PhD-FSTC-2015-63
The Faculty of Sciences, Technology and Communication

DISSERTATION

Defense held on 15/12/2015 in Luxembourg

to obtain the degree of

DOCTEUR DE L'UNIVERSITÉ DU LUXEMBOURG

EN SCIENCES DE L'INGÉNIEUR

by

Tim SCHWICKART
Born on 6 April 1985 in Hermeskeil, (Germany)

ENERGY-EFFICIENT DRIVER ASSISTANCE SYSTEM FOR ELECTRIC VEHICLES USING MODEL-PREDICTIVE CONTROL

Dissertation defense committee

Dr.-Ing. Holger VOOS, dissertation supervisor
Professor, Interdisciplinary Centre for Security, Reliability and Trust, Université du Luxembourg

Dr.-Ing. Jean-Régis HADJI-MINAGLOU, Chairman
Professor, Université du Luxembourg

Dr.-Sc. Dr.-Ing. Mohamed DAROUACH, Vice Chairman
Professor, Centre de Recherche en Automatique de Nancy, Université de Lorraine

Dr. rer. nat. Thomas ENGEL
Professor, Interdisciplinary Centre for Security, Reliability and Trust, Université du Luxembourg

Dr. Driss BOUTAT
Professor, INSA Centre Val de Loire

Declaration of Authorship

I, Tim Schwickart, declare that this thesis titled, 'Energy-Efficient Driver Assistance System for Electric Vehicles Using Model-Predictive Control' and the work presented in it are my own. I confirm that:

- This work was done wholly or mainly while in candidature for a research degree at this University.
- Where any part of this thesis has previously been submitted for a degree or any other qualification at this University or any other institution, this has been clearly stated.
- Where I have consulted the published work of others, this is always clearly attributed.
- Where I have quoted from the work of others, the source is always given. With the exception of such quotations, this thesis is entirely my own work.
- I have acknowledged all main sources of help.
- Where the thesis is based on work done by myself jointly with others, I have made clear exactly what was done by others and what I have contributed myself.

Signed: _____

Date: _____

Abstract

This thesis investigates a method to save energy and thus also extend the range of a series-production battery electric vehicle by influencing the driving style automatically with the help of a of a cruise controller. An exploration of existing methods shows that the contextual consideration of the current and upcoming driving situation is necessary to realise safe and energy-efficient driving. This limits the appropriate approaches to online methods using updated predictions of the vehicle behaviour. It turns out that the most suitable method for the intended purpose is model-predictive control (MPC).

The MPC generates controls for the accelerator pedal of the vehicle based on optimised predictions of the vehicle motion and energy consumption subject to the current and future road slope, curvature, speed limits and distance to an eventually preceding vehicle. The non-linear nature of the vehicle dynamics generally necessitates the use of a non-linear prediction model and solving a non-linear optimisation which goes along with difficulties in the online real-time implementation. However in this work - by exploiting and extending the tool sets of classical MPC - a controller based on a quadratic optimal control problem with linear constraints can be formulated that approximates the non-linearities of the plant dynamics with equivalent accuracy as a non-linear formulation.

A linear prediction model of the vehicle motion is derived by a change of the model domain from time to position and a change of variables to predict the kinetic energy of the moving vehicle instead of the driving speed. Further, a convex piecewise linear energy consumption model is included in the inequality constraints of the problem according to the methodology of separable programming to capture the consumption characteristics of the vehicle in different operating points. In this form, real-time capability and the energy-saving potential of the presented control approach can be demonstrated by simulations of the closed loop and by implementing the controller for driving experiments. A *Smart ED* series-production battery electric vehicle is chosen for the practical tests and all models and parameters are identified and adapted to the characteristics of the car. In this application case, a significant energy-saving potential could be demonstrated compared to human drivers.

To further reduce the computational burden and speed up the computation, the so-called *move blocking* method for input parameterisation of the MPC control trajectory is investigated and extended within this work to a *flexible move blocking* approach which enables a fast computation and at the same time high tracking performance.

Acknowledgements

First of all, I would like to thank my supervisor Prof. Dr.-Ing. Holger Voos for all the support provided throughout the thesis work period and his kind and pleasant supervisory style. For the same reasons, I would also like to thank the members of my CET committee Prof. Dr.-Ing. Jean-Régis Hadji-Minaglou and Prof. Dr.-Sc. Dr.-Ing. Mohamed Darouach.

Furthermore, I would like to thank my colleagues Amin Sajadi, Dr. Albert Rosich, Vladislav Gribov, Dr. Miguel Angel Olivares Mendez, Jan Dentler, Dr. Somasundar Kannan and the whole team of the Automation Research group for the fruitful discussions, their support regarding practical experiments and publications as well as the comfortable working atmosphere.

My special thanks also go to the technician Raphael Hinger who provided a lot of knowledge and support in the practical implementation of the control system and made my work a lot easier.

Moreover, I would like to thank the company *Delphi Automotive Systems Luxembourg S.A.* for the possibility of carrying out the measurements of the power train characteristics of the test vehicle, the *Centre de Formation pour Conducteurs* for providing their test track at the village Colmar Berg for the drive tests and the Automobile Club of Luxembourg *ACL* for offering their workshop.

Finally, I have to thank my parents and my friends for their support during the time of my P.h.D. studies without which the accomplishment of the work would not have been possible.

Contents

List of Figures

List of Tables

Abbreviations

ACC	**A**daptive **C**ruise **C**ontrol
ADAS	**A**dvanced **D**river **A**ssistance **S**ystem
CAN	**C**ontroller **A**rea **N**etwork
CC	**C**ruise **C**ontrol
DP	**D**ynamic **P**rogramming
ECU	**E**lectronic **C**ontrol **U**nit
EV	**E**lectric **V**ehicle
GP	**G**eometric **P**rogram
GPS	**G**lobal **P**ositioning **S**ystem
HMI	**H**uman **M**achine **I**nterface
LQR	**L**inear **Q**uadratic **R**egulator
MIMO	**M**ulti **I**nput **M**ulti **O**utput
MPC	**M**odel-**P**redictive **C**ontrol
NEDC	**N**ew **E**uropean **D**riving **C**ycle
NLP	**N**on-**L**inear **P**rogram
NMPC	**N**on-Linear **M**odel **P**redictive **C**ontrol
OBD	**O**n- **B**oard **D**iagnosis
OCP	**O**ptimal **C**ontrol **P**roblem
PC	**P**ersonal **C**omputer
PMP	**P**ontryagin's **M**aximum **P**rinciple
QP	**Q**uadratic **P**rogram
RHC	**R**eceding **H**orizon **C**ontrol
ROS	**R**obot **O**perating **S**ystem
SDP	**S**emi- **D**efinite **P**rogram
SISO	**S**ingle **I**nput **S**ingle **O**utput

SOCP	**S**econd- **O**rder **C**one **P**rogram
USB	**U**niversal- **S**erial **B**us

