, and P_{ech} the electrochemical power output from the battery [Paganelli et al., 2001].

$$H = P_f + \lambda P_{ech} \tag{4.1}$$

The Hamiltonian can be interpreted as the sum of the cost of power from fuel and an equivalent cost of power from stored electric energy. Since the instantaneous total power demand is known, how it is supplied by both the fuel and electric energy in the cheapest way is defined by minimizing the Hamiltonian as (4.2). Since the fuel and electrochemical power can not be directly compared in terms of cost, the equivalence factor is used to described the price for using electrical energy in terms of fuel [Guzzella, 2007; Böhme and Frank, 2017]. It is important to note that P_{ICE} and P_{ISG} include efficiency losses, thus the ECMS algorithm uses models of the ICE and ISG to split the torque in such a manner that the ICE and ISG together are used as efficiently as possible.

$$[P_{ICE}, P_{ISG}] = \arg\min H \tag{4.2}$$

The equivalence factor in ECMS corresponds to the adjoint state in classical optimal control theory. Under the assumptions that the open-circuit voltage and internal resistance of the battery do not depend on the SoC of the battery, and also that the SoC thresholds are not reached, the value of the optimal equivalence factor becomes constant. The optimization problem is thus reduced to determining the constant value for λ [Guzzella, 2007].

The value of the equivalence factor can be guessed with previous experience depending on the future drive cycle characteristics. The use of the battery, i.e. the

SoC trajectory, will however begin to deviate away from the expected result and eventually cross either the high or low thresholds for SoC. This is due to the fact that the battery use is sensitive to small changes in the equivalence factor. As shown in [Berntsson and Andreasson, 2018], a lower guess of the equivalence factor will cause the cost of using the battery being too cheap and thus the battery will be depleted during the observed drive cycle, and conversely if the guess is too high the cost of using the battery will be too steep and the battery will reach its maximum capacity threshold.

In the studied KERS vehicle layout the SoC of the battery is to be charge neutral meaning that the SoC at the end of drive cycles should be near the initial SoC value. The optimization problem of finding the constant value for λ that yields charge neutrality is possible only if the entire future drive cycle is known, and the computation of the correct equivalence factor is then commonly done numerically [Larsson, 2014]. A drawback with using ECMS alone is that the low and high thresholds of the SoC may be violated even with an equivalence factor which is not modified by using a feedback from the current SoC [Guzzella, 2007].

4.5 Convex optimization

A general mathematical optimization problem has the form

$$\begin{aligned} &\text{minimize} && f_0(x) \\ &\text{subject to} && f_i(x) \leq b_i, \;\; i = 1,...,m, \end{aligned} \tag{4.3}$$

where the vector $x = (x_1,...,x_n)$ is the variable to be optimized. The function $f_0 : \mathbf{R}^n \to \mathbf{R}$ is called the objective function, the functions $f_i : \mathbf{R}^n \to \mathbf{R}, i = 1,...,m$, the inequality constraint functions and $b_1,...,b_m$ the constraint bounds of the optimization. A convex optimization problem is an optimization problem in which the objective and constraint functions are convex, meaning that they satisfy (4.4) for all $x, y \in \mathbf{R}^n$ and all $\alpha, \beta \in \mathbf{R}$ with $\alpha + \beta = 1, \alpha \geq 0$ and $\beta \geq 0$. [Boyd and Vandenberghe, 2004]

$$f_i(\alpha x + \beta y) \leq \alpha f_i(x) + \beta f_i(y) \tag{4.4}$$

The convex optimization algorithm does not have the possibility to solve discrete problems such as the discrete ISG variable described in Section 3.3. The convex optimization formulation can however be used to solve the power-split and also be combined with DDP for the discrete ISG variable as done in [Tobias et al., 2014; Berntsson and Andreasson, 2018]. The losses in the vehicle can be formulated as convex functions (see (3.9) and (3.12)), and the battery and other component limitations in the vehicle can be formulated into the constraints described by (4.3). The hardware constraints can therefore be handled by using convex optimization which yields an EMS with a global optimal solution [Murgovski et al., 2012].

5

Energy Management System

5.1 Overview

The energy management is divided into the three following subproblems:

1. Data preparation
2. Reference generation
3. Reference following

The first subproblem concerns the effect of the way the data will be sampled. The precise power demand for the vehicle for a specific drive cycle will not be known beforehand, and neither the exact trajectory of its speed and elevation. Furthermore, in order to reduce computational burden, segmentation of the input data to the optimization tool is needed. This segmentation is to reduce input data while still maintaining necessary information about the route. During testing, all of the requested signals can be sampled with a high sampling rate. By calculating the optimal SoC & EOCM state trajectories on this data, and comparing it with a segmented input, the influence on fuel consumption due to the loss of data can be measured. Method of segmentation and the magnitude of sparsity is to be explored and compared.

The second subproblem of the energy management concerns calculating optimal SoC and EOCM state trajectories based on the segmented input data.

The last part concerns the real-time power-split problem. The optimal SoC and EOCM state trajectories that are calculated in the previous step should be followed as they are predictive of future events. At the same time, necessary deviation from the trajectory tracking should be allowed for minimizing fuel consumption as the actual driving profile will not be exactly as predicted from the segmentation due to not only different driver and traffic conditions, but also due to the segmentation itself.

5.2 Mathematical problem formulation

The problem formulation of the global optimization is minimizing (5.1)

$$\min \sum_{i=1}^{T} P(i)_f \Delta t \tag{5.1}$$

subject to (5.2)-(5.13) where k corresponds to a segment and ISG_{on} is the discrete variable as described in Section 3.3.

$$\begin{aligned} P_{req} = {} & P_{ICE,mech} + ISG_{on} P_{ISG,mech} \\ & - P_{brake} - P_{beltloss} \end{aligned} \tag{5.2}$$

$$P_f = P_{ICE,mech} + P_{ICE,loss} \tag{5.3}$$

$$\begin{aligned} P_{ech} = {} & P_{ISG,mech} + P_{ISG,loss} + P_{aux} \\ & + P_{batt,loss} \end{aligned} \tag{5.4}$$

$$P_{ech} = I_{batt} U_{oc} \tag{5.5}$$

$$P_{batt,loss} = R_i I_{batt}^2 \tag{5.6}$$

$$\Delta SoC = -\frac{I_{batt} U_{oc}}{E_{batt}} \Delta t \tag{5.7}$$

$$EOCM_{k+1} = f_1(EOCM_k) + f_2(I_{batt}^2) \tag{5.8}$$

$$0 \leq EOCM \leq EOCM_{max} \tag{5.9}$$

$$SoC_{min} \leq SoC \leq SoC_{max} \tag{5.10}$$

$$P_{ICE,min}(\omega) \leq P_{ICE,mech} \leq P_{ICE,max}(\omega) \tag{5.11}$$

$$P_{ISG,min}(\omega) \leq P_{ISG,mech} \leq P_{ISG,max}(\omega) \tag{5.12}$$

$$P_{batt,min}(SoC) \leq P_{batt} \leq P_{batt,max}(SoC) \tag{5.13}$$

The problem is formulated in discrete-time as the simulations operate in discrete time-steps.

5.3 Optimization structure for trajectory production

When input data (drive cycle information) is available, an optimization algorithm is used to calculate an optimal trajectory for SoC and the EOCM state. This optimization refers to the problem formulation in Section 5.2.

A different Master's Thesis concerning a similar issue, [Berntsson and Andreasson, 2018], utilized convex optimization to produce SoC trajectories. The convex optimization used is based on the findings in [Murgovski et al., 2012]. The structure for the optimization algorithm for this project is shown in Figure 5.1. In the convex optimization, state constraints are incorporated into the computation. There is still a feedback from the convex optimization into DDP, this control law checks whether subsequent optimizations produce approximately the same result. When the solution has converged enough, it exits and the final SoC trajectory is produced.

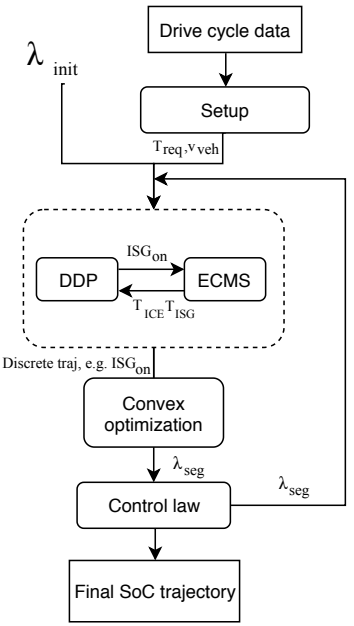

Figure 5.1 First suggested trajectory production layout producing a SoC trajectory to follow by utilizing convex optimization. This layout includes feedback from the convex optimization calculation back to DDP.

The optimization algorithm used in this thesis uses an algorithm similar to the one used in [Berntsson and Andreasson, 2018], where yet again the convex optimizer is based on findings in [Murgovski et al., 2012]. In the new algorithm, the EOCM state constraints are included as constraints in the convex optimizer. Due to a high demand for the optimization algorithm to be efficient, the feedback from the convex optimization and into the DDP is removed, as shown in Figure 5.2. This change causes the algorithm to no longer optimize the discrete ISG variable for fuel efficiency. The ISG discrete variable is instead determined once before the optimization. A specific equivalence factor is used for deciding the discrete variable in order to have a reasonable guess on the variable. The reasoning for this is that the changes in the ISG discrete variable decisions between iterations do not affect the output trajectories all too much. Furthermore, as the toggling of the ISG discrete variable is fast and considered lossless, this decision can be left to a lower level controller determining this variable in real-time. This change in the optimization structure causes a major increase in the overall computational efficiency as without any feedback terms, the optimization is done only once.

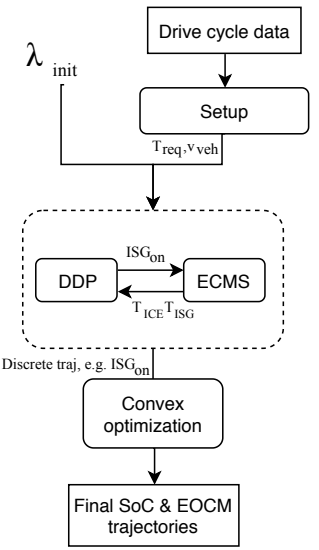

Figure 5.2 Modified optimizer, feedback from the convex optimizer to the DDP-algorithm for obtaining the optimal ISG discrete variable is removed.

The optimization tool is further modified into the structure shown in Figure 5.3. The DDP is removed altogether as the discrete ISG variable is instead decided by a simple rule depending on current vehicle status. The motivation for replacing DDP which optimizes the discrete variable is that by judging from the results given by the DDP, a similar trajectory of the discrete variable can be obtained using simple

rules. Nonetheless, due to this approximation, the result obtained from the modified algorithm will be suboptimal. The idea is to sacrifice a small amount of optimality to obtain significantly increased algorithm efficiency. It is important to note that the actual driving trajectory will not be exactly as expected and input to the optimization algorithm. Instead of optimizing the discrete ISG variable for a certain drive cycle, the deviation from optimality can be reduced later on in the overall optimization algorithm by implementing a decision for the discrete ISG variable in the real-time tracker optimization.

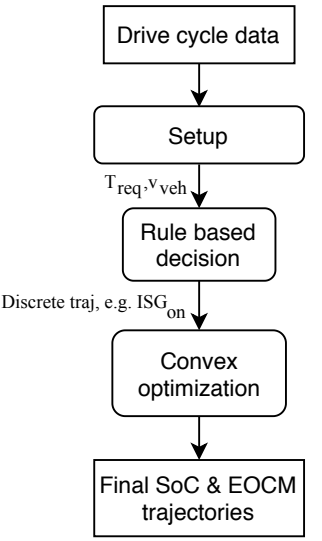

Figure 5.3 Furthermore modified optimizer, the discrete ISG variable is rule-based and DDP is removed.

5.4 Tracking controller

When the global optimization problem is solved, and the trajectories for the SoC and the EOCM states are produced, there is a need for a real-time controller in the vehicle that determines the real-time power-split. This is the objective of the tracking algorithm. The reason why the real-time controller is called a tracker is that the produced SoC and EOCM state trajectories are used in order to control the actual SoC and EOCM states of the on-board vehicle battery. The tracker algorithm controls the vehicles SoC and EOCM states to track the produced trajectories, as they are the near-optimal trajectories based on models.

The algorithm for the tracking is explored more in detail in Chapter 6.

5.5 Overall structure of the EMS algorithm

An overview of the overall structure of the optimization tool developed in this thesis is shown below. The segmentation of data is explored in Section 7.2 and the trajectory tracker in Chapter 6.

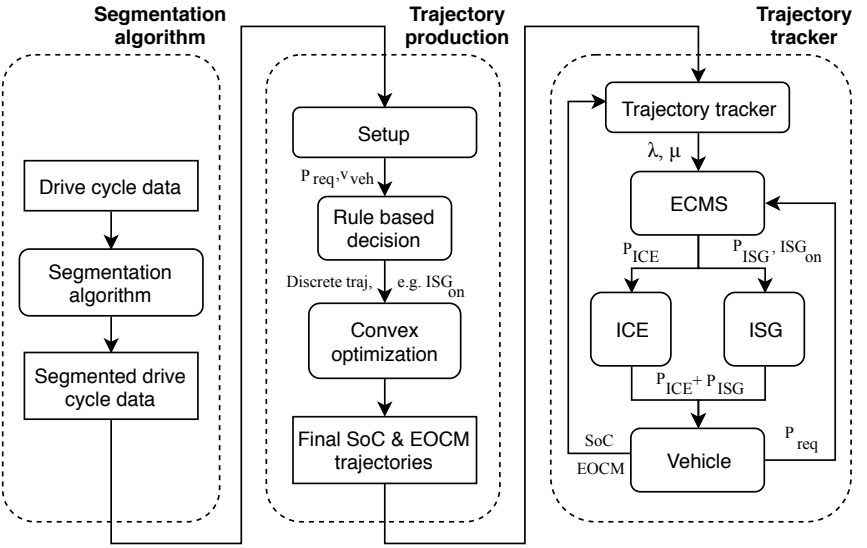

Figure 5.4 Structure of the entire EMS algorithm. The segmentation of data, trajectory production and trajectory tracker all coupled together.

6

Trajectory tracking

6.1 Overview

The trajectories obtained from the convex optimizer are to be used in a control strategy so that the battery in the actual vehicle is utilized in the same manner.

The control problem becomes interesting as there are two trajectories to follow and yet features which might have not been included in the optimization, such as moments of unplanned braking scenarios, should be included without diverging too much from the produced trajectories.

The tracker utilizes an extended ECMS formulation of (4.1) where the battery current utilization is added to the original cost function in order to be able to control the EOCM state as well as the SoC. Due to the state description of the EOCM state, as explained in Section 3.5, the new Hamiltonian is given by (6.1).

$$H = P_f + \lambda P_{ech} + \mu I_{batt}^2 \tag{6.1}$$

Given the two adjoint states (λ and μ) and P_{req}, the optimum instantaneous power-split between the ICE and ISG can be found using (5.2)-(5.13) and power-train component efficiency maps.

As the instantaneous power-split is found using ECMS, and P_{req} is calculated based on driver request and road conditions, the only way in which the power-split and thus the battery use can be controlled is by controlling the values of λ and μ.

As the μ value is added into the ECMS formulation, it also begins affecting the evolution of the SoC trajectory. The two adjoint states are therefore intertwined as λ not only affects the SoC, but also the EOCM state. Likewise, the inclusion of μ in ECMS affects not only the EOCM state, but also the SoC of the battery.

By approaching the problem using PID-controllers, the two variables can be controlled separately as (6.2)-(6.5).

$$\begin{aligned} \lambda = \lambda_0 &+ K_{p,s} e_{SoC} \\ &+ K_{i,s} \int e_{SoC} + K_{d,s} \frac{de_{SoC}}{dt} \end{aligned} \tag{6.2}$$

$$e_{SoC} = SoC_{traj} - SoC_{act} \tag{6.3}$$

$$\mu = \mu_0 + K_{p,e} e_{EOCM} + K_{i,e} \int e_{EOCM} + K_{d,e} \frac{de_{EOCM}}{dt} \tag{6.4}$$

$$e_{EOCM} = \max(EOCM_{act} - EOCM_{traj}, 0) \tag{6.5}$$

The implementation of a controller for the real-time tracking problem in this thesis is made in a more comprehensive and hierarchical way than a traditional PID-controller. During a considered drive cycle, the error between the current state and the reference state changes not only due to the used tracker controller, but also due to the ever-changing reference and the changing power demands in the vehicle. The trajectory controllers try to change the current values of the states they are controlling to the corresponding reference values through ECMS by altering the adjoint states. This control problem is not a traditional control problem, as the controller output is not the actuator output itself, and neither can the state response to the actuation be predicted. In this control problem the power demand is never constant, and efficiency maps of the actuators are not flat.

The reason for controlling the battery use through ECMS in the tracker algorithm is that the main goal of the controller is minimizing fuel and not minimizing the error to the reference trajectory. The trajectory provides a trend for the battery utilization and this results in a tracker controlling the adjoint states that are input to the ECMS. It is the ECMS algorithm that determines the actual use of the ICE and EM, which is the final output, and thus the changes in the battery states.

The reason for only using positive errors in (6.5) is that the battery should not be incentivized to use high currents to reach higher internal temperatures for the sole reason that it is expected to heat up. The EOCM state trajectory is instead used as a roof for the state of EOCM. The state is only controlled once it goes beyond the expected trajectory. The battery will thus be restricted from heating up more than expected, as the use of high currents (and thus an increase in the battery thermal state) is predicted in the near future and the battery should be able to output these. Once the state of EOCM is reduced to below the roof value, the integrator-term in (6.4) uses the negative error until $\mu = \mu_0$ where it is then reset, this is done in order to prevent jerking.