Symbols

A_c	System matrix of a continuous linear state space model	
A_d	System matrix of a discrete linear state space model	
A_{ineq}	Constraint coefficient matrix in linear inequality constraints	
A_v	Projected front surface of the vehicle	m^2
a_{dec}	Exponential decay factor of the host vehicle speed reference	
a_i	Constraint coefficient	
a_{inc}	Exponential increase factor of the host vehicle speed reference	
a_l	Lateral acceleration of the vehicle	m/s^2
a_s	Amplification factor on the sensitivities	
B_c	Input matrix of a continuous linear state space model	
B_d	Input matrix of a discrete linear state space model	
b_i	Constraint coefficient	
b_{ineq}	Constraint coefficient vector in linear inequality constraints	
C_{batt}	Charge supplied by the battery	As
C_{tot}	Total capacity of the battery	As
c_d	Air drag coefficient of the vehicle	
c_i	Constraint coefficient	
c_r	Rolling resistance coefficient of the tyres	
c_{rc}	Curve resistance coefficient of the tyres	
c^*_{rc}	Linearised curve resistance coefficient of the tyres	
$\hat{D}$	Discrete known disturbance sequence	
d_1	Known disturbance related to the sine of the slope	
d_2	Known disturbance related to the cosine of the slope	
d_h	Distance between the host vehicle and the preceding car	m
$d_{h,0}$	Initial distance between the host vehicle and the preceding car	m

$d_{h,safe}$	Safety distance between the host vehicle and the preceding car	m
E_{batt}	Electrical energy supplied by the battery	J
E_{ineq}	Constraint coefficient matrix in linear inequality constraints	
$E_{ineq,QP}$	Constraint coefficient matrix on the input sequence	
e_i	Mass gain factor to consider the rotational inertia of the drive train	
e_{kin}	Kinetic energy of the moving car	
F_c	Curve resistance force	N
F_d	Air drag resistance force	N
F_{gr}	Grade resistance force	N
F_{ineq}	Constraint coefficient matrix in linear inequality constraints	
$F_{ineq,QP}$	Constraint coefficient matrix on the system state sequence	
F_r	Rolling resistance force	N
F_{rc}	Curve resistance force	N
F_{trac}	Traction force at the wheel	N
$F_{trac,est}$	Heuristic estimate of the optimum traction force trajectory	N
$F_{trac,lim}$	Limit of the traction force value	N
f_{end}	End term of the cost function of an optimal control problem	
f_{eq}	Equality constraint term of an optimal control problem	
f_{ineq}	Inequality constraint term of an optimal control problem	
f_{int}	Integral term of the cost function of an optimal control problem	
G	State propagation matrix to be multiplied by the inputs	
g	Gravitational acceleration	
g_i	Constraint coefficient	
g_{ineq}	Constraint coefficient vector in linear inequality constraints	
$g_{ineq,QP}$	Constraint coefficient vector in a constraint sequence	
H	Weighting matrix on the squared variables in a quadratic program	
H_f	Hamiltonian function	
H_{QP}	Weighting matrix on the squared variables in a quadratic program	
h	Altitude level on the track	m
I_{batt}	Terminal battery current	A
i	Prediction step in an optimal control problem or counter variable	
i_{tot}	Total gear ratio of the transmissions of the vehicle	
J	Cost function of an optimal control problem	

J^o	Optimal cost function of an optisation problem	
k	Sample step in MPC control	
L_{QP}	Weighting matrix on the linear state tracking error sequence	
l	Counter variable	
$l_{e,b}$	Weighting on the terminal consumption in the basic cost	
$l_{e,s}$	Weighting on the terminal consumption in the stable cost	
$l_{e,z}$	Weighting on the terminal consumption in the zone cost	
M	Number of input blocks in an MPC with move blocking	
m	Number of inputs of a state space model	
m_{eq}	Equivalent vehicle mass	kg
m_i	Slope of a linear interpolating function	
m_l	Payload of the vehicle	kg
m_v	Curb mass of the vehicle	kg
N_c	Number of sample steps in the control horizon	
N_p	Number of sample steps in the prediction horizon	
n	Number of states or number of elements in a set	
O	Vector of offsets from the linear quadratic control law	
$\tilde{O}$	Reduced-order offsets (vector) from the LQR control law	
o	Offsets from the linear quadratic control law	
$\tilde{o}$	Reduced-order offsets from the linear quadratic control law	
P	Weighting matrix on the squared terminal system state	
P_0	State propagation matrix to be multiplied by the initial state	
P_i	Constraint function modelling the energy consumption	
P_{batt}	Electrical battery power	kW
P_{wheel}	Mechanical power at the wheels	kW
$\mathcal{P}_{fun}$	Set of constraint functions	
Q	Weighting matrix on the squared system states	
$\bar{Q}$	Weighting matrix on the terminal system states for stability	
Q_{QP}	Weighting matrix on the squared state tracking error sequence	
$q_{e,b}$	Weighting on the terminal tracking error in the basic cost	
$\bar{q}_{e,s}$	Weighting on the terminal tracking error in the stable cost	
$q_{s,b}$	Weighting on the accumulated tracking error in the basic cost	
$q_{s,m}$	Weighting on the accumulated tracking error with move blocking	

$q_{s,s}$	Weighting on the accumulated tracking error in the stable cost	
$q_{s,z}$	Weighting on the accumulated slack variables in the zone cost	
R	Weighting matrix on the squared inputs	
R_i	Internal ohmic resistance of the battery	Ω
R_{QP}	Weighting matrix on the squared input sequence	
r	Grid points	
r_c	Curve radius of the road	m
r_{max}	Maximum change rate of the speed reference per meter	$1/s$
$r_{s,s}$	Weighting on the input deviation in the stable cost	
r_{dyn}	Dynamic wheel radius of the tyres	m
S	Weighting matrix on the squared input moves	
SOC	Battery state of charge	
s	Position coordinate of the host vehicle	m
s_j	Sensitivities of the cost function	
s_l	Slack variables	
s_{GPS}	Position measurement provided by the GPS sensor	m
T	Blocking matrix in an MPC with move blocking	
T_{eng}	Torque at the shaft of the electric machine	Nm
t	Time coordinate	s
t_0	Start time of a prediction in an optimal control problem	s
t_{end}	End time of a prediction in an optimal control problem	s
t_{GPS}	Time passed since the last GPS measurement	s
U	Vector of control input variables	
$\hat{U}$	Sequence of discrete system inputs	
$\hat{U}^*$	Sequence of optimal discrete system inputs	
$\tilde{U}$	Reduced-order controls (vector) in move blocking	
U_{batt}	Terminal battery voltage	V
u	Input of a state space model	
$\tilde{u}$	Reduced-order control inputs in move blocking	
u_{cons}	Dummy variable for the energy consumption per meter	
u_{max}	Upper limit of the control input of a state space model	
u_{min}	Lower limit of the control input of a state space model	
V_i	Internal no-load voltage of the battery	V

v	Driving speed of the vehicle	m/s
v_{cur}	Admissible driving speed reference of the vehicle in curves	m/s
v_h	Driving speed of the host vehicle	m/s
v_{lim}	Legal speed limit	m/s
v_p	Driving speed of the preceding vehicle	m/s
v_{ref}	Driving speed reference	m/s
$v_{ref,smooth}$	Smoothened driving speed reference	m/s
W	State propagation matrix to be multiplied disturbance	
w	Vector of weightings in a quadratic program	
w_{QP}	Weighting on the linear term in a quadratic program	
$\hat{X}$	Sequence of discrete system states	
$\hat{X}_0$	Vector of initial states of the discrete system	
x	State of a state space model	
x_0	Initial state of a state space model	
y	Optimisation variables in a quadratic program	
$\hat{y}$	Data points of a discrete function	
α_{sl}	Road slope angle	deg
η_{batt}	Efficiency factor of the battery	
η_{ch}	Efficiency factor of on-board charger	
$\eta_{DC/AC}$	Efficiency factor of the power electronics	
η_{gear}	Efficiency factor of the gear box	
η_{mot}	Efficiency factor of the electric machine	
λ	Lagrange multiplier	
ρ_a	Density of the surrounding air	kg/m^3
ω_{eng}	Rotational speed of the shaft of the electric machine	rad/s

Chapter 1

Energy-Efficient Driving

The population of the world is growing. Along with this fact, the need for transportation is increasing and more and more people are using cars. At the same time, the amount of fossil fuels is limited and the emission of carbon dioxide is causing a global warming, that endangers the ecological balance in the world [20].

A means to reduce the emission of greenhouses gases in individual transport including the primary energy source are electric vehicles (EV's) [21, 22, 23], i. e. vehicles with a fully electric drive train. However, with the introduction of electric vehicles, new environmental and technical challenges arise.

Firstly, it is important that the production of electrical energy is environmentally friendly. Secondly, the production of electric vehicles needs to be environmentally friendly. For instance, the current battery-fed electric vehicles mostly use lithium-ion batteries. These batteries are dangerous in manufacturing [24], can be dangerous in crashes [25] and do not provide comparable energy storage densities compared to fossil fuels in terms of the weight-to-capacity ratio. In addition to the comparatively low capacity, the recharging time of batteries (in general $>$ 3 hrs.) prevents the recharging of EVs during a trip.

These problems lead to the fact that current electric vehicles have a very limited range and are not suitable for long distance (short-time) travels.

The main alternative to lithium-ion batteries are fuel-cells that generate electrical energy by oxidising hydrogen [26]. This technology has not fully matured yet and goes along

with a lot of challenges like storing the hydrogen or producing it on-board, crash test safety and the durability of the fuel cell itself in everyday usage [27].

Since soon breaking improvements in the on-board storage of electrical energy are questionable and the tank-to-wheel efficiency of electric vehicles is already very high (up to 85 %), the driving style seems to be worthwhile to decrease the energy consumption and increase the range of EVs.

Investigations for fuel-powered cars have shown that an energy-aware anticipatory driving style ("eco-driving", i. e. ecologic and economic driving) can save from 5 % [28] up to 20 % of energy [29, 30]. If similar numbers can be assumed for electric vehicles, the potential efficiency improvement in the driving style is much higher than the potential in the vehicle technology where the efficiency is already very high.

The driver can influence the amount of wasted energy by applying the mechanical brakes and by choosing the operating point of the vehicle drive train (mainly specified by the driving speed and the motor torque) and the driving resistance forces (e. g. the air drag resistance force depends on the driving speed). An energy-efficient driver avoids too much braking, operates the engine at a good efficiency and does not drive too fast.

The promising effects and the need to increase the range of EVs motivates to investigate this topic in this thesis and to develop a driver assistance system that helps to exploit this potential in electric mobility. Along with the aforementioned points that also apply for fuel-powered cars, electric vehicles have the ability to recover energy by regenerative braking which is a powerful tool to avoid wasting energy. Furthermore, most fully electric cars do not need a shift gearbox and thus have a fixed transmission ratio. These specifications will be investigated and considered in this work.

To achieve an energy-efficient driving behaviour, different approaches have already been investigated for fuel-powered cars and recently for electric vehicles as well. The approaches differ in the way the driving data are analysed and in the way the eco-driving input is provided to the driver or the car.

Apart from general hints for energy-efficient driving, there are personal trainings for drivers giving hints that can be incorporated in their driving style. Furthermore, electronic devices can help to analyse the driving and provide advice. Finally, the internal

control system of the car can include strategies for efficient driving and directly actuate the drive train in a manner of an intelligent cruise controller.

The realisation of a device to support energy-efficient driving requires position and / or speed measurement and recording. Additional measurements of the engine speed and the fuel flow can increase the quality and accuracy of the evaluation. The most sophisticated approaches include a preview of the upcoming speed limits, road slopes and curves. Devices that lend themselves to run tools for energy-efficient driving are smartphones, navigation systems and the on-board computers of the cars.

There are studies advocating eco-driving [31] and others that also highlight negative impacts on the macroscopic traffic flow in urban scenarios [32]. The contradictory results show that it is important to design an energy-efficient control system in a proper way and use it in the right situations.

The different methodologies - both for fuel-powered and electric vehicles - are reviewed in the following sections and subdivided in

- general hints and tips for efficient driving,
- systems that provide hints for driving and routing before the trip,
- systems that record data like the speed and position profile during the trip and provide an analysis to the driver after the trip,
- systems that analyse the driving situation and provide hints during the trip,
- sophisticated anticipatory eco-driving approaches using optimal control and
- the direct speed control of the car (cruise control) realised an energy-efficient way.

1.1 General Rules for Energy-Efficient Driving

Several governmental organisations, car manufacturers and other institutions have published general rules for energy-efficient [33, 34, 35, 36, 37]. The rules reflect general principles and target common flaws of many drivers. Most of the rule sets have been created for fuel-powered cars but mostly also apply for electric cars.

A collection of the most common eco-driving rules is given in the sequel with explanations of the physical principles they are based on.

- **Drive at low engine speeds (revolutions per minute), shift up early.**
  This hint refers to the fact that fuel-powered engine generally show a better efficiency at low engine speeds and high torques.

- **Maintain steady driving speeds.**
  This advice indicates that braking and accelerating wastes energy in many cases.

- **Decelerate smoothly.**
  A slow deceleration implies that the accelerator pedal is released earlier and thus the consumption of energy stops earlier. In consequence, energy is saved. For electric vehicles, energy can additionally be recovered in smooth decelerations.

- **Anticipate traffic flow.**
  By anticipating the traffic flow, unnecessary braking (and thus wasting of momentum) can be avoided.