In contrast to the battery SoC, the EOCM state of the battery can not be directly controlled in both directions. There is no possibility to lower the state more than by setting the ISG current usage to its minimum value. In that scenario the EOCM state will only decrease as a function of the state itself. Thus, the increase in EOCM state can be directly controlled, while only a lesser degree of control exists for the reduction of the EOCM state. This has to be kept in mind when the EOCM state

gets too high and is to be reduced. The PID-controller will force the battery currents down, but the cooling effect of this control output will not be noticed directly. It is important then to not become even more aggressive, as the state can not cool slower than predetermined by the model.

The integrator accumulates the error over time and is needed to make certain exclusions. In the SoC trajectory following, the integrator needs to be put on hold whenever a regenerative region is encountered. This is to prevent the controller from becoming more aggressive during unplanned braking scenarios.

The value of μ_0 is never set to zero as it acts as a safeguard against the SoC controller using unnecessarily high currents and thus heating up the battery rapidly. Furthermore, depending on the magnitude of μ, the SoC tracker is differently limited in terms of how aggressive it can be. The values of the adjoint states thus not only have an effect on the trajectories, but the characteristics of the controller as well.

Another controller type is the tangent function controlling of the values of λ and μ. An example for controlling λ is shown in (6.6)-(6.7).

$$\lambda = \lambda_0 + K_a \tan(e_{SoC} + K_b) \tag{6.6}$$

$$e_{SoC} = SoC_{traj} - SoC_{act} \tag{6.7}$$

The adjoint state related to the thermal loads of the battery is of critical importance during recharging phases. If there are no thermal limitations in the battery, the ECMS Hamiltonian (6.1) turns back into (4.1) and the solution to (4.2) becomes $P_{ISG} = P_{ISG,min}$. The value of μ is therefore an important factor which limits the magnitude of negative currents.

Due to unforeseen scenarios, there may be cases when the real-time tracker wants the battery SoC to decrease at a slow pace, but the ECMS algorithm wants to discharge it much more rapidly due to the fuel consumption being lower for that set of adjoint states. This results in the two algorithms working against each other, making this control problem even more intricate.

The discrete ISG variable which dictates whether the ISG should be turned on or in standby mode is determined by taking the minimum of two Hamiltonians in the real-time controller. The two Hamiltonians correspond to the two possible states of the ISG. For the case when the ISG is turned on, the same Hamiltonian is used as before. When the ISG is in standby mode, the considered costs in the Hamiltonian are: fuel from the ICE, costs of the auxiliary loads and the fixed cost of the ISG losses.

6.2 Baseline controller

The existing algorithm that determines the battery use in the vehicle is nonpredictive and rule-based logic is used for the ISG utilization. All other strategies are compared to the Baseline controller.

6.3 SoC trajectory modification

Braking phases usually last for a short period of time. When the input data to the optimization tool is segmented and also due to unplanned traffic interaction, occasional regeneration phases may be missing in the input to the optimization tool. In the real-time tracking of the SoC trajectory, the SoC will then increase as the uncaught regeneration phase occurs. Without any modification to the current controller implementation, during the recharging section the controller will see an increasing error in the SoC trajectory, causing it to output decreasing values of λ and trying to force down the SoC. This occurs even if the integrator in the SoC controller is put on hold due to the proportional error. After the recharging phase ends, the ISG will be utilized aggressively in order to force down the SoC back to the original trajectory. This aggressive controlling of the SoC could cause increase in fuel consumption, as well as the EOCM state increases due to more aggressive battery use.

The proposed method alters the reference SoC trajectory in real-time as the both predicted and unpredicted regeneration phases occur. As the power demand reaches a certain negative threshold, the SoC reference is set to the current SoC value. This results in the SoC controller not attempting to force down the SoC by increasing the λ value when the battery is being recharged as there is no error between the current SoC and reference SoC. When the unpredicted regeneration phase finishes, the future SoC trajectory is altered as to reach the next minima in the original SoC trajectory from the current SoC value more smoothly. This is done by taking the current SoC error and adding it to the reference SoC in a linear fashion, going from $SoC_{ref} + SoC_{error}$ to SoC_{ref} between the end of the unpredicted regeneration phase and the next minima in the SoC trajectory.

6.4 Tracking controllers

Several different versions of a tracking controller are developed and benchmarked. Most of these are different versions of the basic PID-controller. The versions of the tracking controller implementation explored in this thesis are listed in Table 6.1. The tuning of the controller parameters is made so that the trajectories are followed not too aggressively as that causes an alternating fluctuation in the actuation between the ICE and the ISG. A balance between good tracking of the trajectories and low amount of fluctuation in the actuation is found by manually tuning the parameters

for one drive cycle, the found parameters are then used for the remaining drive cycles.

Name	Controller characteristics
Baseline	The current implementation of a tracker as explained in Section 6.2.
BN	As Baseline but the reference SoC is replaced by the output SoC trajectory from the convex optimizer.
PIDC	PID-controller with λ_0 and μ_0 set as constants.
PIDSC	As PIDC with addition of SoC modification as explained in Section 6.3.
PIDT	PID-controller with λ_0 and μ_0 set as the current solution adjoint states obtained from the trajectories of the convex optimizer.
PIDST	As PIDT with addition of SoC modification as explained in Section 6.3.
TAN	Aggressive tangent function controller as (6.6)-(6.7).

Table 6.1 Tracking controllers explored in this thesis.

6.5 Terminal SoC compensation

As it has been mentioned, the battery of the vehicle has to be charge neutral at the end of the drive cycle. As the terminal SoC usually deviates from $SoC_{neutral}$, meaning that the battery ends the drive cycle with different levels of charge, the comparison in the measured fuel consumption will be inaccurate. To resolve this issue, the fuel consumption for the controllers given in the results of this thesis is compensated. The calculation assumes that during the next second after the drive cycle has ended the ISG provides enough power to reach $SoC_{neutral}$ from its current SoC value. The compensation uses the average λ of all simulations for the considered drive cycle, which can be interpreted as the equivalence factor of using electrochemical energy. This provides an estimate of the additional fuel consumed which can be converted to the CO_2 emissions as

$$\zeta = \frac{\lambda_{avg}(SoC_{neutral} - SoC_{end})E_{batt}CO_{2,fac}}{D_{tot}\rho_{fuel}LHV}, \tag{6.8}$$

where λ_{avg} is the average adjoint state from all controller types on the considered drive cycle, SoC_{end} is the considered simulation's terminal value of the SoC, $CO_{2,fac}$ is a factor for conversion between carbon dioxide emissions and fuel consumption, D_{tot} the total distance of the considered drive cycle, ρ_{fuel} the fuel density and LHV the lower heating value of the fuel. The value ζ has the unit $[\frac{CO_2}{km}]$ as that is the unit given by the advanced simulation software. The emission of carbon dioxide is assumed to be directly related to fuel consumption and is therefore treated as such.

7

Drive cycle segmentation

7.1 Drive cycles

As this thesis investigates fuel consumption benefits via SoC shaping using look-ahead control, drive cycles containing topography variations are of interest. One drive cycle, here named MTN, is given and is used in this thesis. The MTN drive cycle is a mountain route and is characterized by having a long uphill section followed by a long downhill section. The MTN drive cycle has real altitude data. In addition to the MTN drive cycle, for validation purposes and also due to a need of several drive cycles on the same road, but with different driving behaviours, a new drive cycle is created for this thesis. A route in Gothenburg is chosen such as to include city traffic, rural driving and hilly regions. This route is driven several times in a VCG vehicle and the necessary signals are measured and logged. The drive cycles used in this thesis are summarized in Table 7.1.

Name	Characteristics
MTN	Mountain route, long uphill section followed by long downhill section.
GBG1	Gothenburg route including city traffic, rural areas, hilly regions. Driven normally.
GBG2	As GBG1, driven slowly.
GBG3	As GBG1, driven normally.
GBG4	As GBG1, driven aggressively and fast.
GBG5	As GBG1, driver attempting ecodriving.

Table 7.1 Driving cycles used in this thesis.

The available information for the MTN drive cycle contains the vehicle velocity and inclination angle, thus the power demand trajectory can be calculated by approximating the acceleration as changes in velocity. For the GBG1-GBG5 drive cycles, the vehicle velocity and inclination angle is sampled from the real vehicle. The signal for the inclination angle has a minor offset and a suggestion for compensating for this discrepancy is presented in Appendix A.

7.2 Segmentation

One issue in the real-life implementation of the optimization tool is the amount of data that has to be processed and calculated upon. The goal is to have a system acting on many produced vehicles meaning that the online optimization and data processing will be done for many vehicles simultaneously. A requirement for this to be realizable is that the optimization runs on small amounts of data. Since the amount of information about the drive cycle which is input to the optimization tool determines the optimization speed and the output information magnitude, the segmentation of the input data (drive cycle information) is explored in this thesis. The segmentation of the drive cycle should be made in such a manner that the amount of samples throughout a certain drive cycle is reduced while still maintaining the same results, or at least closely similar results, as obtained from the original dataset.

The segmentation for the optimization tool is to be defined. All of the data points in-between the segmentation data points are, however, allowed to be used to define the values of the segments. In this thesis two different ways of using the measured values are explored. Once a new data point is to be calculated according to a chosen segmentation method, the values used for that segment can be calculated as the following:

- use instantaneous values from the segmentation instant and zero-order hold (ZOH) the value until the next segmentation instant,
- use the average of all sampled values since last segment instant for the current segment, ZOH this value until next segmentation instant.

As can be seen in Table 7.2, with fine segmentation the use of instantaneous values yields SoC trajectories that are more representative of the optimal one. For coarser segmentation, the average values begin to yield better results, especially during the aggressive drive cycle GBG4. In that scenario, certain high magnitude values may be caught by the instant sampler which it then assumes to be constant for an entire segment.

Drive cycle	Instant @ 5s	Average @ 5s	Instant @ 10s	Average @ 10s
GBG1	0.0139	0.0173	0.0232	0.0278
GBG2	0.0194	0.0291	0.0411	0.0313
GBG3	0.0169	0.0169	0.0325	0.0289
GBG4	0.0155	0.0176	0.0586	0.0364
GBG5	0.0233	0.0156	0.0490	0.0224
Average	**0.0178**	**0.0193**	**0.0409**	**0.0294**

Table 7.2 RMS of error between optimal SoC trajectory calculated from using all data samples and the SoC trajectories produced by sparsing the data.

The deviations from the SoC and EOCM state trajectories for the drive cycle GBG4 using instant and average values are shown in Figure 7.1.

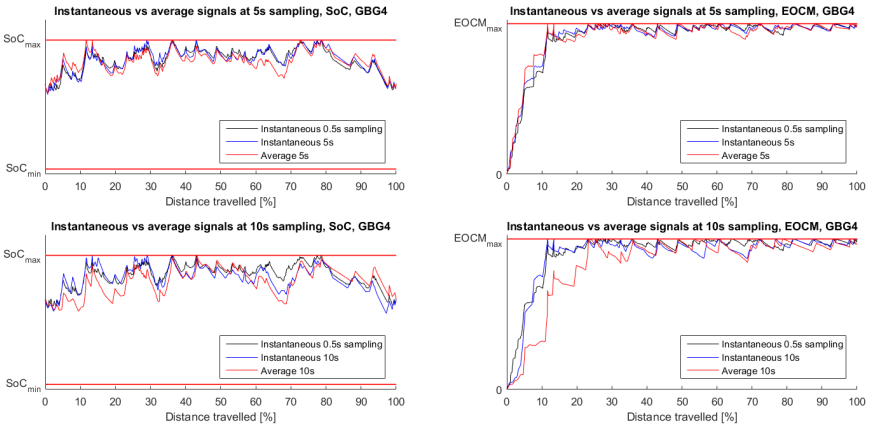

Figure 7.1 Influence of taking the average of signal values during the segments vs taking instantaneous values at the sampling instances. The averaged trajectories deviate from the actual trajectory more than the instantaneous trajectories as the amount of segments decreases.

In addition to the choice of calculating the used signal values, the segmentation algorithm is to be designed. In this thesis 11 different methods of segmenting the data are presented. The logic for each segmentation method is shown in Table 7.3 where they are also motivated.

Segment. mthd. nr.	Logic the segmentation algorithm is based on and motivation
1	**Speed limit changes and altitude differences**, by using this information the sampling positions could be predetermined beforehand with traffic information and GPS alone. Sampling instances could therefore be triggered externally without the need for sampling the vehicle variables continuously. The changes in speed limit and altitude should also correspond to changes in kinetic and potential energy of the vehicle.
2	**Speed limit changes, altitude differences and maximum sample distance**, as method 1, but additionally a maximum sample distance is added so that excessively long segments do not occur. Very long segments could risk the integrity of the calculated trajectories.
3	**Fixed sample time interval**, by having a fixed sample time interval the implementation of triggering the samples would be relatively easy. The sampler would be triggered every fixed amount of time. This removes the need of external data checking and sampling.
4	**Fixed sample distance interval**, the fixed distance sampling is attempted as well for benchmarking purposes.
5	**Speed changes and altitude differences**, segmenting where speed and altitude changes occur should result in finer segmentation during transients and more important events in the drive cycle, while the uneventful periods are neglected.
6	**Speed limit changes, altitude differences and power sign change**, as method 1, but the power sign changes are captured as well. This would cause more segmentation during transitioning between braking and accelerating, and thus these data points could be of value.
7	**Speed limit changes, altitude differences and power sign change delayed by 0.5s**, as method 6, but the sampling due to power sign changes is delayed by 0.5s as to catch the actual deceleration/acceleration value and not when the power demand is near zero.
8	**As segmentation method 7, but the delay is increased further**, as to allow the deceleration/acceleration value to stabilize before sampling.
9	**Power requirement differences**, the idea of only using the calculated power request requires that the power is estimated before the segmentation. Since the power demand is what is essentially of interest for the optimization tool, it could be a potential candidate for segmenting.
10	**Power requirement differences and peaks in power**, as segmentation method 9, with addition that the peaks in power are sampled. This requires additional calculation beforehand, as a low-pass filter is needed for finding the peaks. It can however still be done without hindsight, and should further represent the drive cycle more correctly.
11	**Speed changes, altitude differences, peaks in speed and altitude**, as segmentation method 5, but in addition the peaks in speed and altitude themselves are caught as well to add weight on segmenting during eventful periods in the drive cycle.

Table 7.3 The 11 different methods of segmenting the data explored in this thesis.

8

Drive cycle merging

So far is has been assumed that there is one single run on a drive cycle, a single power demand curve, which created a single SoC and EOCM state trajectory to be followed. In this chapter the possible method of merging multiple drives on the same path in order to create a merged SoC and EOCM state trajectory is explored.

In reality, there will be different types of drivers that drive on the same route and also the driving characteristics for the same driver may differ depending on things like weather and traffic conditions. These variations are reflected in the drive cycles GBG1-GBG5, as explained in Section 7.1. The idea is to use data from different driver characteristics and try to merge the information in such a manner that provides SoC and EOCM state trajectories that best match a new and unknown driver.

The first approach to merge the data from different vehicles is explained as following. The data sampling for each vehicle occurs according to a chosen segmentation algorithm from Section 7.2. From this data set, SoC and EOCM state trajectories are calculated for each vehicle after they have driven along a drive cycle. These resulting trajectories are then merged together for the next vehicle to use by means of averaging the trajectories.

In the real implementation this method of merging data might not be the best approach. Since a considered vehicle may have different SoC and EOCM states upon entering the drive cycle, the calculated trajectories assuming different initial states are not appropriate. An additional issue with this method is that the considered vehicle parameters, such as the mass of the vehicle and efficiency maps, are not necessarily the same for all vehicles and the calculated trajectories thus can not adapt to these differences.

A different approach is to sample the information about the road itself and also have the sampling for each car during a route occur at the same positions. If this is done, the velocity and inclination angle profile throughout the drive cycle can be merged. From these merged values, the calculation of the power demand can be done for each vehicle with their own specific parameters. Furthermore, the information about the drive cycle can be easily merged together and input to the convex optimization algorithm as the data is merged to one set of data point positions. As the input to the optimization tool itself is merged, and not the output from it, con-

sideration to each vehicle's unique initial state is allowed. By applying the previous approach of merging trajectories, the SoC trajectories would be fixed to a single starting and terminal value. In this approach, however, the SoC and EOCM state trajectories can be calculated for any initial status of the vehicle as the current state and the predicted power demands in the future are known.

The problem with merging the velocity and altitude profile is that it requires synchronized sampling instances in order to serve as input to the optimizer. If two vehicles sample according to a certain segmentation algorithm and store the information at non-synchronized sampling instances, there will be twice as many data points. Different approaches can be made to try to achieve synchronized data sets, some are to use either distance-triggered samples or coordinate-triggered samples. In this thesis, the vehicles sample according to a chosen segmentation algorithm from Section 7.2. The data points are thus not synchronized between the vehicles. To synchronize these, the segmentation positions (in terms of distance from the starting of the drive cycle) from one vehicle are used. For all the other vehicles the trajectories of velocity and inclination angle are interpolated to this set of data point positions.