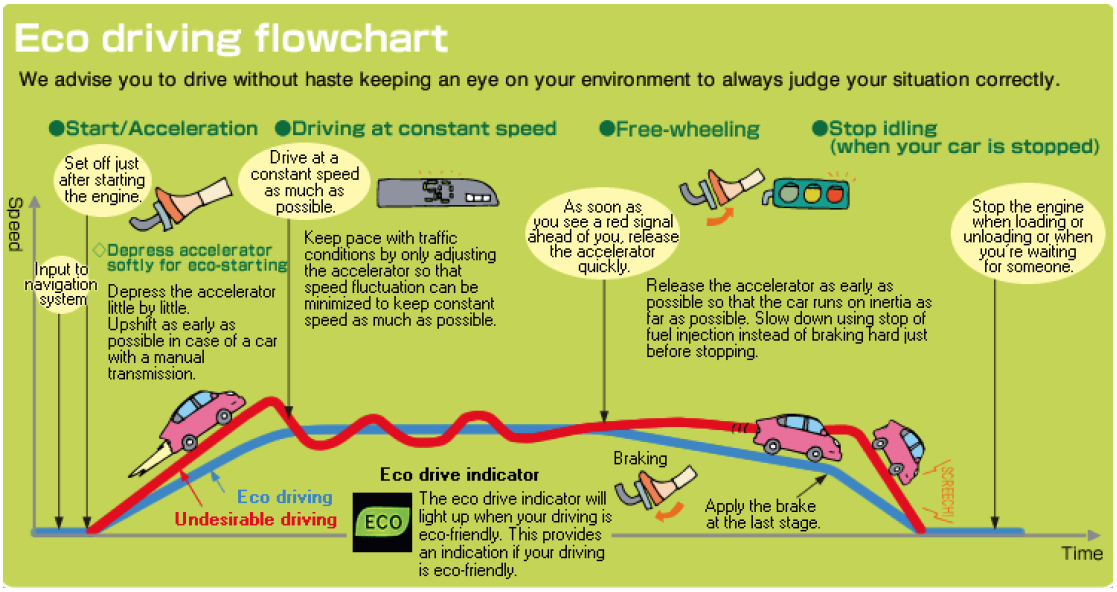
- **Turn off the engine at longer stops.**
  When using a fuel-powered car, the fuel consumption at idling can be avoided at longer stops like traffic lights or in traffic jams.

- **Avoid high cruising speeds.**
  The air drag resistance force increases with the square of the driving speed. Thus, higher speeds require progressively higher traction forces and thus energy expenditure. This applies for fuel-powered as well as electric vehicles.

- **Check tyre pressure regularly.**
  Low tyre pressure causes the tyres to deform more than usual when rolling. This increased churning causes higher rolling resistance forces, requires higher traction forces to maintain the speed and consequently higher motor torques going along with higher energy expenditure.

- **Use the air-conditioning sparingly.**
  The air conditioning system is generally based on a compressor driven by the engine. It represents a high additional load for the engine and causes a considerable additional energy consumption.

FIGURE 1.1: The basic rules of energy-efficient driving in an intuitive illustration [1].



- **Avoid dead weight.**

  A lot of people use to carry a lot of things in their car which they do not really need all the time. This additional weight requires higher traction forces to accelerate and consequently more energy.

- **Remove unused roof racks and close the windows at higher speeds.**

  These measures reduce the air drag resistance force of the vehicle during driving, especially at higher driving speeds.

Fig. 1.1 shows the basic rules in an illustrative way.

The presented hints can help drivers to adopt an economic and more efficient driving style. However, the hints are general and cannot be followed in any situation. For example, if a vehicle is preceding, steady driving speed can eventually not be ensured. Moreover, it is still up to the driver to accelerate, decelerate and drive in a way that he believes to be efficient. More sophisticated real-time methods are necessary to evaluate whether a driving style is really efficient in a particular situation.

1.2 Preparation for Energy-Efficient Driving before the Trip

Energy-efficient driving can already begin before the trip. For example, maintaining the vehicle and the correct tyre pressure, training of the driver and proper planning of the route contribute to a low energy consumption.

Especially truck and transport companies offer trainings to their professional drivers to improve their skills in energy-efficient driving [38, 39, 40]. The extremely high mileages and the high absolute fuel consumption of trucks make it especially worth to realise an efficient driving style. The hints are given in training sessions and the drivers try to incorporate them in their driving style afterwards. The fuel saving potential is expected to be around 5 % [28].

Planning an energy-efficient route based on the road type, the road slopes and the traffic situation is called "eco-routing" [41, 42]. One study has evaluated the energy-saving potential of this technique with up to 9 % for fuel-powered cars [43]. Another study including real-world tests with a series-production electric vehicle (*Nissan Leaf*) has even found up to 51 % of energy saving potential of an "eco-route" compared to the fastest route planned by a navigation system in a particular case [44].

Some navigations systems and on-board computers in cars provide eco-routing features, however with only basic functionality [3]. Considering road slopes is for example not yet considered in available systems. The energy-efficient routes are mostly planned based on the road type and traffic information.

The presented results demonstrate the importance of considering the energy consumption in planning and preparing a trip. However, in every-day travels like commuting to work, the freedom in planning different routes may be very limited and it is more important to influence the driving style, in that case.

1.3 Analysing the Driving Style after the Trip

Car manufacturers like *FIAT* provide a web portal to evaluate driving data recorded by a smartphone and compare it to other drivers to create incentives and a challenge

to improve the driving style [45]. The same idea is followed by other applications like [46, 47].

The analysis of the driving style after the trip has several drawbacks. First, all mentioned tools evaluating the recorded position, speed and acceleration data on the basis of heuristic rules such as high accelerations and decelerations should be avoided or steady cruising speed should be maintained. This evaluation does not consider the traffic situation the vehicle was in. Eventually, an emergency braking was necessary because of a preceding car and a recorded high-deceleration event was a perfect reaction of the driver in that particular situation. These contextual dependecies cannot be considered by the aforementioned approaches. Apart from this, most of these systems do not consider the engine characteristics of the specific car and cannot evaluate which cruising speed is really the most efficient.

An even more crucial drawback of analysing the driving style after the trip is that it requires a high motivation of the driver to spend time on learning a more efficient driving style. It can be expected that not many drivers are ready to undertake this effort. Gamification elements like creating a challenge among drivers cannot fully compensate this drawback.

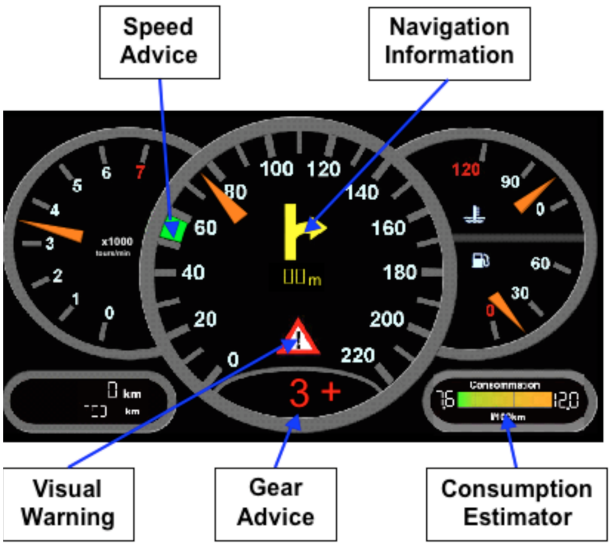
Finally, the feedback to the driver is very indirect with a long delay. The user can eventually not remember the driving situation to which the recorded events with positive or negative feedback are related.

1.4 Tools Providing Hints for Energy-Efficient Driving During the Trip

To overcome the problems with the delay of feedback in analysing the driving style after the trip and the motivation and extra time related to this, systems and tools have been developed that provide feedback to the driver in real-time during the trip. An example for a human-machine interface (HMI) giving hints regarding speed and gear is presented in Fig. 1.2.

This real-time advice is based on measurements of the driving speed, accelerations [48, 49] and eventually the actual fuel flow / energy consumption of the car if an interface to

FIGURE 1.2: A human-machine interface to provide recommendations regarding the preferable choice of speed and gear for energy-efficient driving [2].



the data bus of the vehicle is included [50, 51]. Some of the more sophisticated systems include data from the digital maps of a navigation system in the evaluation to realise an anticipatory behaviour.

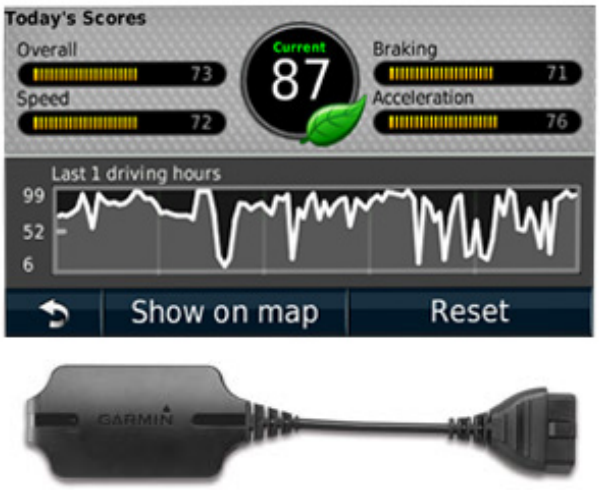
The emergence of smartphones with integrated GPS sensors and accelerometers has facilitated recording and processing speed and position profiles during the trip with no additional hardware equipment and has thus led to a lot of smartphone applications to evaluate and support energy-efficient driving like [52, 53]. There are also recent solutions especially for electric vehicles [54]. A solution with access to the internal sensor data of the vehicle through the CAN bus via a Bluetooth adapter is given in [55].

Other applications run on navigations systems based on the speed measurements through GPS with optional access to the data bus of the vehicle [3], see Fig. 1.3.

Vehicle manufacturers provide functionalities to give hints regarding the driving efficiency as well. An indication whether the gear should be changed as well as a display of the current consumption is included in almost all on-board computers of current fuel-powered cars with manual transmission. Some manufacturers, like for example *NISSAN* provide additional advice regarding the use of the accelerator pedal, optionally in combination with an analysis of the driving style after the trip [56] (cf. Section 1.3).

The aforementioned systems evaluate the actual driving speed and / or energy consumption to tell the driver if the current state of driving is energy-efficient or not and if the

FIGURE 1.3: Analysis of the speed profile by a navigation system providing a score to the driver using a bluetooth adapter to gain engine data via the on-board diagnosis (OBD) interface of the vehicle (bottom part of the picture) [3].

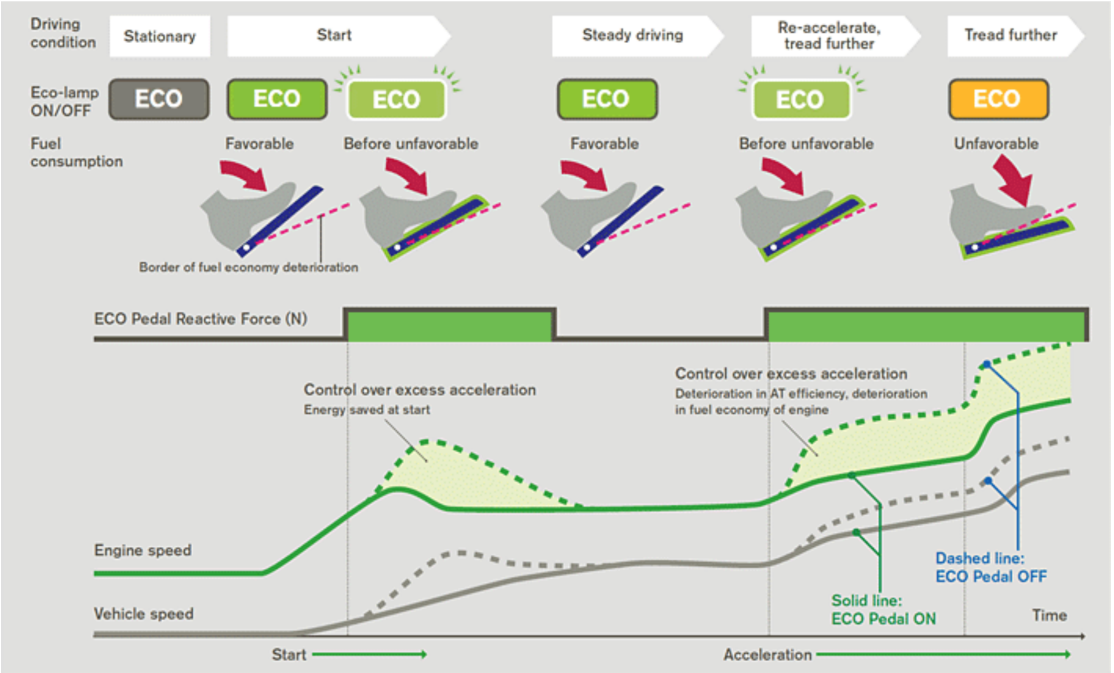


speed / acceleration should be reduced or the gear should be changed. Once more, this kind of evaluation does not consider the terrain and traffic context the vehicle is in, nor the engine characteristics. For example, gaining momentum (and thus accelerating) in front of a hill to overcome the slope afterwards with less effort requires at first a higher momentary consumption but can save energy in the end. Furthermore, driving at a high engine load can be efficient in fuel-powered cars, i. e. releasing the accelerator pedal is not always economic. As already mentioned, strong braking can be necessary in some situations.

A crucial drawback of systems that provide hints during the trip is the distraction of the driver. It is easy to imagine that if a system is constantly telling the driver to push the accelerator less or more needs a lot of attention and can lead to dangerous distractions. This problem can partially be solved by giving haptic force feedback through the accelerator pedal to tell the driver to push the accelerator less. This has e. g. been investigated and realised by *NISSAN* [4], see Fig. 1.4.

However, the drivers cannot perfectly follow the hints of the systems which leads to suboptimal behaviour. This drawbacks leads over to systems that directly take over the cruise control of the vehicle in an automatic way.

FIGURE 1.4: Working principle of the *Nissan Eco-Pedal* system with haptic feedback through the pedal to the driver [4].