9

EOCM state estimation learning

The EOCM state of the battery is a function with two terms, one dissipative term which dictates how the battery cools down, and one heating term which dictates how the battery heats up. The cooling is proportional to the current state value and the heating is proportional to the battery current squared, see (3.24). Issues may arise from this when using the optimization tool with coarser segmenting of the drive cycle. With a decreasing amount of segments, as the length of each segment is increased, the convex optimizer will be all the less inclined to allow high currents in the battery as that would heat up the battery significantly during a single long segment. If even higher currents are suggested, the upper threshold for the EOCM state will not be satisfied. In the real-time tracker utilizing ECMS, and also when the segmentation is not as coarse, higher currents may be allowed and preferred during shorter periods, such as during short downhills or braking, resulting in a battery use that is much more fluctuating. This discrepancy between the estimated use of the battery and the real-time use of the battery may result in the real-time tracker being restricted from using higher currents even though it may be more optimal.

9.1 Alpha correction

The approach taken to correcting the EOCM state trajectory is to apply feedback learning from previous runs on the same drive cycle and by using information about how the real EOCM state evolved. As the actual EOCM state values can be measured and logged, the regions in the drive cycle where the increases in the EOCM state are underestimated can be modified by adding an α-term into the original equation (3.24) for the heating term, resulting in (9.1).

$$EOCM_{k+1} = f_1(EOCM_k) + \alpha_k f_2(I^2_{batt,k}) \tag{9.1}$$

The calculation of a new α is made as

$$\alpha_{new,k} = \frac{EOCM_{meas,k+1} - f_1(EOCM_{meas,k})}{f_2(I^2_{batt,meas,mean,k})}, \tag{9.2}$$

where k refers to the individual segments, $EOCM_{meas}$ the measured EOCM state of the simulated vehicle at the segmentation start instances, and $I_{batt,meas,mean}$ the mean current of the simulated vehicle during the considered segment.

As the EOCM state trajectory in the simulated vehicle may, due to segmentation or other causes, increase with a slower rate than predicted, or even decrease in value during a segment where the state is expected to increase, the calculation of new α values may yield unrealistic scenarios. If the α value approaches low values, the effect of currents on the EOCM state will start to be neglected. To make sure that it takes feasible values, and also in order to allow for many iterations of the learning feedback, the new trajectory of α is given as (9.3) where n denotes the iteration number of the feedback learning, $\alpha_{new,k}$ is the new calculated α value for segment k as in (9.2), $\alpha_{upperthresh}$ is an upper threshold for the new calculated α value, and $0 \leq \beta \leq 1$ acts as a weighting factor for how much each learning iteration affects the used α values.

$$\alpha_{k,n} = \beta\alpha_{k,n-1} + (1-\beta)\max(\alpha_{k,n-1}, \min(\alpha_{new,k}, \alpha_{upperthresh})) \tag{9.3}$$

The reason for including an upper threshold for α is that with high values of the heating term, the convex optimizer might not be able to find any feasible solution to the problem as even the sole use of battery current for the auxiliary system might overheat the battery.

The dissipative term, f_1, in (3.24) is kept unmodified.

10

Receding horizon

So far it has been assumed that entire drive cycles are known. In the real implementation, even though information about all existing roads may be available, there are two limiting factors still preventing the use of all available future information about the route. One is the fact that most roads are not very long before a new intersection/choice of path occurs. A new intersection induces uncertainty in planning the entire drive cycle as it is never sure which road the driver will pick at the intersection. The other factor limiting the use of entire drive cycle information is the high bandwidth and computer processing power the simultaneous planning of entire drive cycles for many vehicles would demand. Along with the above mentioned issue in the absence of predicted route data, an alternative would be to optimize until the next intersection, as that would remove the uncertainty of the decision on the intersection. As mentioned in Section 7.2, segmentation of the drive cycle information is done in order to reduce the amount of data transmitted and calculated. By introducing a short horizon of future information, both these problems can be satisfied furthermore. The amount of calculation needed is reduced as the optimization is not made on entire drive cycles, and also the amount of data bandwidth is reduced as the length of the SoC and EOCM state trajectories sent to the vehicle is reduced.

10.1 Horizon calculation

The SoC trajectories from a simple implementation of a receding horizon and a simple PID-controller acting as the tracker is shown in Figure 10.1. The terminal SoC for each horizon is set to $SoC_{neutral}$ in order to force charge sustenance in the vehicle. Due to the charge sustenance implementation, the calculated trajectories never venture far away from the neutral charge state ($SoC_{neutral}$) and this thus limits the battery use. The entire SoC range is not utilized.

As can be seen in the Figure 10.1, the trend of the power for each receding horizon approximately follows the same pattern as the optimal SoC trajectory calculated when the entire drive cycle is known, albeit shifted with the horizon length. The power trend is calculated as the mean of the expected power demand throughout

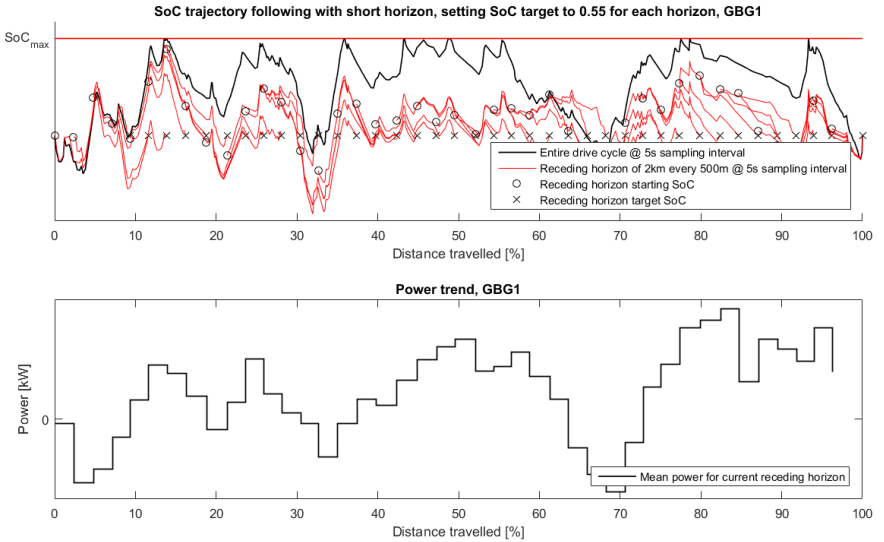

Figure 10.1 Top: SoC trajectories from the first implementation of a receding horizon. The initial SoC for each horizon is the instantaneous value of the current, real-time controlled, SoC value. The terminal SoC for each horizon is set to $SoC_{neutral}$ in order to remain charge neutral. Bottom: Mean power for each receding horizon.

the considered horizon. This information is used to attempt to improve the receding horizon implementation. The idea is that if the overall future trend of the power shows that it is negative, implying recharging phases, the SoC target is set to a higher value as the real-time controller should not start penalizing larger magnitudes of recharging. Conversely, if the power demands are higher than usual, implying acceleration and uphill phases, the SoC target is set to a lower value so the battery is allowed to be discharged more. The thought is that deceleration and downhill phases follow acceleration and uphill phases, and vice versa.

A total of five receding horizon versions, named A-E, are tested against each other. The receding horizon methods are explained in Table 10.1. All versions have a trigger that once the overall travelled distance at the end of the current considered horizon is the drive cycle overall length, the target SoC for the optimization calculation is set to $SoC_{neutral}$ in order to remain charge neutral over the entire drive cycle.

In horizon method B the terminal SoC is set to the current SoC, resulting in no punishment for being at either high or low SoC values. The real-time controller has no set demands for forcing any larger trends in the trajectories. The method C of the horizon calls for adding a soft constraint in the convex optimization problem of the

Horizon method	Explanation
A	Horizon target SoC set to $SoC_{neutral}$ to keep SoC neutrality at all times.
B	Horizon target SoC set to actual SoC of vehicle at instant of calculation.
C	Horizon target SoC set as varying from SoC_{max} to $SoC_{neutral}$ as the fraction of the overall distance travelled increases. The SoC value is set as a soft constraint in the convex optimizer with a set cost for deviating from that terminal value.
D	Horizon target SoC varies as function of power trend during horizon and actual SoC of vehicle at instant of calculation.
E	Horizon target SoC varies as function of power trend during horizon and charge neutral value.

Table 10.1 The five different horizon methods explored in this thesis and how their terminal SoC is set.

terminal SoC value. The idea is that the terminal SoC of each horizon is chosen by the optimization tool itself for optimality and best fuel consumption. As the vehicle approaches the end of the drive cycle this freedom is reduced in order to achieve charge sustenance.

In the method D implementation of a receding horizon the terminal SoC for each receding horizon optimization is a function of the current SoC and the power trend, as in (10.1). The parameters k_{pos} and k_{neg} are manually tuned for GBG1. The terminal SoC for the horizons near the end of the drive cycle (those which have the terminal distance as the entire drive cycle distance) is set to $SoC_{neutral}$ in order to obtain charge sustenance over the entire drive cycle and thus yield comparative results in fuel economy. The value of k_{pos} is negative and the value of k_{neg} is positive.

$$SoCtgt,horizon = \begin{cases} SoC + k_{pos}P_{mean,horizon}, \text{if } P_{mean,horizon} \geq 0. \\ SoC + k_{neg}P_{mean,horizon}, \text{if } P_{mean,horizon} \leq 0. \end{cases} \tag{10.1}$$

The calculated terminal SoC for each horizon is also saturated such as to not optimize for infeasible terminal states (setting the terminal SoC very near SoC_{min} or SoC_{max}).

The method E implementation of the receding horizon is as the version D implementation, except that the terminal SoC is now a function of both the power trend of the current horizon and $SoC_{neutral}$ as in (10.2).

$$SoCtgt,horizon = \begin{cases} SoC_{neutral} + k_{pos}P_{mean,horizon}, \text{if } P_{mean,horizon} \geq 0. \\ SoC_{neutral} + k_{neg}P_{mean,horizon}, \text{if } P_{mean,horizon} \leq 0. \end{cases} \tag{10.2}$$

The target value of the EOCM state for all of the horizon calculations is set to $EOCM_{max} - \delta$ and is set as a hard constraint. This is done due to the fact that if

the terminal state of EOCM in the considered horizon happens to have a value very near or equal $EOCM_{max}$, and as the real-time tracker may overshoot the reference trajectory, the real-time value of the state at the end of the considered horizon may be greater than $EOCM_{max}$. When a new calculation begins with this value as the initial condition of the state, it will not be able to find any solution as the initial condition itself is not within the allowed range.

The calculation of a new horizon is done before the end of the previous horizon is reached. This allows for using information from both the old and the new calculated trajectory for the current position. The EOCM state trajectory used is always the previous horizon's trajectory. It is lagged due to tracking an EOCM state trajectory that is optimized from the current value would cause low controlling of it. Without any controlling of the EOCM state, the tracker would unrestrictedly reach and go beyond the hard EOCM state limit.

10.2 New/old horizon trajectory use ratio and delays in calculations

The algorithm is set up so that the calculation of a new horizon occurs before the current horizon ends. This allows the use of the previous horizon data in the calculations of the next horizon. This is used as

$$SoC_{horizon}(k) = \begin{cases} SoC_{horizon}(k-1)\vartheta + SoC_{horizon}(k)(1-\vartheta), \text{ for } SoC_{horizon}(k-1) \in horizon(k), \\ SoC_{horizon}(k), \text{ for } SoC_{horizon}(k-1) \notin horizon(k), \end{cases} \tag{10.3}$$

where $0 \leq \vartheta \leq 1$.

As mentioned in Section 10.1, the corresponding ϑ for the EOCM state trajectories is always 1. Figure 10.2 shows how different weights for SoC correspond to the trajectory being followed.

In the real implementation of updating the trajectories to be followed as the vehicle drives there will be a delay between the instant when the vehicle requests a new receding horizon to be calculated, and the instant once the calculation is done, sent back and starts being used by the vehicle. The effect on fuel efficiency due to delays in calculating the new receding horizon trajectories is explored. In Figure 10.3 the effect of delays is visualized.

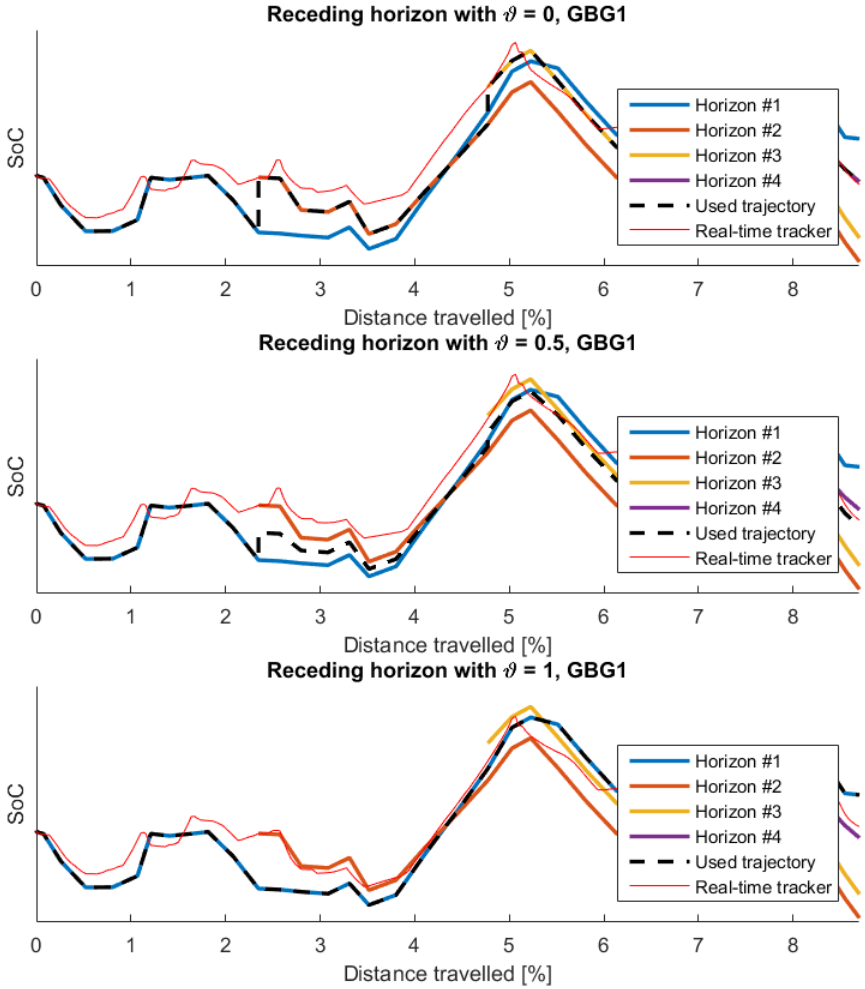

Figure 10.2 Value of ϑ influence on the followed SoC trajectory. Dashed black line corresponds to resulting reference trajectory.

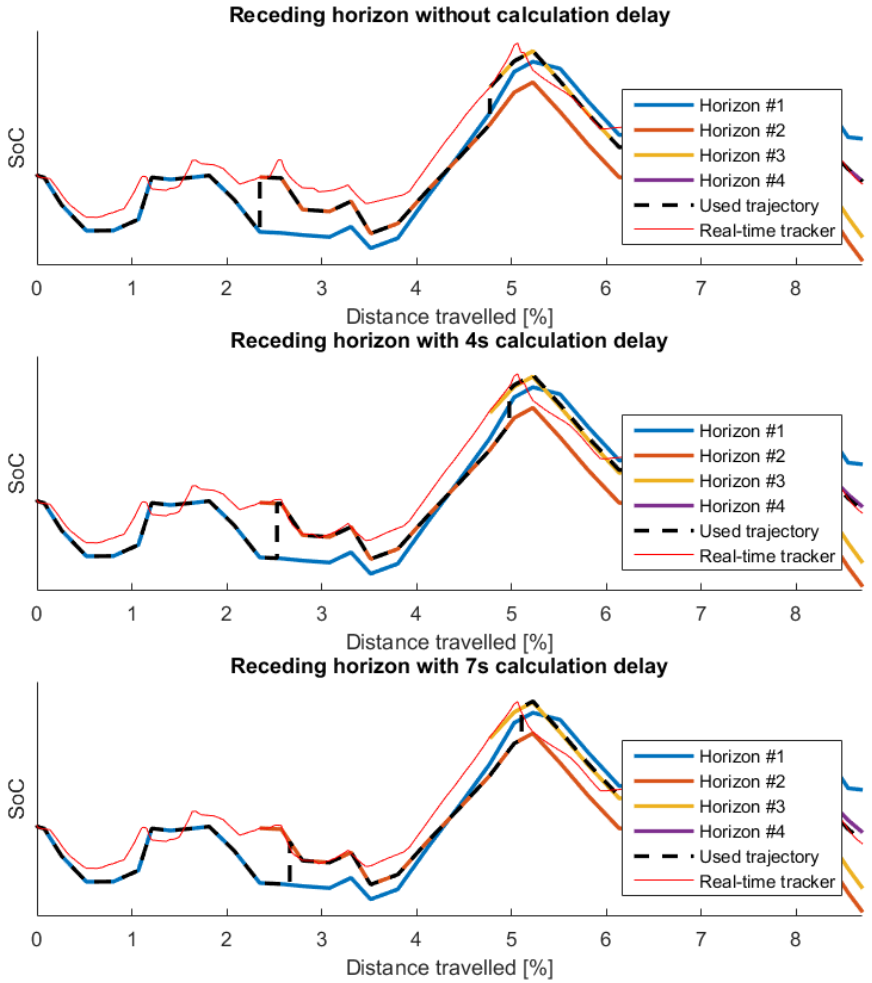

Figure 10.3 Delay influence in calculations of receding horizon trajectories on the followed SoC trajectory. Dashed black line corresponds to resulting reference trajectory available and used by the vehicle.