1.5 Optimal Control Based Systems for Energy-Efficient Driving

The drawbacks of the aforementioned systems show that a sophisticated evaluation of the driving style should consider the terrain in front of the car (road slope and curvature), the traffic situation (speed limits, traffic lights, preceding cars, ...) and the characteristics of the specific host vehicle.

A methodology that can consider all these aspects is optimal control. It is a predictive strategy that determines the trajectory of the control inputs such that a cost functional defining the control objectives is minimised. A preliminary work defining energy-efficient driving of fuel-powered cars as optimal control problem (OCP) is [57]; a recent presentation of the topic is given in [58].

A cost or objective function is defined containing weighted sums of terms related to the energy consumption of the vehicle and the driving speed. The relation between these terms is given by dynamic prediction model taking the characteristics of the vehicle into account. The inputs of the model are the motor torque / brake torque (or the traction

force at the wheel) and optionally the gear which are determined by minimising the cost function.

The road slope and the road curvature are extracted from digital maps providing an electronic terrain preview horizon and considered as a measurable influence (i. e. disturbance) on the system. Constraints on the system are for example vehicle limitations like the maximum torque the motor can provide, the maximum lateral acceleration in curves or legal aspects like the speed limits.

Minimising the energy consumption and the travel time are in most cases contradictory targets since going faster generally requires more energy. As a consequence, the minimisation of a weighted sum of the two objectives leads to a compromise according to the weightings.

According to this, optimal control is a promising approach but not straightforward to implement. The associated dynamic optimisation problem is not easy to solve in real-time. The vehicle motion and energy consumption equations are generally non-linear which leads to a non-linear dynamic optimisation. Gear shifting poses a special challenge in the solution of the OCP since it requires discrete decision variables, leading to a mixed-integer non-linear problem. The application of energy-efficient driving based on optimal control in real-time is safety critical and requires short sample times (around 0.3 s, the reaction time of a human). These challenges have often prohibited the application of the eco-driving approaches in practical tests.

The solution of such an optimal control problem can be obtained in different ways. On the one hand, there are analytical solutions based on the *Hamiltonian* function and *Pontryagins Maximum Principle* (PMP) [59]. This is called *indirect solution*.

Energy-efficient driving as optimal control problem with an analytical solution has been investigated in [57, 60, 61, 62, 63] and for electric vehicles in [64, 65]. However, analytical solutions can only be derived for simple constraints on the control inputs (and no constraints on the states) and allow less freedom in the definition of the optimal control problem. This drawback leads to the fact, that the vehicle and the traffic cannot be modeled in detail and some physical effects need to be neglected.

Another way of solving an OCP is *dynamic programming* (DP) [66]. Here, the state and and input space are discretised in the form of a network and the optimal path through

the network (here, the optimal input trajectory of the optimal control problem) is sought after systematically from the end of the trajectory to the beginning. DP can solve any kind of OCP. The drawback is the high computation time which grows exponentially with the dimension of the problem (*"the curse of dimensionality."* [66]). In most cases, this disadvantage prohibits the use of DP algorithms for an energy-efficient cruise control in real-time [67, 68, 69].

Thus, the works investigating DP for eco-driving often focus on theoretical studies [70] or target an optimisation of the speed profile of the complete route offline before the trip. The resulting reference speed is then tracked online during driving. This has been investigated for fuel-powered [71, 72] and electric vehicles [73, 74, 75].

Computing an optimised speed profile before the trip is only of limited use. The traffic conditions cannot be considered in advance and thus, preceding cars may prevent following the precomputed reference speed. Then, the optimisation result is not realisable and thus not meaningful.

However, with some modifications, DP has also been used in real-time application for energy-efficient driving of fuel-powered vehicles [76, 77]. Other approaches try to find heuristic solutions that approximate the optimal DP solution and enable real-time implementation. [78, 79] follow this methodology to find an eco-driving policy for electric vehicles.

Another option to solve the optimal control problem related to energy-efficient driving are direct discretisation approaches. Here the problem of finding an optimal trajectory is converted into a finite dimensional optimisation problem by discretising the differential equations directly in the first step. For the solution of the (in most cases non-linear) finite dimensional optimisation problem there exist several fast solving methods like *active-set* or *interior-point* strategies which eventually enable a solution in real-time.

A driving simulator study of an energy-efficient cruise control for fuel-powered cars based on non-linear model-predictive control considering slopes, preceding vehicles and traffic lights is given in [12, 80, 81, 82, 83, 84, 85]. However, no practical implementation is presented.

Simulations of a real-time capable formulation for a fuel-powered car including gear-shifting is proposed in [86]. A simplified model-predictive control for energy-efficient driving with quadratic cost function and linear constraints is proposed in [87, 88].

A method based on rounding strategies to deal with mixed-integer non-linear model-predictive control in real-time with an exemplary application of eco-cruise control for a truck with 16 gears (representing the discrete decision variables) is presented in [89]. Before, this topic has been addressed by using a systematic search strategy in combination with fast non-linear optimisation in [90, 91].

It can be concluded that optimal control is a sophisticated method to find an energy-efficient driving strategy because it is able to consider the vehicle characteristics and limitations, to take anticipatory action and find the best possible control with respect to a cost function. However, it is difficult and challenging to apply this method in real-time which requires a problem formulation and solution strategy that is especially tailored to the problem. The difficulty of the real-time implementation has again been pointed out recently in [92]. Further, it is practically hardly possible to take an optimisation of the whole trip into account at once / in one step.

This limits the useful methods for practical implementation to those with updated predictions over a finite horizon, namely *Model-Predictive Control* (MPC) in a receding horizon fashion.

1.6 Automatic Control of the Longitudinal Vehicle Dynamics in Series-Production Vehicles

In the preceding sections, it has become clear that it is advantageous to realise an energy-efficient driving assistance system as an automatic cruise control to avoid distraction of the driver and achieve higher saving benefits.

Automatic cruise control was introduced in commercially available vehicles in 1958 by Chrysler. Conventional systems keep a constant driving speed set by the driver despite disturbances like varying road slopes. This is realised with a control loop using a measurement of the driving speed and influencing the motor torque by actuating the

accelerator. As soon as the accelerator or brake pedal is touched by the driver, the automatic cruise control is deactivated.

More advanced systems like the *Adaptive Cruise Control* (ACC) were developed in the 1990's to keep a constant distance to a preceding vehicle despite a varying cruising speed. This has been realised by measuring the distance to the preceding vehicle with the help of radar sensors. If there is no slower preceding vehicle, the constant set-speed is tracked. The first system on the market was *Distronic* introduced by Daimler [93] in 1998. Other manufactures followed in the ensuing years.