11

Results

11.1 Optimization tool design

In Chapter 5 the optimization tool is developed and simplified. The results in the form of SoC and EOCM reference trajectories given from the first layout which entailed disregarding the EOCM state can be seen in Figure 11.1 where it is compared with the results from the same algorithm but with the EOCM state added to the convex optimization.

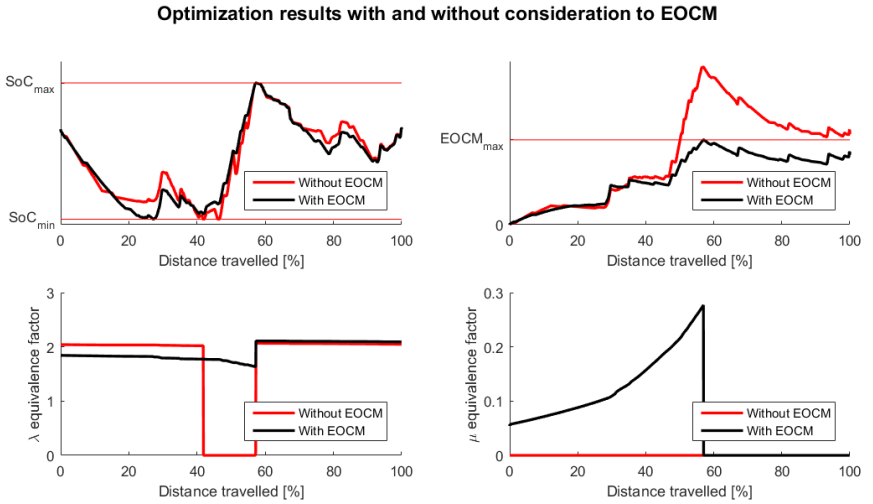

Figure 11.1 Two optimization tool results, one in which the EOCM state is neglected and the other where it is not.

It can be seen that when the EOCM state is neglected in the optimization, the state evolves and reaches values well beyond the upper threshold. The thermal limi-

tation is therefore an issue that may not be neglected in the optimization. The effect can clearly be seen in the SoC trajectory as well. The calculated optimal SoC trajectory becomes smoother when the EOCM limit is considered, and the effect can be mainly seen during the regenerative phases. The discharge phases are also following the same trend in certain segments. This reflects that at least for those segments the EOCM state is not of concern.

The effect of simplifying the optimization tools by replacing the DDP-based ISG discrete decision with a rule-based logic is shown in Figure 11.2.

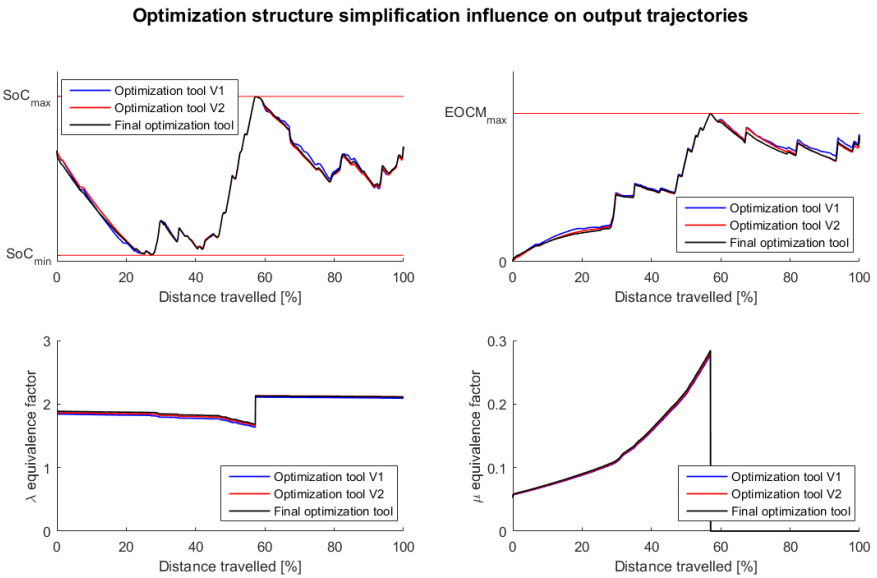

Figure 11.2 Optimization tool results for the increasing simplifications made in the structure of it. V1 refers to optimization tool in Figure 5.1, V2 refers to optimization tool in Figure 5.2. As the ISG is not turned off as much in the simpler tools, the losses in electrical energy cause decrease in the remaining electrical energy left for propulsion. The values of the adjoint states are therefore higher. The SoC and EOCM trajectories, however, are very similar.

The results show how the different versions of the optimization tool compare against each other, and as it can be seen, there is minimal loss of optimality as the iterative loop between the convex optimizer and the DDP is removed. The overall removal of DDP, and the decision to determine the discrete variable of the ISG for the convex optimization problem formulation with simple logic, yields minimal loss in optimality as well. A reason why the influence of the different losses between the

two ISG modes on the produced trajectories is small is the fact that, relative to the power demands of the drive cycle, the difference between the ISG losses is minimal. It can be noted that there is a slight increase in the adjoint state values as the tool is simplified, making the use of electric energy marginally more expensive. This is due to the more complex tools allowing the ISG to be switched off more often. By having the ISG switched off, less electrical energy is used for keeping the ISG idle, allowing more of the electrical energy to be used for actual propulsion instead.

The small effect of the discrete ISG variable for the calculation of the trajectories does not mean that the discrete ISG variable can be neglected completely. There is potential to save fuel by switching the ISG into standby mode by controlling it in the low-level real-time controller as the switching of the ISG between the two modes is nearly instantaneous and comes with no cost.

It is noted that by adding the threshold of the EOCM state into the convex optimizer, the values of μ start coming into play which in turn cause λ to no longer be piecewise constant as seen in Figure 11.1. The reason for this is explained in Chapter 6 where it is noted that the two adjoint states are intertwined as these affect each others trajectories.

11.2 Segmentation algorithms

The results from the benchmarking of the segmentation methods for the drive cycle GBG2 are shown in Figure 11.3 where the error as a function of the segmentation magnitude is shown for each type of segmentation method listed in Section 7.2. Both the method of taking the instantaneous values during segment instances, and taking the average of all signals during the last sampling instance are evaluated. For each segmentation method the thresholds for triggering new data points are varied such that the amount of segments are reduced. The effect of the level of segmentation is calculated as the RMS of the error between the calculated SoC trajectory from the segmented data and the calculated SoC trajectory from perfect segmentation (reference, high sampling rate).

Due to increased sparsity of the information known about the drive cycle, the calculated trajectories start deviating from the optimal trajectory and thus produce larger errors. The mark of a good segmentation method in Figure 11.3 is it having a small slope with low magnitude which implies that it does not lead to high deviations from the perfect trajectory as the amount of segments is reduced. Not all segmentation methods are capable of producing low amounts of segments during a specific drive cycle. The segmentation methods with this issue are those utilizing sign changes or peaks found in the used input signals. The segmentation method 6, for example, is ultimately limited by the amount of sign changes the power demand has, this results in a minimum threshold of the amount of segments that can be created.

The errors in the SoC trajectories for all segmentation methods shown in Fig-

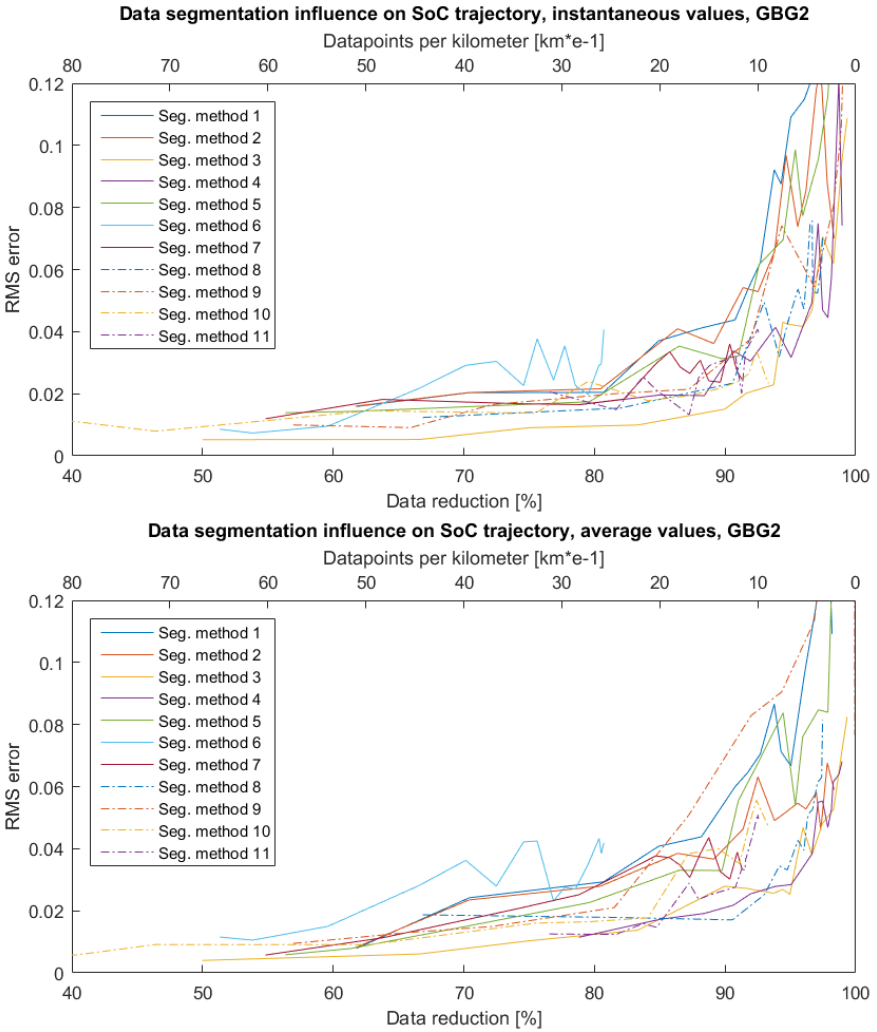

Figure 11.3 Error in calculated SoC trajectory due to segmentation in different segmentation algorithms. Drive cycle GBG2, using instantaneous and average values. The segmentation method 3 has, on average, the least RMS error.

ure 11.3 is only for the GBG2 drive cycle, the results for all drive cycles considered in this thesis are summarized in Table 11.1. The comparison is done at the sparsity of approximately 5 segments per kilometer. In these results it is seen that the segmentation method 3 performs the best. This method of segmentation gives the least

Sg. nr.	GBG1 i.	GBG1 a.	GBG2 i.	GBG2 a.	GBG3 i.	GBG3 a.	GBG4 i.	GBG4 a.	GBG5 i.	GBG5 a.	MTN i.	MTN a.	Avg
1	0.0535	0.0868	0.1081	0.0669	0.0770	0.0677	0.0640	0.0752	0.1058	0.0500	-	-	**0.0755**
2	0.0928	0.0613	0.0883	0.0529	0.0595	0.0780	0.0792	0.0909	0.0673	0.0502	-	-	**0.0720**
3	0.0286	0.0301	0.0423	0.0260	0.0355	0.0296	0.0558	0.0378	0.0414	0.0289	0.0400	0.0539	**0.0375**
4	0.0568	0.0488	0.0321	0.0284	0.0464	0.0600	0.0598	0.0748	0.0389	0.0457	0.0588	0.0682	**0.0516**
5	0.0579	0.0647	0.0870	0.0660	0.0991	0.1057	0.0800	0.1073	0.0672	0.0588	0.0570	0.0654	**0.0763**
6	-	-	-	-	-	-	-	-	-	-	-	-	**-**
7	-	-	-	-	-	-	-	-	-	-	-	-	**-**
8	0.0304	0.0451	0.0457	0.0359	0.0455	0.0527	0.0552	0.0653	0.0446	0.0466	-	-	**0.0467**
9	0.0543	0.1003	0.0688	0.0966	0.0738	0.1325	0.1347	0.1508	0.0489	0.1277	0.0238	0.0502	**0.0885**
10	-	-	-	-	-	-	-	-	-	-	-	-	**-**
11	-	-	-	-	-	-	-	-	-	-	-	-	**-**

Table 11.1 RMS error between non-segmented SoC trajectory and segmented SoC trajectory for the different segmentation methods at approximately 5 segments per kilometer. All considered drive cycles. The 'i' implies instantaneous and 'a' implies average calculation of the segment values.

error from the perfect SoC trajectory for both the instantaneous and the average calculations. One reason behind why the fixed sample time interval works best is that with most other segmentation methods, the extremes of the powers are the ones that are sampled, and therefore the SoC trajectory would begin to be fluctuating more as the non-extreme power demands are more neglected. The smooth power demands are as important as the highly fluctuating power demands for creating a well representative power demand curve of a drive cycle.

Another reason for why the fixed sample time segmentation outperforms the other methods in this study is due to even spread of data points. The route of the GBG1-GBG5 drive cycles contains many different driving scenarios such as city driving, slow regions, hilly regions and some highway driving. As the considered route is filled with eventful sections and thus contain intricate power demand trends throughout, there might be a bias towards segmentation methods providing more uniform positioning of data points. If the GBG1-GBG5 drive cycles were redone outside of the main city of Gothenburg, and instead out on country roads with long high-speed roads few intersections, and also in regions not as hilly as in the considered drive cycles, the results comparing the different segmentation methods could differ and another segmentation method could possibly outperform the fixed sample time interval segmentation method.

The RMS errors tell how the segmentation methods perform relative each other. These do not, however, provide much insight as to how much the trajectories deviate. In Figure 11.4 the trajectories for different values of the RMS error in SoC trajectories are plotted. It can be seen that the resulting trajectory yielding an RMS error of approximately 0.017 (point A in Figure 11.4) follows the perfect trajectory well. As the RMS increases through 0.06 (point B) to 0.09 (point C) the trajectory begins to overestimate the charging and recharging possibilities in the drive cycle.

By using a primitive tracker implemented in MATLAB, the fuel consumption obtained from following the chosen trajectories in Figure 11.4 relative to the fuel consumption obtained by following the perfect trajectory is shown in Table 11.2. The increase in fuel consumption shows a linear trend where an increase in RMS of

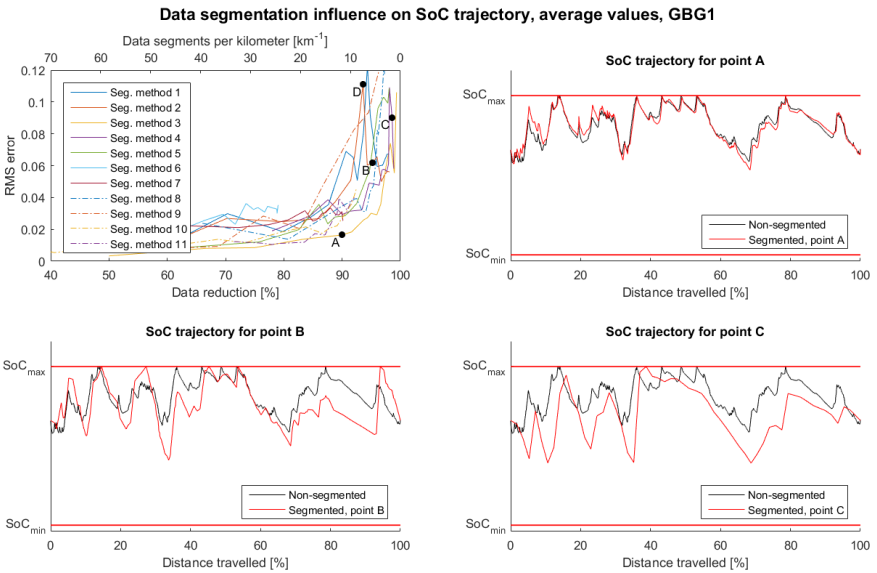

Figure 11.4 Corresponding errors in SoC trajectories for different errors in RMS. The trajectories for points A, B and C in top left plot are shown in the remaining plots.

the SoC trajectory increases the relative fuel consumption.

Trajectory	RMS in SoC	Relative fuel consumption
Non-segmented	0.0000	100.00%
Point A	0.0165	101.16%
Point B	0.0617	106.95%
Point C	0.0900	113.03%

Table 11.2 Corresponding fuel consumption penalty for different values of SoC trajectory deviations. Fuel consumption relative to consumption result for non-segmented input data. Segmentation points A, B and C as shown in Figure 11.4.

The segmentation methods using changes in speed limit, velocity and altitude (segmentation methods 1,2,5,6,7,8 and 11) seemed promising initially due to these factors being the most influential on the changes in the kinetic and potential energies of the vehicle. In the benchmark made in this thesis, these methods perform worse than the segmentation from a fixed sample time interval. Except for the aforementioned possible bias due to the characteristics of the drive cycle, one additional motivation for these methods poor performance is noted. Segmentation method 1, for ex-

ample, uses differences in the vehicle velocity and altitude in order to determine the positioning for data points. By applying this logic, the differences in velocities and altitude might be triggered in quick succession due to acceleration/deceleration and large uphills/downhills. This concentration of data points is shown in Figure 11.5 where the segmentation is done according to method 1. As the amount of data points across the entire drive cycle is to be the same when compared to other segmentation methods, concentrating many data points during certain smaller sections reduce the amount of data points that are used for the remainder of the drive cycle.

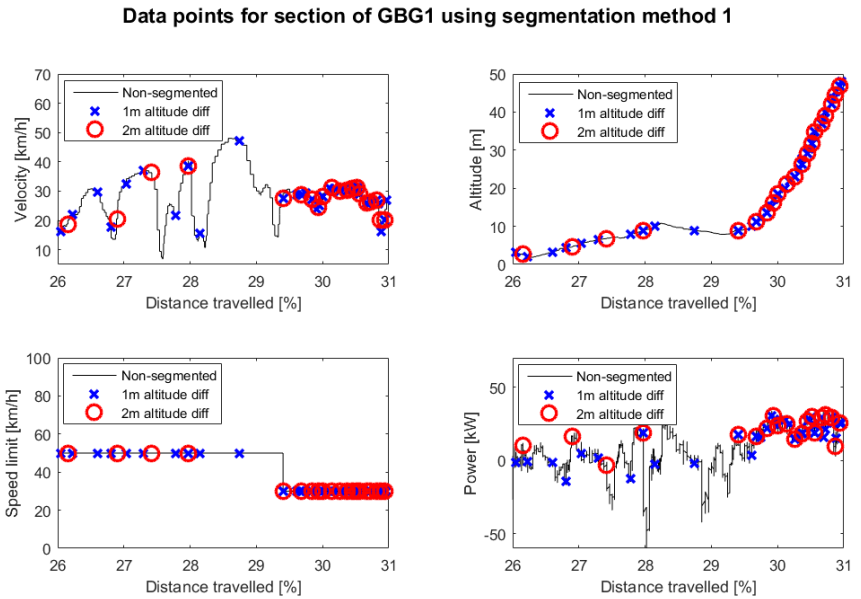

Figure 11.5 Data point positions using segmentation logic for speed limit differences and altitude differences. Concentration of data points occurs during large slopes.

A better approach could be to have three data points during acceleration/deceleration and uphill/downhill events, one before, one during and one after the event. This is due to the power demand changing as much on a constant uphill as on a horizontal plane, and the same for constant acceleration/deceleration phases. This, however, requires hindsight as the end of the current event needs to be known beforehand.

Another suggested segmentation method explored in this thesis is to use changes in the sign of the acceleration and the inclination angle of the vehicle, complemented with a maximum allowed segment distance. The idea behind this segmenta-

tion method is that the transients of hills and the transients of speed can be caught. The resulting data point positioning in the drive cycle can be seen in Figure 11.6 where it is compared to the data point positioning from the fixed sample time interval segmentation. The amount of data points from each algorithm throughout the entire drive cycle is approximately the same but since the more complex segmentation method tends to produce multiple data points in quick succession, the overall coverage of information throughout the entire drive cycle is worse than for the simple fixed sample time interval method.

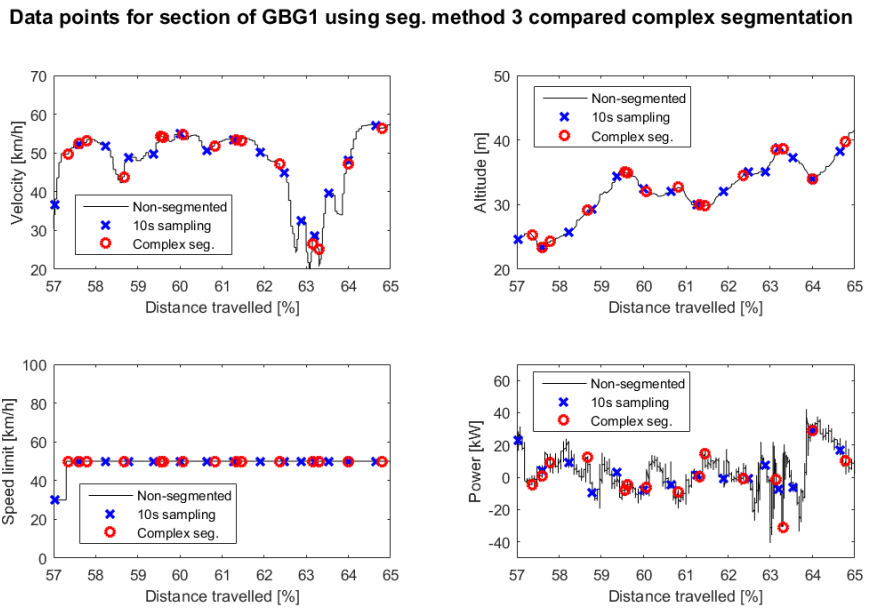

Figure 11.6 Data point occurrences for two different segmentation algorithms. The simple segmentation algorithm of fixed sample times compared to a more advanced segmentation algorithm considering peaks in velocity and altitude, and also with a cap on the maximum segment length. The amount of data points throughout the entire drive cycle is almost equal.

The point of using a smart segmentation method is to concentrate more data points where more things happen. The results show that, for the considered drive cycles, the benefit of concentrating data points in specific regions is outweighed by the disadvantage of less available data across the rest of the drive cycle.

In addition to the fixed sample time interval segmentation method providing the least errors in the produced SoC trajectories when compared to the optimum SoC trajectories, it is noted that by using this segmentation method, the implementation

of triggering the segments is simply realizable. There is no requirement to calculate changes or trends in any measured signals, the only trigger for creating a new data point is time. Furthermore, since the EOCM state evolution is dependent on time (as it contains a dissipation term which is dependent on time), there is no need for a recalculation of the EOCM state evolution parameters when the time steps are constant.

Peaks in calculation error

The cause for the high peaks occurring in the RMS of the segmented trajectories seen in Figure 11.4 (point D) is the incorrect assumption that certain power demands are held over entire segments. This effect can be seen in Figure 11.7 where at approximately 93% of the total distance travelled the large negative power demand is held for a segment that is longer than in reality, thus causing a spike in the SoC trajectory.

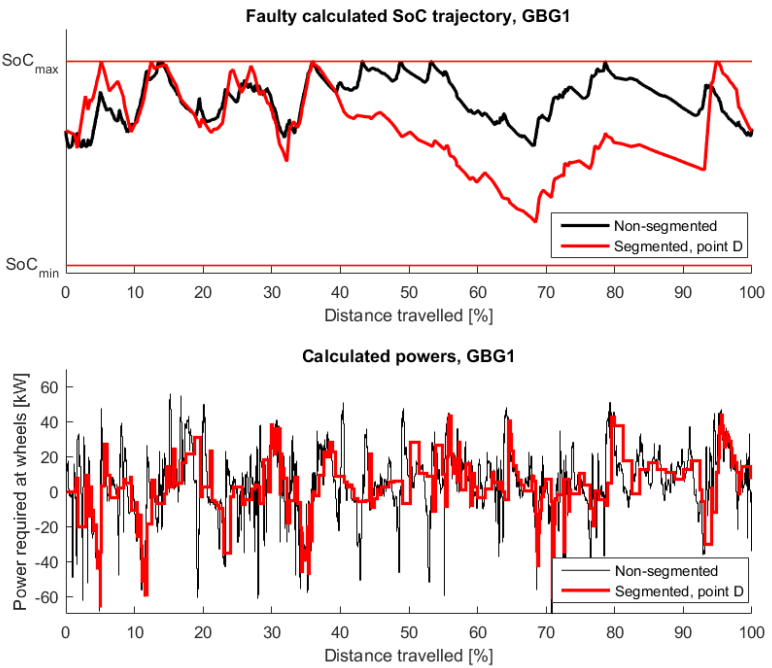

Figure 11.7 Calculated SoC trajectory for point D from Figure 11.4 and the corresponding power trend. At 93% of total distance travelled a negative power demand is held during a longer segment, causing an overestimation of the recharging capabilities.

The occurrence of these errors in the estimated power demands during a drive cycle show that, for coarser segmentation, the calculation of the power demands itself must consider that the calculated value can be held over long periods.

EOCM state trajectory

The three points A, B and C in Figure 11.4 represent different magnitudes of segmentation. The corresponding EOCM state trajectories obtained for these points are shown in Figure 11.8.

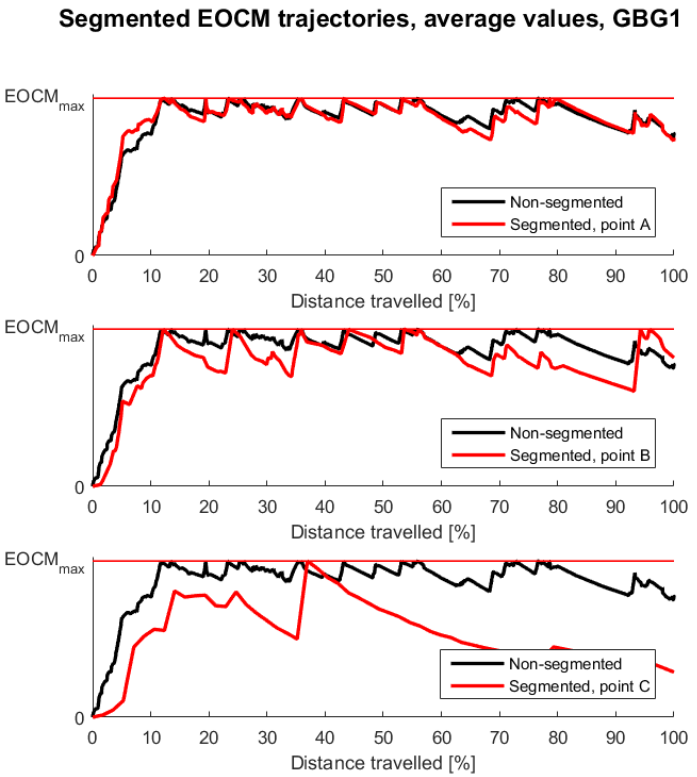

Figure 11.8 EOCM state trajectory obtained for the three points A, B and C in Figure 11.4.

Whereas the segmentation magnitude for point A, the 5 second segmentation, does not cause large discrepancies in the EOCM state trajectory it can be clearly seen how the state begins to be underestimated as the magnitude of segmentation increases.

11.3 Trajectory tracking

The trajectory tracking controllers introduced in Chapter 6 are tested in the advanced simulation software. The efficiency from this software is given in terms of CO_2 emissions per kilometer. As the emission of carbon dioxide is assumed to be directly related to fuel consumption, it is treated as such. The deviation from $SoC_{neutral}$ is corrected as explained in Section 6.5.

The λ_0 and μ_0 values for the PIDC, PIDST and TAN controllers are set to 2 and 0.03, respectively. The best choice for λ_0 is dependent on the drive cycle considered, however, when simulations are analyzed it is noted that the controlled value usually is in the range of 1.5-2.5. In order to have a guess that is appropriate for all drive cycles, it is set to 2. The value of μ_0 is set to a non-zero value as setting it to zero would allow the controller to use excessively high currents during recharge phase as there is no limit to the current drawn resulting in very rapid heating of the battery. The initial guess is set to a low value as to still allow reasonable magnitudes of the current to be utilized. The parameters are tuned for the drive cycle GBG1 using 1 second segmentation. The tuning is made as a balance between how well the trajectories are followed and how little fluctuation in the actuation there is.

The resulting trajectories for the tracking controller algorithm BN and PIDC compared to the Baseline controller is shown for the drive cycle MTN in Figure 11.9. It is seen that the Baseline controller does not prepare for the large recharging phase.

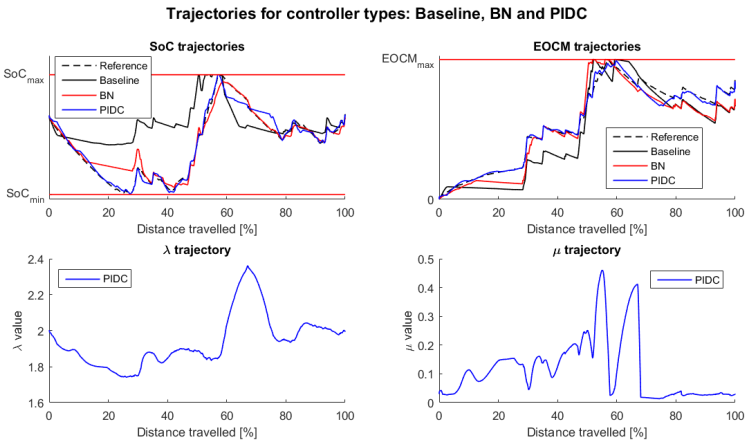

Figure 11.9 Resulting trajectories for different real-time controllers. Algorithm BN and PIDC compared to Baseline controller. Trajectory calculated from 1s sampling rate. MTN drive cycle. The Baseline controller does not prepare for the large regeneration phase.

The trajectory tracking algorithms are simulated on all drive cycles with different segmentation magnitudes. The results for each controller/segmentation permutation is shown in Table 11.3. The results correspond to the average values of the equivalent fuel consumption relative to Baseline controller from all drive cycles and therefore is also a reflection on how robust the control parameters remain.

Tracker	1s segmentation	10s segmentation	30s segmentation
BN	99.47%	99.43%	99.87%
PIDC	99.07%	99.97%	101.77%
PIDSC	99.01%	99.91%	102.17%
PIDT	99.68%	102,15%	107.56%
PIDST	99.62%	100,38%	102.80%
TAN	98.97%	99,45%	102.31%

Table 11.3 Equivalent fuel consumption between used tracker algorithm and the Baseline controller and absolute difference in the terminal SoC. Average values from all drive cycles segmented with 1 second, 10 second and 30 second sample rate.

The real-time controller yielding the best results in term of fuel consumption based on 1s segmentation of the drive cycle is the TAN controller. This is a consequence of how aggressive the TAN controller is set up. Only minor deviations from the trajectories are allowed before the corresponding value of the adjoint state is affected. Within the minor deviations the penalty is very low. While the fuel consumption results due to this aggressive tuning are good, the actuation of the ICE and ISG are not acceptable in terms of driveability. With an aggressive controller, positive fluctuations in the ISG (and ICE) actuation begin occurring.

In Figure 11.10 the fluctuations in the actuation is shown where it is compared to a corrected algorithm where the fluctuations are reduced. The positive fluctuation in the ISG actuation is a result of an aggressive tracker. The fluctuations occurring in the negative torque outputs are a result of a poor activation logic where the ECMS controller is sending a negative torque limit of regenerative torque and the current is being controlled by the Baseline controller. This logic kept switching between the two controllers and therefore induces fluctuations in the negative region of the ISG actuation.

It can be seen in Table 11.3 that while the TAN controller yields the lowest fuel consumption when the segmentation magnitude is low, the same controller does not perform as well as the BN, PIDC and PIDSC once the segmentation magnitude increases to 30s segmentation. This is once again the effect of following the calculated trajectories too aggressively. As the segmentation magnitude increases, the output trajectories deviate from the optimal trajectories. By being aggressive in the tracking of suboptimal trajectories, the fuel consumption increases. The controllers PIDC and PIDSC are not as aggressive, and thus perform better with coarse segmentation.

The controllers PIDT and PIDST do not perform as well for coarse segmentation. These controllers are characterized by using the output adjoint state trajectories

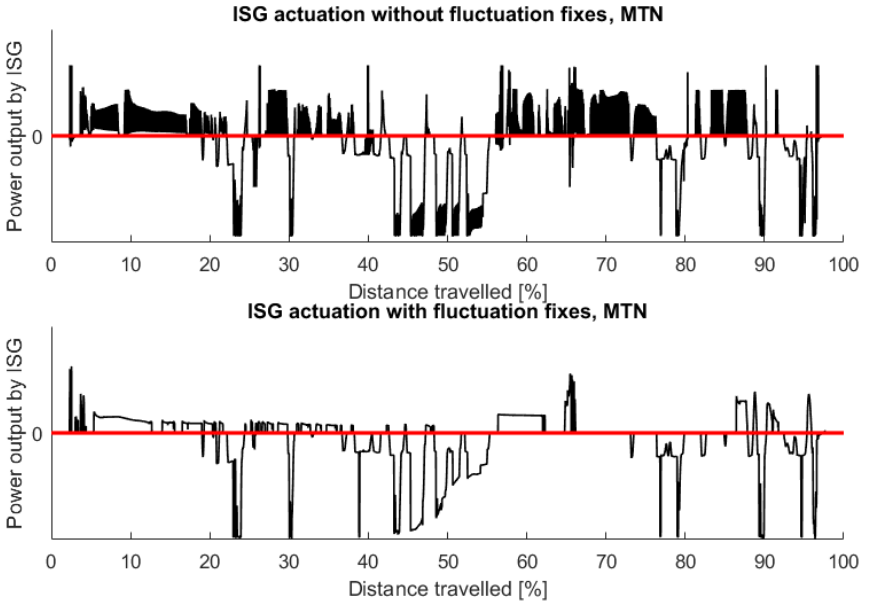

Figure 11.10 Top: ISG fluctuations occurring in both the positive and negative torque output regions before any attempt is made for correcting it. Bottom: ISG fluctuations after correction. MTN drive cycle.

from the convex optimization. As the segmentation magnitude increases, the output adjoint state trajectories begin deviate from the optimal ones. By using a constant guess on λ_0 and μ_0 the controllers are kept more robust. The reason that the constant guess controllers PIDC and PIDSC work better than the PIDT and PIDST even for fine segmentation is due to the interaction between the changing values of λ_0 and μ_0, and the PID-algorithm. As the PID-controller calculates new values of λ and μ for the next time step based on the error in states in the previous time step, the new calculated adjoint states should correct the current evolution of the SoC and EOCM states. But as the values of λ_0 and μ_0 change simultaneously, the calculated adjoint states are disturbed.

Comparing the PIDSC to PIDC and PIDST to PIDT, the controllers modifying the SoC trajectory in real-time according to the method explained in Section 6.3 show general improvement in fuel consumption.