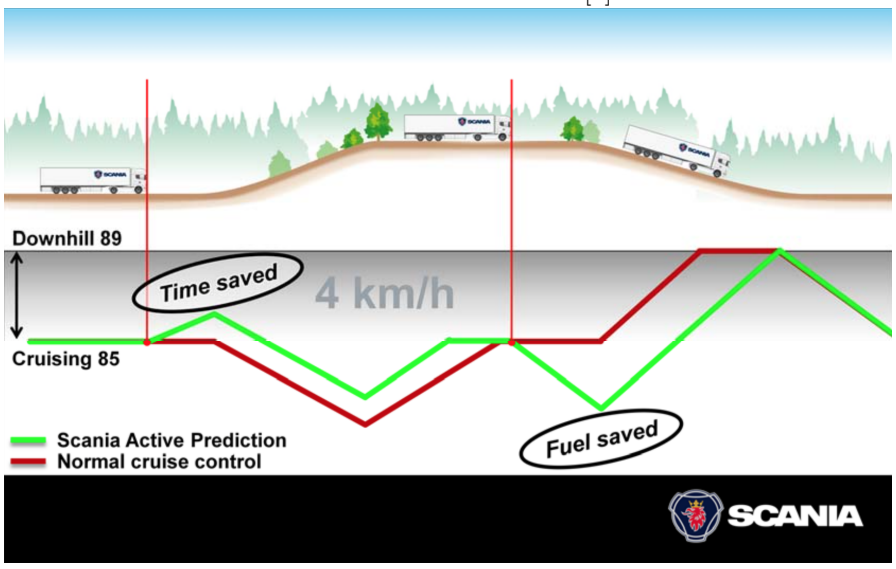
At first, the energy consumption of the vehicle was not considered by these systems and only safety and comfort were in focus. Then, energy efficiency was considered indirectly by avoiding high accelerations and decelerating smoothly.

While available ACC systems were based on classic controllers, model-predictive control with the application to a cruise controller with conflicting objectives related to safety, comfort and energy efficiency has been considered in research [94, 95, 96, 97, 98].

Recently, with the introduction of more and more advanced driver assistance systems, energy-efficient predictive cruise control has been introduced in commercially available fuel-powered vehicles. Daimler [99] and Scania [5] have introduced such a system for heavy trucks where the fuel consumption plays a key role, Porsche has realised a first system for a passenger car that promises up to 10 % fuel savings [100, 101]. These systems use a terrain preview including the road slope and curvature together with a measurement of the distance to preceding cars. Fig. 1.5 exemplarily shows the working principle of the *Scania Active Prediction System*.

For electric vehicles, there is no such system available. However, optimisation-based control is expected to play a key-role in electrified vehicle power trains [102]. Anyway, in the use case of a small electric vehicle, the power consumption of the electronic control units (ECUs) themselves is more crucial than in the case of a heavy truck. Thus, in the case of a small EV, it is important to find a computationally efficient way to implement the predictive cruise controller in real-time without necessitating a very powerful on-board computer that consumes a lot of energy.

FIGURE 1.5: Working principle of the *Scania Active Prediction* system. In the illustrated example, the truck accelerates before a hill to gain momentum and to save time. Then it overcomes the up-slope in a higher gear. During the down-slope, it uses no tractive effort to save fuel [5].



1.7 Conclusion of the Chapter

As a result of this chapter, it can be concluded that an efficient driving style can reduce the energy consumption of a vehicle considerably. Every driver can learn some basic techniques and try to implement eco-driving rules to save energy. An evaluation of the driving style after the trip with gamification elements can support this effort if the driver is motivated to spend his spare time to learn energy-efficient driving.

However, to determine which driving behaviour is the most efficient, simple rules are not sufficient and the specific driving situation must be evaluated with respect to various criteria. The characteristics of the vehicle, the current operating point, the upcoming speed limits, road slope and curvature as well as the traffic on the road influence the best driving strategy.

This evaluation is only possible with optimisation based approaches working with a dynamic prediction model of the vehicle, namely optimal control. Since the traffic situation (mainly preceding cars) should be considered, the optimisation must be updated during driving. Approaches that rely on finding an optimised speed profile in advance before the trip cannot consider the traffic situation and are thus not a suitable approach for energy-efficient driving.

An energy-efficient cruise control in real-time is challenging since the underlying optimisation problem is generally non-linear and short sample times are required. Here, fast solving methods are required together with the restriction to a finite prediction horizon. This leads over to fast model-predictive control.

Moreover, a technical system to support an energy-saving driving style should not only provide hints to the driver. The driver cannot perfectly follow the hints from the system and it may lead to dangerous distractions if the driver focuses on the eco-advice all the time. Thus, an automatic cruise control is preferable.

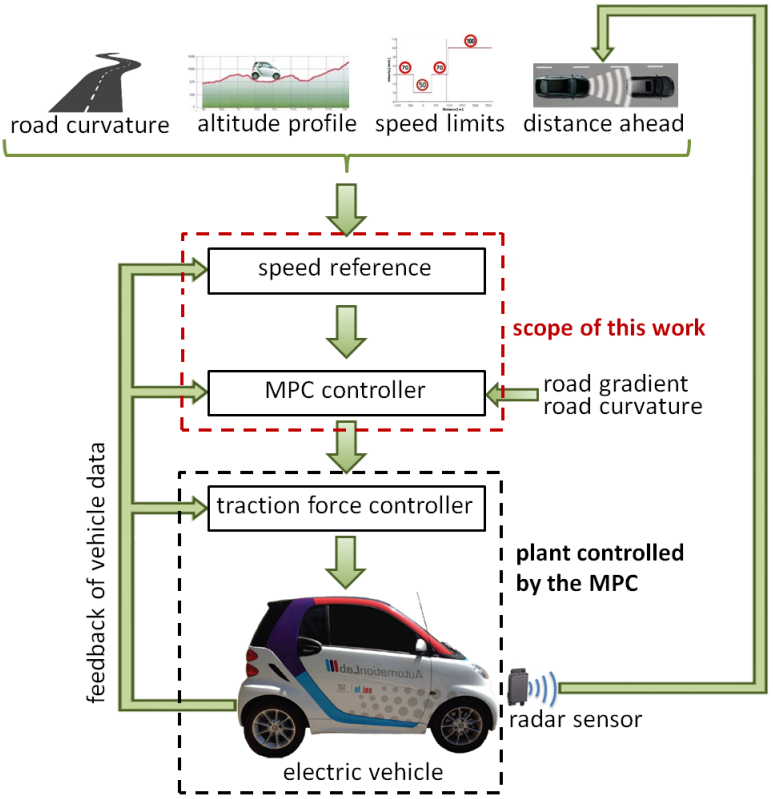
In summary, a sophisticated method for energy-efficient driving should have the following properties:

- The characteristics of the specific host vehicle must be considered.
- The system must be predictive to consider the upcoming road and traffic conditions.
- The predictions must be updated online during driving, thus the system must be implementable in real-time.
- The actions of the system must be realised automatically to avoid distraction of the driver and to achieve a better tracking.