With fine segmentation of the drive cycle, relative to the Baseline controller, using the PIDSC controller the fuel consumption can be reduced by up to 0.99% while keeping the amount of fluctuation in the actuation low. It is noted that the BN controller is the only controller yielding improvements in the fuel consumption relative

to the Baseline controller for the 30 second fixed sample time interval segmentation. The coarser the segmentation becomes the more rule based the controller needs to become. The problem with the BN controller is that it does not take into account the effect of the thermal state.

Importance of μ after EOCM peaks

In the drive cycle MTN the main region of concern for the optimization optimality is the discharge phase followed by a fast recharge phase up to the SoC limit at around 60% of the total distance travelled, as can be seen in Figure 11.9. After the EOCM state hits the upper limit, and until the end of the drive cycle, the upper limit is never reached again. It is investigated whether it is better to neglect the EOCM state after the first peak. A simulation using PIDC is ran on the MTN drive cycle where the μ value is kept at 0 after the first peak in the EOCM state has passed. The resulting trajectories for this are shown in Figure 11.11.

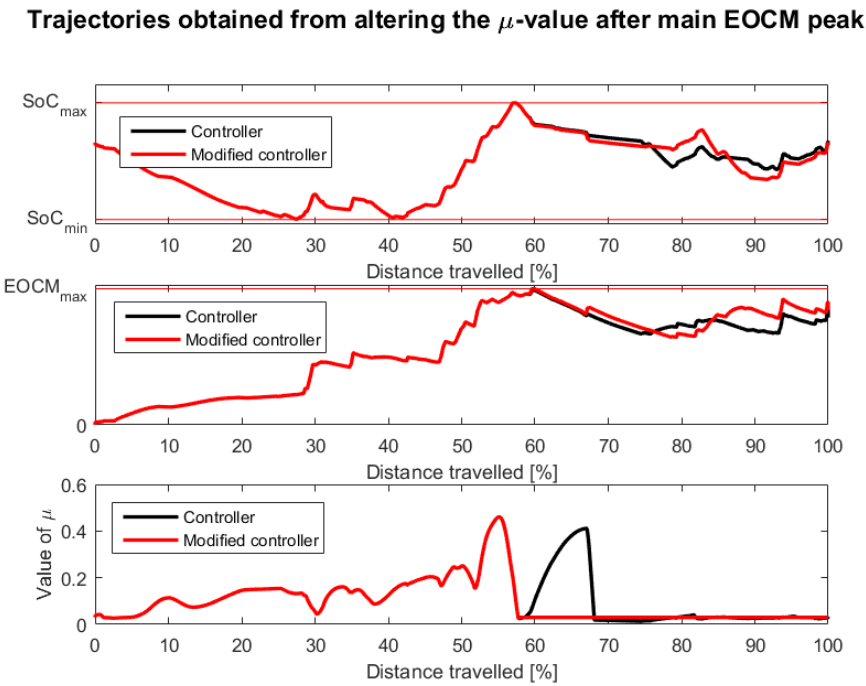

Figure 11.11 The effect on the SoC and EOCM state trajectories when μ is set to zero after the first peak in the MTN drive cycle is reached.

The question is whether there is any negative effect of still controlling and lim-

iting the EOCM state even though the upper threshold is never reached again in the remaining part of the reference trajectory. Relative to the original PIDC controller, the fuel consumption decreases by 0.07%. As there are no more concerns about the battery heating the upper thermal boundary, there is no benefit of limiting the currents.

11.4 Receding horizon

Here, the explored receding horizon methods are compared. If not mentioned otherwise, the weight for SoC as in (10.3) is set to $\vartheta = 0$. The value for the EOCM state is always set to $\vartheta = 1$ and a hard constraint for the terminal value of the EOCM state is added to the convex optimization. The hard constraint value is set to $EOCM_{max} - \delta$ as explained in Section 10.1.

The original receding horizon result is shown in Figure 10.1 where the terminal SoC for each horizon optimization is set to $SoC_{neutral}$. Figure 11.12 shows the resulting SoC trajectories for the horizons using horizon methods B and C.

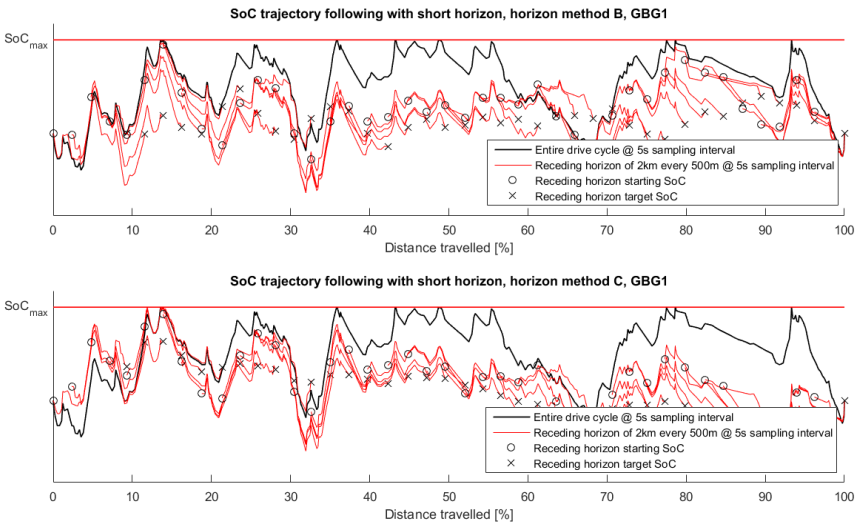

Figure 11.12 Top: SoC trajectories from method B implementation of a receding horizon. Bottom: SoC trajectories from method C implementation of a receding horizon.

The difference between the horizon method B and C is the set terminal SoC for the optimization done for each horizon. The idea is to match the used horizon trajectory with the trajectory that is obtained when optimizing for the entire drive cycle.

Horizon method B shows more deviation from $SoC_{neutral}$ as compared to method A. The results for horizon method C show that during the first half of the drive cycle the obtained SoC trajectory from the horizons is close to the optimal SoC trajectory, but it begins to undershoot the optimal trajectory during the second half of the drive cycle. This is a consequence of how the cost of the terminal SoC deviation is set. A high cost on the deviations results in mimicking a hard constraint, and a too low of a cost causes the suggested terminal SoC for the trajectories to become too low. This short-sighted optimization penalizes the overall fuel consumption as during the final horizon the SoC is forced back up in order to remain charge-sustaining at the end of the drive cycle.

Figure 11.13 shows the output horizon trajectories for the horizon methods D and E. As the terminal SoC for the current horizon in version D is dependent on the current value of SoC, the terminal SoC deviation from $SoC_{neutral}$ is higher. The thought is that this causes the battery utilization to be higher, and thus the fuel efficiency to be higher as well.

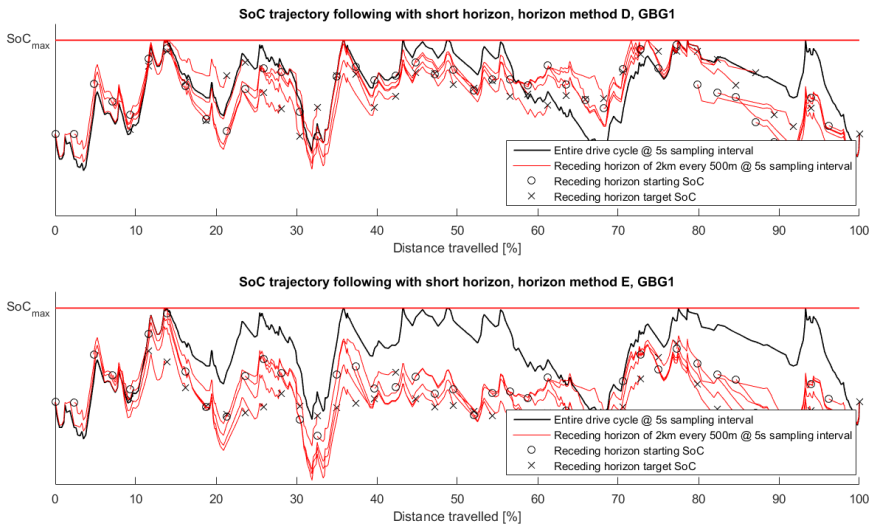

Figure 11.13 Top: SoC trajectories from method D implementation of a receding horizon. Bottom: SoC trajectories from method E implementation of a receding horizon.

The trajectories that best follows the perfect SoC trajectory are the trajectories produced by horizon method D in Figure 11.13. As new trajectories are calculated and used before the previous horizon has ended, only the followed trajectory is of interest. The used trajectories for horizon method D and method A compared with

the perfect trajectory can be seen in Figure 11.14 and Figure 11.15, respectively. It is noted that the trajectory followed from horizon method A is more conservative and remains near the $SoC_{neutral}$ region as the method D follows the perfect trajectory more.

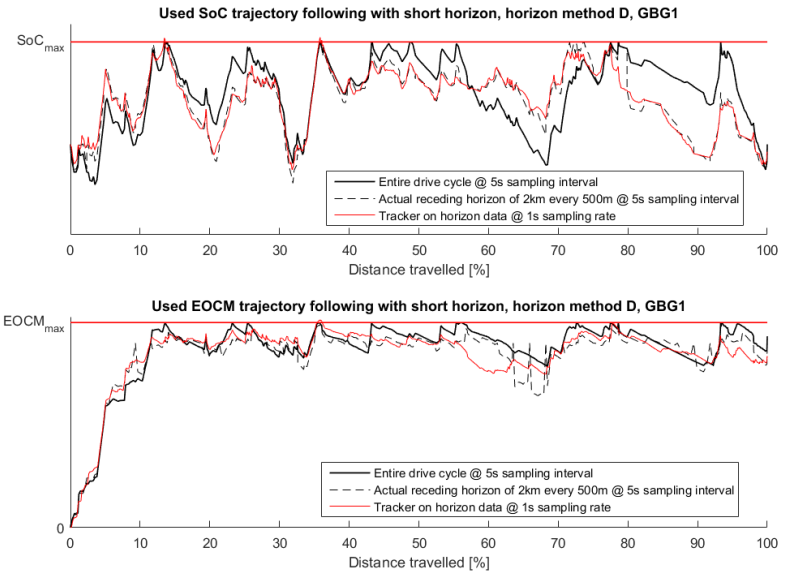

Figure 11.14 Actual SoC and EOCM state trajectory used for the real-time controller calculated on receding horizons according to version D.

The relative fuel consumption obtained from the different versions of receding horizon are shown in Table 11.4 for different recalculation intervals and different horizon lengths. The interesting aspect is that, although horizon method D follows the perfect trajectory the closest, it is not the horizon method that performs the best. The method performing best is the horizon method E. Receding horizon method E performs better than method A as the battery is better utilized depending on the future trends. While method A would prevent the battery from draining/charging more, the horizon method E allows a little more deviation from $SoC_{neutral}$. When horizon method E is compared with method D, the perfect trajectory is not followed equally well. It can be seen that for horizon method D in Figure 11.14 at 35% distance travelled, the SoC saturates at the upper threshold and actually crosses it as well. The battery is not drained enough before the recharging phase and the upper limit is hit too early, this results in a large penalty on the fuel consumption as that recharging energy is not captured. The reason for the battery not being sufficiently

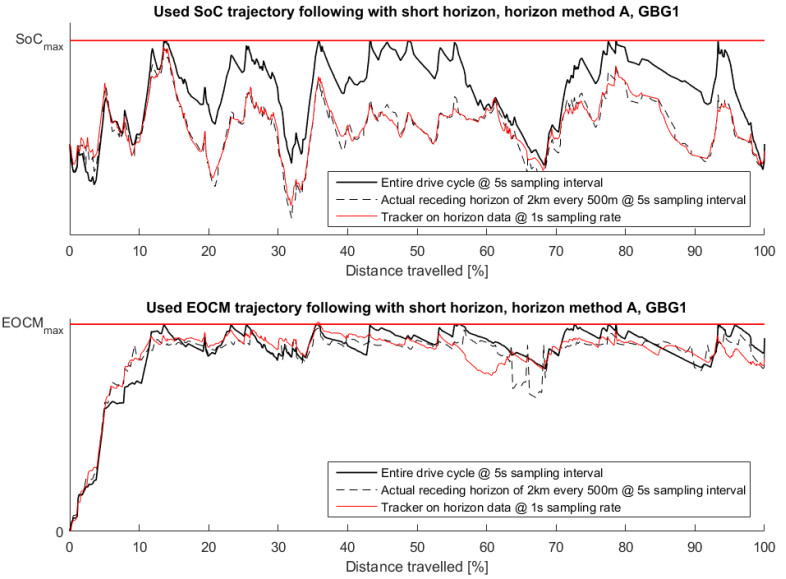

Figure 11.15 Actual SoC and EOCM state trajectory used for the real-time controller calculated on receding horizons according to version A.

discharged before the recharge phase is due to the setting of terminal SoC values for the horizon calculation. The method D accumulates the power trends, whereas method E does not. Essentially, the SoC is to be allowed to deviate from $SoC_{neutral}$, but at the same time not too much as that can cause the battery to be in such a state that it saturates easily as it does not get enough time to prepare for future recharging phases.

Recalculation interval [m] / Horizon length [m]	A	B	C	D	E
5000/20000	99.92%	99.92%	99.92%	99.92%	99.92%
5000/10000	100.07%	100.08%	100.08%	100.06%	100.02%
2500/5000	100.09%	100.11%	100.14%	100.03%	100.02%
1000/4000	100.01%	100.07%	100.05%	100.11%	100.02%
500/2000	100.34%	100.40%	100.64%	100.42%	100.28%
500/1000	100.88%	100.96%	100.83%	101.19%	100.68%
250/500	101.77%	102.03%	101.49%	101.81%	101.26%

Table 11.4 Fuel consumption for drive cycle GBG1, relative to the fuel consumption when using the trajectory based on the entire drive cycle.

New/old horizon trajectory use ratio and delays in calculations

The effects of the weighting factor and induced delays in the calculations of the horizon trajectories on the fuel consumption are shown in Table 11.5.

delay [s] / ϑ	Relative fuel consumption
0/0	100.00%
0/0.25	100.03%
0/0.5	100.12%
0/0.75	100.22%
0/1	101.05%
4/0	100.02%
7/0	100.06%

Table 11.5 Fuel consumption for changing the ϑ and delay in the receding horizon implementation relative the fuel consumption obtained with no delay and $\vartheta = 0$. Drive cycle GBG1, using horizon method E with sampling rate of 500m and horizon length 2000m.

For the weighting factor for SoC, it can be seen that 0 is the best. If the weighting factor approaches 1, only the older trajectory is used and for SoC this results in worse fuel consumption as using old trajectory values do not prepare the battery for the near future as well as the current horizon calculation results do.

The induced delays in the calculations of new trajectories cause no significant increase in fuel consumption. This is a result of the old horizon trajectory already being followed well, and as the new horizon trajectory always begins with the SoC set to the current value, there is a small difference in the actual followed trajectory. Furthermore, as the drive cycle duration is significantly longer than a realistic delay in the optimization calculations, the influence on the overall fuel consumption during the entire drive cycle is minimal.

11.5 Drive cycle merging

The results for merging drive cycle information from multiple vehicles are compared to the results obtained from assuming the velocity and altitude profile according to GBG1, and also the optimal trajectories for each drive cycle. The GBG1 drive cycle is characterized by the driver driving normally. The comparison between the amalgamation of drive cycle information and the drive cycle information from the regular driver is whether the merging has any positive effect on the predictability of the power requirements.

Table 11.6 shows the RMS errors between the used SoC trajectories and the perfect SoC trajectories for specific drive cycles.

The output trajectory obtained from merging the drive cycle data is shown in Figure 11.16 where it is compared to the optimal trajectory for each driving style.

Used input data	GBG1	GBG2	GBG3	GBG4	GBG5	**Average**
Average input data	0.0335	0.0406	0.0272	0.0499	0.0289	**0.0360**
Data from GBG1	(0)	0.0449	0.0269	0.0309	0.0440	**0.0367**

Table 11.6 RMS of error between used SoC trajectory and perfect SoC trajectory.

The data point positions used by the optimization tool are taken directly from the segmentation of GBG1.

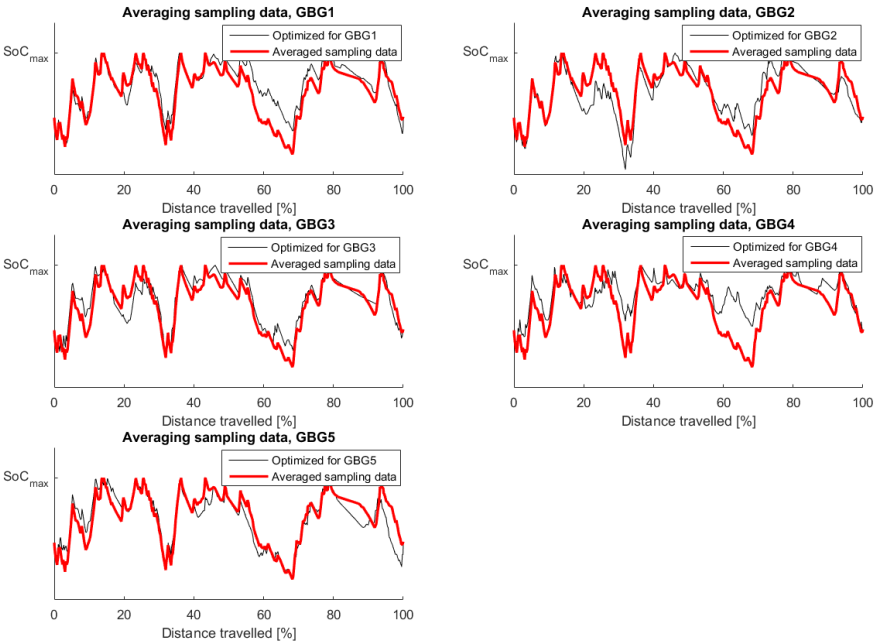

Figure 11.16 The resulting SoC trajectory from interpolating and averaging the data points. The overall trends are the same, although a larger discrepancy occurs for GBG4 where a regeneration phase is expected but does not occur.

The optimal trajectory for GBG1 compared with all considered drive cycles is shown in Figure 11.17.

By merging the route information from multiple drives, there is, as expected, an error in the output SoC trajectory as compared to one optimized for a single driving style. As can be seen in Table 11.6, the average deviation from the optimal trajectory is slightly better than that obtained from using the GBG1 trajectory. Only in the drive cycle GBG4 does the averaging of the data input perform significantly worse than the method of using the GBG1 SoC trajectory. This large error can be seen

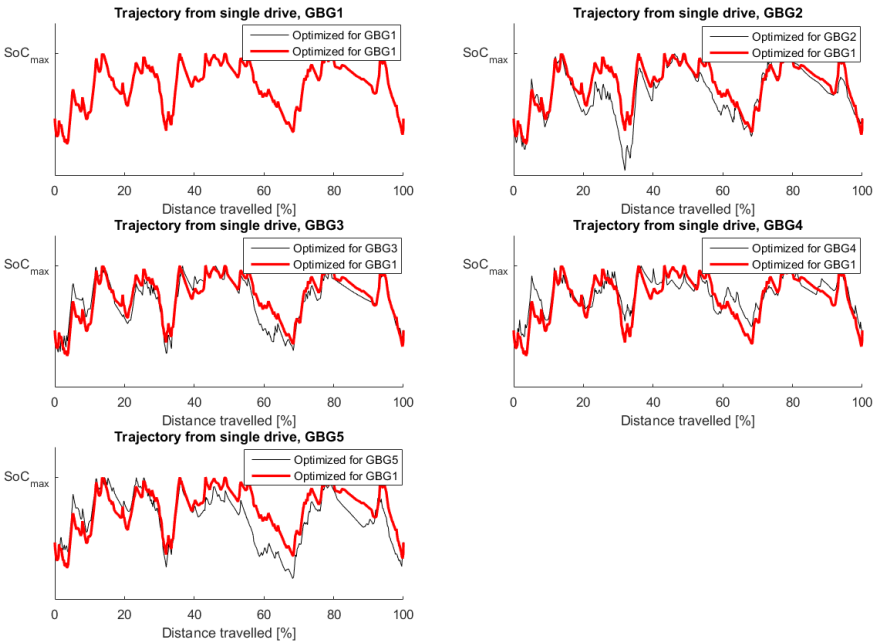

Figure 11.17 The resulting SoC trajectory from GBG1 used on other drive cycles.

in Figure 11.16 where for the GBG4 drive cycle at around 70% distance travelled, there is an expected recharging phase. In the GBG4 drive cycle, however, the driver is driving aggressively and the recharging region is significantly reduced and thus the two SoC trajectories differ in that region. In the remaining part of the considered drive cycle, however, the overall battery use trend is followed.

The RMS error due to the aforementioned absence of a recharging phase in GBG4 is low for the trajectory produced by GBG1. This is because, as can be seen in Figure 11.17, the expected SoC trajectory is first overestimating the optimal trajectory. The low overall error in the trajectory is therefore not due to similar trends being expected.

By using the GBG1 drive cycle, there is a large sensitivity to how representative the lone drive along the route is of the average driver. To represent the expected velocity profile for a certain route using information from one drive across it is necessary for that driver to have normal driving characteristics, which can be difficult to imitate. Additionally, different weather and road conditions, amongst other factors, could occur for the average driver whereas for the reference driver they do not. If the characteristics of a route change after the expected velocity and altitude profiles for a route have been measured, due to for example road construction or a

change in the road itself, the real power demand trajectory will thereafter no longer be approximate to the measured trajectory. By using the method of merging route information, however, the power loads can be kept up-to-date. Although the method of merging information may not represent the outlying drivers, such as the considered aggressive driver, it has the potential to improve the calculated power demand throughout considered drive cycles. As the data set of drives increases, the velocity (and altitude) planning for the average driver can be improved. Furthermore, as it is the velocity and altitude trajectories that are stored, and not the power demand trajectory itself, the actual vehicle parameters, such as mass and auxiliary power, can be input to the optimization to tailor the calculated future power demands to each considered vehicle.

11.6 EOCM state estimation learning

For the EOCM state learning algorithm the learning rate β value for the α calculation as in (9.3) is set to 0.5 as to have a large influence on the EOCM trajectory. The algorithm is iterated three times on the MTN drive cycle segmented according to segmentation method 5 (see Table 7.3) using the PIDT controller, the results are shown in Figure 11.18.

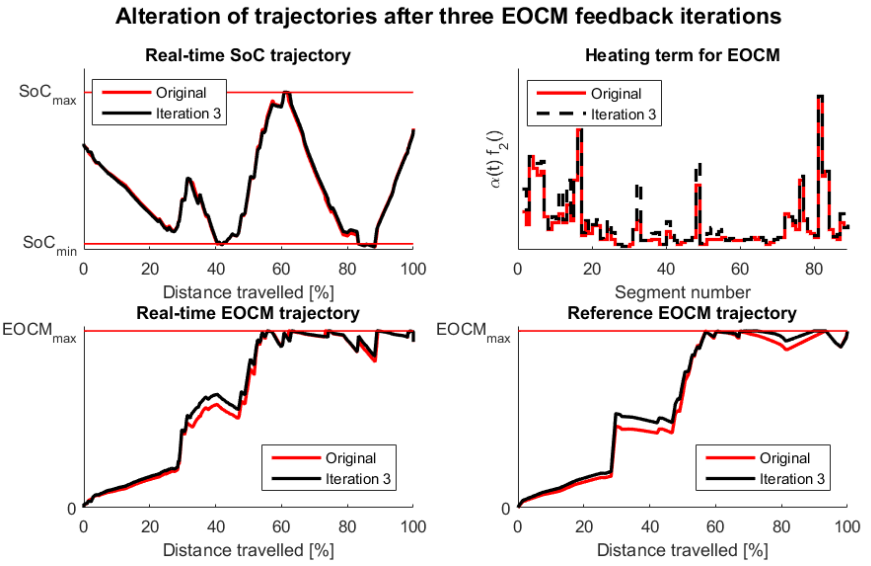

Figure 11.18 Simulation results for the third iteration of the EOCM state feedback learning compared to the original results without any learning.

The fuel consumption for the third iteration increases by 0.1% relative the original. The explanation for the small increase instead of a significant decrease in fuel consumption is that an increase in the value of α only implies a higher heating factor for the considered segment. The idea behind using α is to increase the EOCM state evolution during the segments where it is underestimated, this is however not the outcome obtained by introducing α as an input to the convex optimizer. What the optimization tool sees is that using battery current during certain segments results in even higher increases in the EOCM state. This does not necessarily mean that the optimal solution is to keep the same battery currents during those segment and having the EOCM state increase. Instead, the convex optimization algorithm leads to a solution where the battery usage becomes more expensive for those segments, and the resulting trajectories having little to no effect on the state estimation as the optimum solution is to use lower current magnitudes during those segment.

11.7 Hardware limitations

As the on-board battery and the used EM are, if compared to a fully electric vehicle, small in size, a question arises whether the hardware limits are the limiting factor of the possible fuel savings yielded by the optimization tool. It is also of interest to pinpoint which component, if any, is the bottleneck. The three main hardware limitations listed below are scaled up and the new fuel consumption results obtained by running the same optimization tool are analyzed. The three hardware thresholds are the following:

- Battery and EM maximum power output thresholds ($P_{ISG,max}$ $P_{ISG,min}$, $P_{batt,max}$, $P_{batt,min}$)
- Battery energy capacity (E_{batt})
- Battery EOCM state upper limit ($EOCM_{max}$)

The resulting fuel consumption for the scaling of the three hardware thresholds is shown in Figure 11.19 and Figure 11.20 for the drive cycles GBG1 and MTN, respectively. The fuel consumption shown is relative the fuel consumption obtained by the optimization tool without any scaling of the limits.

It is seen that by increasing the $EOCM_{max}$ value, the fuel consumption decreases for almost all scenarios. Conversely, if the EOCM state upper threshold is not increased, irrespective of how much the battery capacity and the output powers are increased, there will be minimal decrease in fuel consumption. Already at a doubling of $EOCM_{max}$ does the fuel consumption reduction stagnate. The reason for this is that the dissipation term f_1 in (3.24) becomes all the more dominant term as the absolute value of the EOCM state increases.

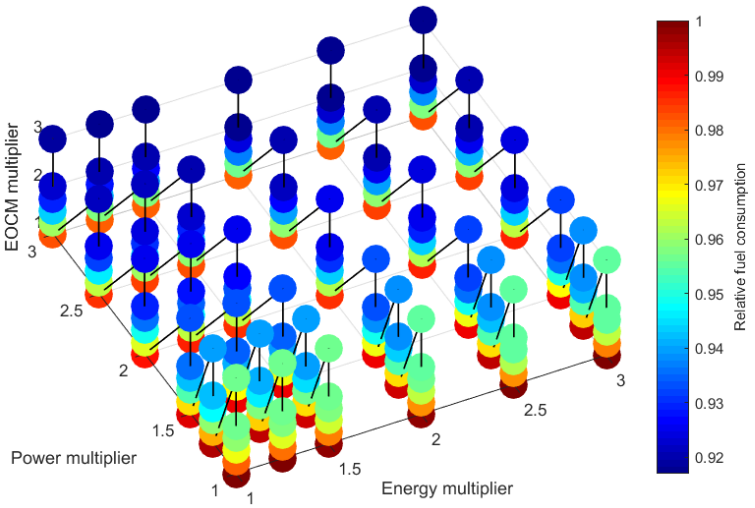

Figure 11.19 Power output, electrical energy storage and EOCM state limitations influence on fuel consumption, drive cycle GBG1.

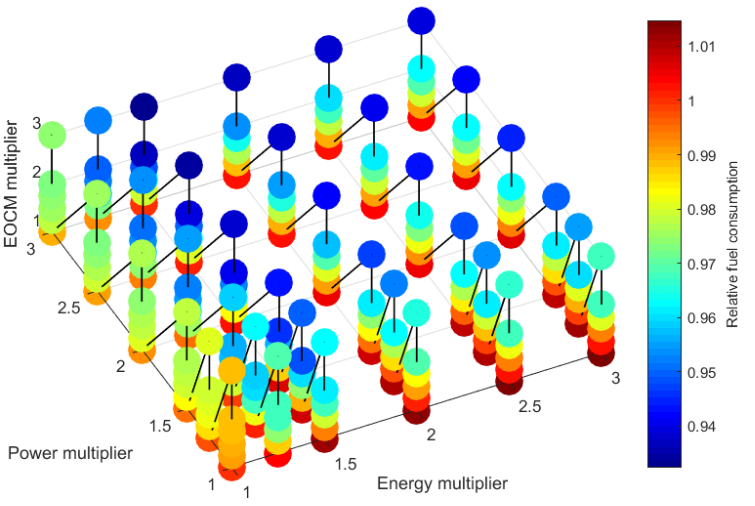

Figure 11.20 Power output, electrical energy storage and EOCM state limitations influence on fuel consumption, drive cycle MTN.

For the MTN drive cycle an important threshold that also becomes a limiting factor for the fuel consumption reduction is the energy storage capacity in the battery. This is a consequence of the large downhill section in the drive cycle. With the current setup, the optimal SoC trajectory always hits the lower threshold before and the higher threshold after the downhill section.

For the GBG drive cycle the effect of EOCM limit becomes important as in that cycle which corresponds to a mixture of city and highway driving has more fluctuating power demands leading to the EOCM state reaching higher values easily and thereby limiting current usage frequently.

The lowest fuel consumption due to hardware limits in both drive cycles is already reached at a power multiplication factor of 2, energy multiplication factor of 1.5 and EOCM state upper threshold multiplication factor of 2. This fuel consumption corresponds to approximately 93% of the consumption using current hardware.

12

Conclusion and future work

In this thesis, the optimization of the power-split in a mild hybrid vehicle has been studied. By introducing the battery thermal state into the optimization algorithm a lot of complexity is added to the system as the two adjoint states controlling SoC and the EOCM state affect each other. An initial guess for the two adjoint states in combination with a control feedback on both terms proves sufficient to make a feedback-based energy management system. When using fine segmentation an improvement over the Baseline controller, in terms of fuel consumption, is made with all of the explored controllers. By using the explored tracking controllers, segmentation of up to 10 seconds provides improvement over the Baseline controller. Segmentation of 30 second intervals causes almost all strategies to perform worse than the Baseline controller. The Baseline New tracker using the trajectories calculated on the coarse segmentation performs better than the Baseline controller.

The tuning of the controller parameters were done for the 1 second fixed sample time interval segmentation and used directly for the 10 second and 30 second segmentation methods. With decreased amount of data, and less trustworthy trajectories, the parameters should be tuned accordingly. If used on a real vehicle, different levels of aggression in the real-time tracking controller should be tested as model approximations in the used simulations might show different results than in reality.

It is important to keep in mind that as the optimization is done on the simplified models of the powertrain, an optimal trajectory according to the MATLAB models may not be the optimal trajectory in reality as the real world scenario due to the fact that model non-linearity as well as driver behaviour impacts the power demand. The power demands should become even more unpredictable in actual testing as many unplanned scenarios such as different drivers and traffic conditions will change the optimal trajectory significantly. Further exploration can be made on the analysis of the vehicle models and how the magnitude of simplification and linearization of the vehicle components affect the output trajectories and how these compare when used in a real vehicle. Furthermore, as has been mentioned in Section 3.4, the effects of inertia and friction losses in the vehicle are neglected and thus the calculated expected power demand throughout a drive cycle should be underestimating the real values. The auxiliary power is also a factor which plays a role as currently it is

being considered as a constant whereas it does not necessarily have to be. As the on-board battery is not very large, the effects from the auxiliaries on-board should not be neglected. The auxiliary loads will vary from vehicle to vehicle based on weather conditions and the personal preferences of the driver. Even though the auxiliary loads may be different each time the vehicle travels through a drive cycle, it could be assumed even on real vehicles that the auxiliary loads are constant throughout drive cycles. The average auxiliary load ever since starting the vehicle may be used to adapt to the current weather, driver and vehicle conditions. Additionally, a multiplication factor can be calculated based on the difference between the actual measured power outputs in the vehicle and the calculated powers based on vehicle velocity and inclination angle in order to obtain power demands that closer represent the real vehicle.

Some of the segmentation methods did not include any need of the information about the actual vehicle velocity or the inclination angle. The segmentation based on speed limits and differences in altitude could be made solely on information obtained by available maps. The expected power demand of the vehicle, however, were still calculated on the vehicle speed and inclination angle, and thus the sampling of data from individual vehicles is still required. One way of removing the vehicle measurements is to calculate the power demands from the speed limits, the differences in speed limits, and the difference in altitude using geographical data. One issue that then arises is the estimation of the real vehicle speed. The vehicle velocity trajectory is required in order to obtain a gear trajectory and rotating speeds of the powertrain components.

Of all the segmentation methods explored in this thesis, the method performing the best in the considered drive cycles was the fixed sample time interval. This segmentation provided least error in the produced trajectories during increasing segmentation magnitude. As mentioned in Section 11.2, the GBG1-GBG5 drive cycles on which the segmentation methods were benchmarked on might be biased towards fixed sampling rate segmentation as it contains many regions of intricate power demand and velocity trends with parts of both highways as well as city driving. If the considered drive cycles were instead mainly done on highways or countryside roads, a different segmentation method could possibly outperform the fixed sample time interval segmentation. The possibility to use location data to switch between different segmentation methods could be explored where the segmentation method used for roads inside cities and towns could be fixed sampling rate, while for long and uneventful roads a different segmentation method could be used. One problem that has to be considered is the varying underestimation of the EOCM state depending on the segmentation magnitude.

The merging of data from different driver styles on the same route show good results, where the overall battery state trends can be predicted for the usual driver. Additionally, by using data from previous drivers to predict the power demands and expected velocity of the vehicle for a certain route, the optimization tool can adapt to new events and changes in the route. By having a large database of many drivers,

specific trends may found and used for increased predictability, such as the varying general driver characteristics depending on time of day and day of the week.

The estimation of EOCM, as explored in Chapter 9, show little improvement in terms of fuel consumption. This has to do with how the learning is implemented, as the convex optimizer sees larger penalties for using battery current during certain segments instead of seeing the increase in the α values as inaccuracies in the power demand during the segments.

The optimization tool can be used for receding horizon implementations without any large modification. As can be seen in Table 11.4 for receding horizon method E, the effect of reducing the horizon length down to 4km is insignificant. Below this horizon length, however, the fuel consumption begins to increase significantly. Even for a horizon length of 1km and a calculation of a new horizon every 500m, however, the fuel consumption penalty is 0.68%.

By increasing the hardware thresholds, as explained in Section 11.7, it is possible to see how the fuel consumption reduction is limited by hardware. In Figure 11.19 and Figure 11.20 it is seen how little the fuel consumption decreases if the EOCM threshold is not raised. The effectiveness of the studied optimization tool can thus be higher if the hardware thresholds are increased.

Overall, this thesis shows that there is potential in making a predictive energy management system for mild hybrids. The effectiveness varies according to the number of regenerative areas in the considered drive cycle and the considered segmentation magnitude. The algorithm could be modified to incorporate the current rule-based logic so that when poor future information is known the Baseline controller is used. Conversely, the investigated optimization tool structure can be integrated to the current implementation of the EMS and be used when reliable future road information is available.

Bibliography

Alagarsamy, T. and B. Moulik (2018). "A review on optimal design of hybrid electric vehicles and electric vehicles". In: *2018 3rd International Conference for Convergence in Technology (I2CT)*. IEEE, pp. 1–5.

Almgren, J. and G. Elingsbo (2017). *Route based optimal control strategy for plug-in hybrid electric vehicles*. eng. Linköpings universitet, Institutionen för systemteknik, Fordonssystem.

Berntsson, S. and M. Andreasson (2018). *Efficient route-based optimal energy management for hybrid electric vehicles*. eng. Linköpings universitet, Institutionen för systemteknik, Fordonssystem.

Böhme, T. J. and B. Frank (2017). "Hybrid systems, optimal control and hybrid vehicles". *Cham, CH: Springer International*.

Boyd, S. P. and L. Vandenberghe (2004). *Convex optimization.* Cambridge : Cambridge University Press, 2004. ISBN: 0521833787.

EIA (2015). *Passenger travel accounts for most of world transportation energy use*. URL: `https://www.eia.gov/todayinenergy/detail.php?id=23832#` (visited on 2019-05-29).

Guzzella, L. (2007). *Vehicle propulsion systems introduction to modeling and optimization*. eng. Second Edition.. Springer, Berlin. ISBN: 1-281-35535-6.

Larsson, V. (2014). *Route optimized energy management of plug-in hybrid electric vehicles*. Chalmers University of Technology.

Murgovski, N., L. Johannesson, J. Sjöberg, and B. Egardt (2012). "Component sizing of a plug-in hybrid electric powertrain via convex optimization". *Mechatronics* **22**:1, pp. 106–120.

Paganelli, G., G. Ercole, A. Brahma, Y. Guezennec, and G. Rizzoni (2001). "General supervisory control policy for the energy optimization of charge-sustaining hybrid electric vehicles". *JSAE review* **22**:4, pp. 511–518.

Sivertsson, M., C. Sundström, and L. Eriksson (2011). "Adaptive control of a hybrid powertrain with map-based ecms". eng. In: vol. 44. 1, pp. 2949–2954. ISBN: 978-390266193-7.

Tobias, N., E. Philipp, F. Michael, O. Christopher, and G. Lino (2014). "Convex optimization for the energy management of hybrid electric vehicles considering engine start and gearshift costs." *Energies, Vol 7, Iss 2, Pp 834-856 (2014)* 2, p. 834. ISSN: 1996-1073.

Volvo Car Group (2011). *Integrated starter generator (isg)*. Volvo Car Group. URL: `https://www.media.volvocars.com/global/en-gb/media/pressreleases/5278` (visited on 2019-05-10).

Wirasingha, S. G. and A. Emadi (2010). "Classification and review of control strategies for plug-in hybrid electric vehicles". *IEEE Transactions on vehicular technology* **60**:1, pp. 111–122.

Zhang, P., F. Yan, and C. Du (2015). "A comprehensive analysis of energy management strategies for hybrid electric vehicles based on bibliometrics". *Renewable and Sustainable Energy Reviews* **48**, pp. 88–104.

A

Inclination angle offset correction

One of the measured signals used for calculating the power demand during a drive cycle is the vehicle inclination angle of the vehicle, θ. It is noticed during data analysis that in the collected data for the driven drive cycles GBG1-GBG5 the signal for the inclination angle is slightly offset. The θ signal gives a slightly negative value (vehicle aiming downwards) when the vehicle is positioned on what seems to be a horizontal surface. The explanation for this might be that the assumed horizontal surface is not perfectly horizontal, but the effects of an offset are also measured in altitude calculations. The θ signal is also used for altitude estimation. If the initial altitude is known (it is assumed to be known due to available GPS receiver), the sine of the inclination angle can be multiplied with the instantaneous vehicle velocity and the product can then be integrated over time to yield an altitude profile. By doing this, the resulting altitude profile reveals a drift into negative altitudes. Since it is known that at the end of the drive cycle the altitude is the same as the initial altitude (thus, the net altitude climb is 0), the integration of the θ signal together with the vehicle velocity should give an ending altitude near the initial value. Due to a noticed large negative drift, the sensor is concluded to have a negative offset error. This offset error does not necessarily have to be constant during the drive cycle.

Since the θ signal is of vast importance for the calculations of the vehicle power demands, and thus the optimal SoC and EOCM state trajectories, there is an effort made to correct and compensate for the offset in the signal. Three methods are explored:

- calculate the θ angle from altitude differences,
- add a constant to the original θ signal to compensate for the offset, and
- compensate the θ signal with error between feedback from the calculated altitude and measured altitude.

In the first method the differences in the altitude must be accurate and not noisy, therefore a high accuracy pressure sensor is used to estimate altitude differences by measuring the differences in pressure. The signal extracted from this sensor is processed by a FIR low-pass filter to reduce noise (which cause huge peaks in calculated θ values). This method of calculating the inclination angle yields values that follow the overall trends, but as can be seen in Figure A.1, there are large fluctuations still. This is due to how much the original signal is filtered. If the filtered signal was even less fluctuating, the yielded θ values would begin neglecting short-term fluctuations.

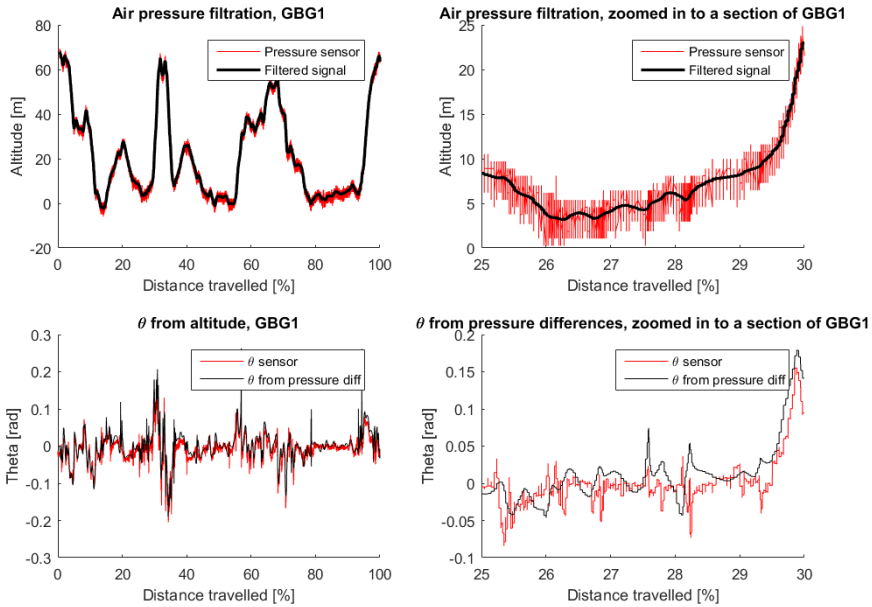

Figure A.1 Altitude and calculated θ from differences in altitude. Left side for an entire drive cycle, right side zoomed in on one section. The overall trends of θ is followed (bottom left) but the short-term values are not matching (bottom right).

The second alternative for fixing the offset error is by adding a constant to the signal. As mentioned before the offset error does not have to be constant, but could also vary with the actual inclination angle of the vehicle among other factors. The offset error can also have different characteristics between the vehicle models, and also between individual vehicles as the sensor may be calibrated differently during the production. The constant offset compensation value for the θ is found by matching the final altitude from the integrated compensated theta signal to the known

value. The major flaw with this method for compensating the offset is that the offset may fluctuate during drive cycles, and also that the calculation for the best constant offset value requires hindsight due to the compensation value being calculated from the terminal altitude.

The found constant offset values for the driven drive cycles are shown in Table A.1.

Drive cycle	Constant offset for θ
GBG1	0.0093
GBG2	0.0055
GBG3	0.0044
GBG4	0.0033
GBG5	0.0049
Average	**0.00548**

Table A.1 Constant θ compensation value for the different drive cycles. For the same vehicle and setup, the found constant offset varies significantly between different drive cycles.

The third method of correcting the θ signal is by using a pressure sensor and the differences in measured pressure as a compensator for the θ signal. The idea is that the altitude obtained by the integrated product of θ and vehicle velocity can be compared to the altitude obtained by a pressure sensor, and the error can then be used as feedback to a PI-controller correcting the θ signal.

There is an available signal for the pressure that is used within the engine. The ambient air pressure is measured. This signal can be used to measure the overall altitude trend, and thus could be used as a compensator for the offset in the θ signal. The pressure sensor in the vehicle is digital and can only detect large altitude variations. The reasoning for using a lower quality pressure sensor is that in this implementation it is not used directly to calculate the inclination angle, but the measured pressure is acting as a compensation reference and only the overall low frequency trends are needed. This pressure measurement is already integrated in the vehicles, and thus no expensive external high accuracy air pressure sensor is needed. The resulting θ signal obtained from this approach is shown in Figure A.2.

The resulting compensation from the PI-controller is shown in Figure A.3 where it is compared to the found constant offset for GBG1.

The results from the calculated positive powers using differently compensated values of θ compared to the measured positive ICE signal in the vehicle used for creating the drive cycles are shown in Table A.2.

These results show that the constant offset solution to correcting the θ signal works the best. The reason for this is that since the calculated powers always underestimate the true output power due to neglected inertia and component losses, the θ offset compensator that yields highest values on the θ signal itself will yield the

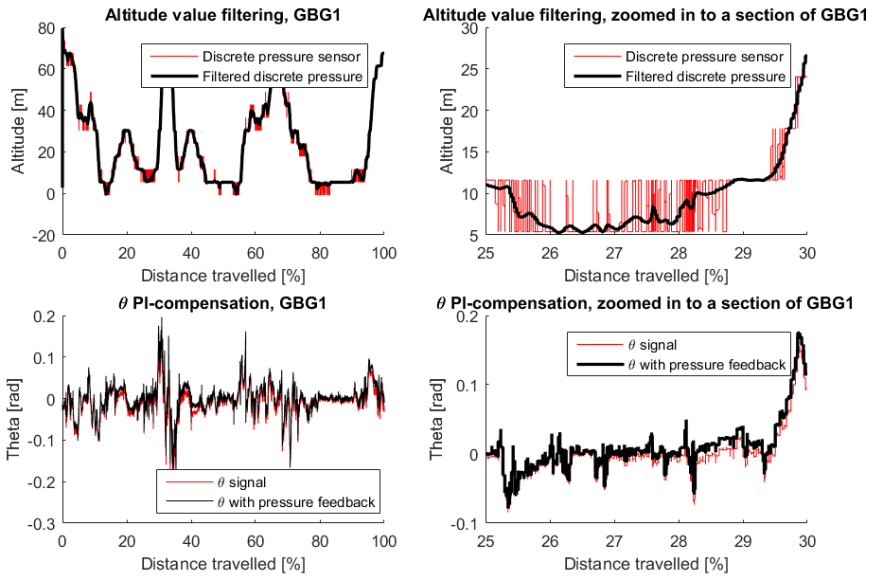

Figure A.2 Filtered altitude from pressure sensor, compensated θ from differences in altitude. Left side for an entire drive cycle, right side zoomed in on one section.

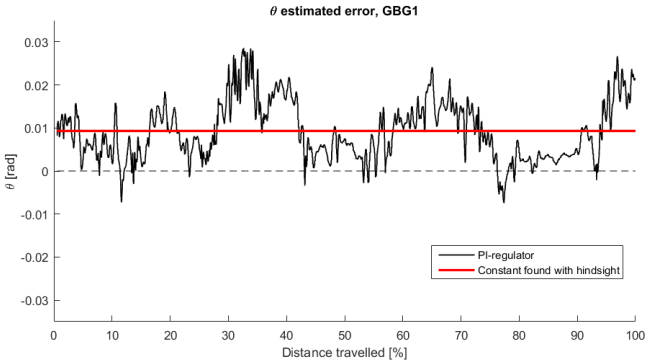

Figure A.3 Found offset in the θ signal using different estimation methods.

closest calculated powers to the actual power output. Modifying only the θ so as to match the calculated powers with the real measured power would be an incorrect approach as the effects from inclination angles would be overestimated relative to the velocity energies. The fact that the θ signal still has an offset is not neglected.

θ signal correction	GBG1 RMS [e+03]	GBG2 RMS [e+03]	GBG3 RMS [e+03]	GBG4 RMS [e+03]	GBG5 RMS [e+03]
Original θ	5.5	3.9	4.9	7.2	4.8
Constant offset from individual GBG	4.7	3.5	4.6	6.9	4.4
Constant offset from average	5.0	3.5	4.5	6.7	4.4
PI-compensated	5.1	3.8	4.8	7.2	4.6

Table A.2 RMS errors between the measured positive ICE signal in the real vehicle and the positive calculated power using different methods of calculating θ.

The PI-controller adjustment is suggested to be implemented. Due to the offset not being constant, this method of correcting the θ signal should be more adaptable for a general implementation as each vehicle compensates for their own unique offset that changes even during drive cycles. This adjustment causes minimal offsets in the θ, and thus causes the power calculations to be more correct. The altitude profile obtained from integrating the θ signal using different compensation methods is shown in Figure A.4.

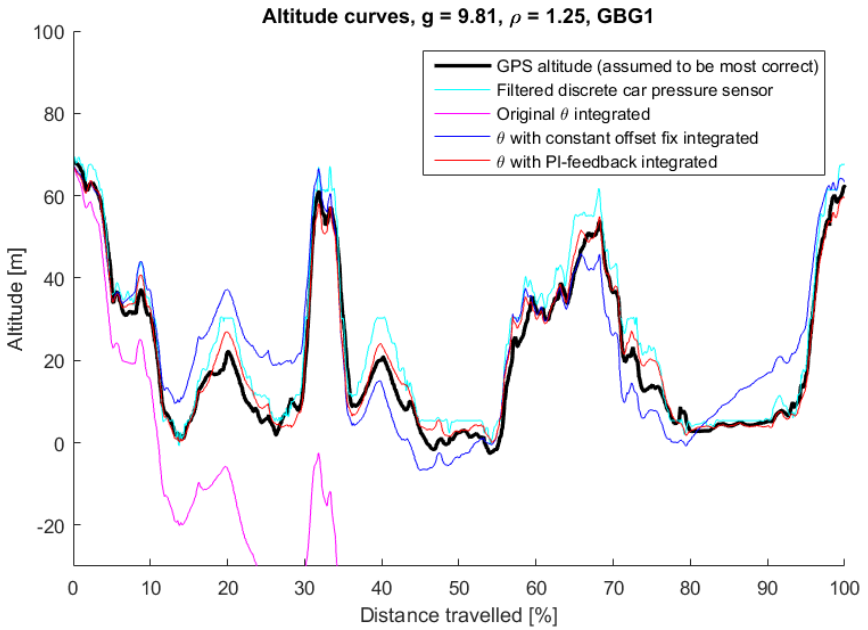

Figure A.4 Altitudes calculated by different methods. The original θ has a negative offset which causes the altitude calculation to drift off into the depths of hell. The correctly compensated θ follows the true altitude readings.

Lund University
Department of Automatic Control
Box 118
SE-221 00 Lund Sweden

Document name
MASTER'S THESIS

Date of issue
June 2019

Document Number
TFRT- 6083

Author(s)
Tomas Tamilinas
Chitranjan Singh

Supervisor
Mitra Pourabdollah, Volvo Car Group
Pontus Giselsson, Dept. of Automatic Control, Lund University, Sweden
Bo Bernhardsson, Dept. of Automatic Control, Lund University, Sweden (examiner)

Title and subtitle
Energy optimization tool for mild hybrid vehicles with thermal constraints

Abstract
The current global scenario is such where impact on the environment is becoming a rising concern. Global automotive manufacturers have focused more towards hybrid and electric vehicles as both more aware customers and governmental legislation have begun demanding higher emission standards. One of the many ways that Volvo Car Group approaches this trend is by mild hybridization which is by assisting the combustion engine by a small electric motor and a battery pack. A smart energy management strategy is needed in order to get the most out of the benefits that hybrid electric vehicles offer. The main objective of this strategy is to utilize the electrical energy on-board in such a manner that the overall efficiency of the hybrid powertrain becomes as high as possible. The current implementation is such that the decision for using the on-board battery is non-predictive. This results in a sub-optimal utilization of the hybrid powertrain. In this thesis, a predictive energy optimization tool is developed to maximize the utility of hybridization and the practical implementation of this tool is investigated. The optimization considers both the capacity as well as the thermal load constraints of the battery. The developed optimization tool uses information about the route ahead together with convex optimization to produce optimal reference trajectories of the battery states. These trajectories are used in a real-time controller to determine the battery use by controlling the adjoint states in the Equivalent Consumption Minimization Strategy equation. This optimization tool is validated and compared with the baseline controller in a simulation environment based on Simulink. When perfect information about the road ahead is known, the average reduction in fuel consumption is 0.99% relative the baseline controller. Several issues occurring in the real implementation are explored, such as the limited computational speed and the length of the route ahead that can be predicted. For this reason the information input to the optimization tool is segmented and the resulting performance is investigated. For a 30 second segmentation of the future route information, the average saving in fuel consumption is 0.13% relative to the baseline controller. It is shown that the main factor limiting the amount of savings in fuel consumption is the introduction of the thermal load constraints on the battery.

Keywords

Classification system and/or index terms (if any)

Supplementary bibliographical information

ISSN and key title
0280-5316

ISBN

Language
English

Number of pages
1-97

Recipient's notes

Security classification