Predictive optimal receding horizon control fulfills all these requirements and is thus chosen in this work to develop an energy-efficient cruise controller especially for electric vehicles. More specifically, optimised traction force values are computed with the help of an MPC controller using predictive information on the upcoming road slope, curvature, speed limits and the distance to a preceding car to realise anticipatory driving. The controller shall be used as a high-level controller while the actual control of the electric machine in the car is taken over by the control system of the vehicle. The energy-efficient cruise control is realised by automatically actuating the accelerator pedal which allows slight regenerative braking when releasing the pedal. The overall system setup is illustrated in Fig. 1.6.

Herein, a special focus is put on fast methods to achieve real-time capability in model-predictive control. To prove the practical implementability and the real-time capability,

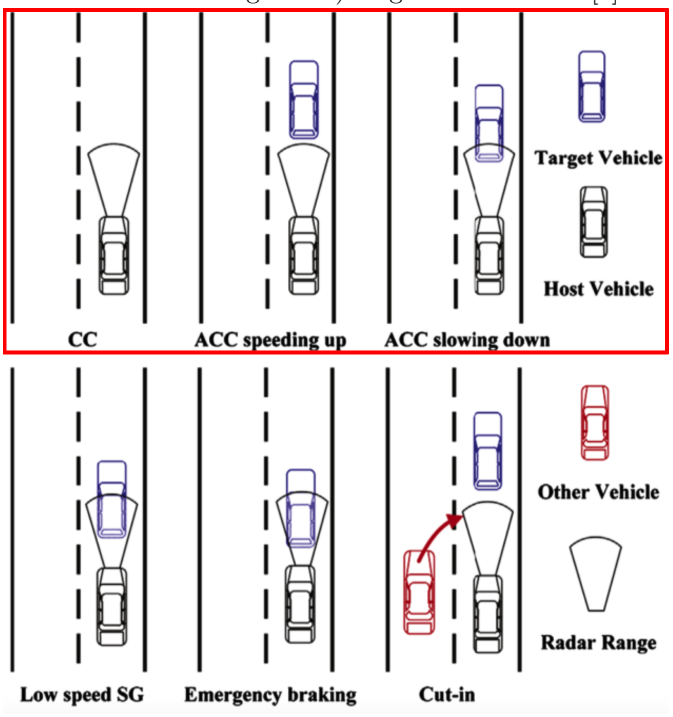
FIGURE 1.6: Structure of the energy-efficient cruise control system investigated in this work. The main focus is put on the high level MPC controller to determine optimised traction force values as a set point for the lower level control system of the vehicle.



the developed control system is implemented experimentally in a real series-production electric vehicle as a test carrier, namely a *Smart Electric Drive* (3^{rd} generation, model 2012). Drive tests are conducted to investigate the energy saving potential of the approach.

The system must be able to handle classical cruise control tasks (driving at constant speed with free headway), as well decelerating and accelerating appropriately according to speed limits and preceding cars. However, it is not meant for stop-and-go driving or emergency braking. The different cases to be considered are illustrated within the red box in Fig. 1.7.

FIGURE 1.7: Different driving situations to be handled by different driver assistance systems. The eco-cruise control presented in this work must be able to handle the three cases in the red box in the upper half of the Figure (Cruise Control, ACC Speeding Up and ACC Slowing Down). Figure edited from [6]



Chapter 2

Model-Predictive Control

As a result of the findings in the literature review in Chapter 1, the most promising method to realise a driver-assistance system to support energy-efficient driving is optimal predictive control and more precisely, model-predictive control (MPC).

One requirement of such a control system is the real-time capability. This means that the control action must be computed and realised within a predefined finite time limit. This in turn requires a bounded computational effort to solve the MPC problem.

General MPC formulations as well as methods to speed up MPC by reducing the computational complexity are presented. Different MPC formulations are discussed since different ways to formulate the problem and to obtain a solution can be considered. Depending on the control task and system model, one approach can be favourable over the other and choosing an advantageous method can lead to a more favourable solution in terms of the necessary computational effort and implementational cost.

2.1 Different Formulations of Model-Predictive Control

In model-predictive control, the control action is determined by optimising predictions of the system behaviour based on a dynamic model of the plant. In many cases, state space models with states x and inputs u as a function of the time t are used:

$$\frac{dx}{dt} = f(x, u) \tag{2.1}$$

The control input as well as the states of the system can be subject to inequality and / or equality constraints. The objective of the optimisation is given by a cost function. The way the cost function, the system model and the constraints are formulated, determines the nature of the resulting optimal control problem.

The fact that constraints on the control and the states can be considered and non-linear models are possible, makes the model-predictive control approach more powerful than the classical linear quadratic regulator obtained by solving the Riccati equation.

To provide feedback, the optimal control problem is solved repetitively with an updated initial state provided by the measurement. At every update step, only the very first part of the solution is fed to the plant. Then, the optimisation is repeated with updated measurement and the prediction horizon window shifted into the future and the control action is updated as well. This principle is called receding horizon control (RHC) and illustrated in Fig. 2.1.

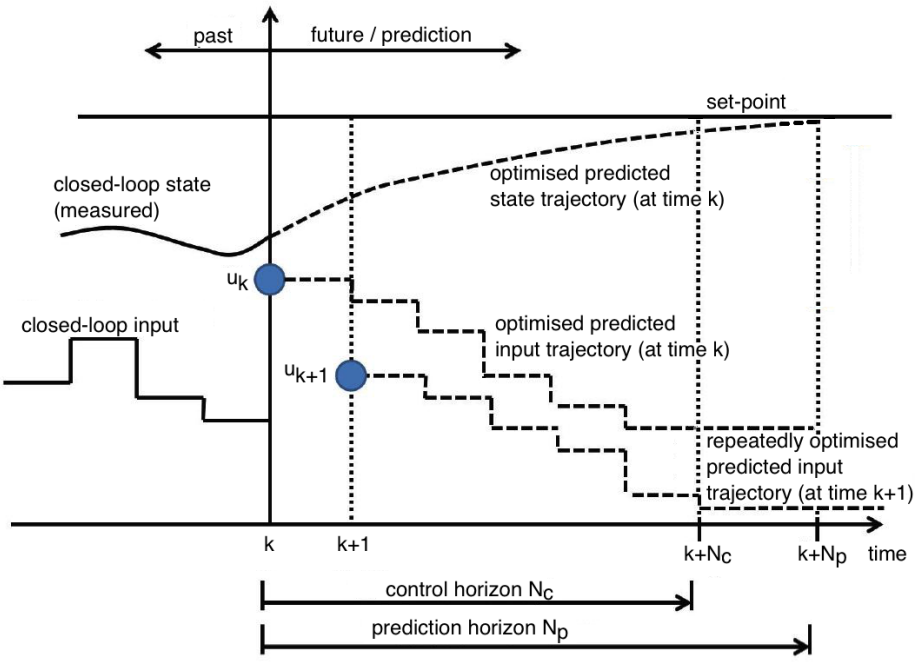
2.1.1 Continuous-Time Optimal Control

Generally, the objective function, the dynamic system model and the constraints of the underlying optimal control problem can be non-linear. Continuous-time optimal control problems often take the following form:

$$\begin{aligned} \min_{u(t)} \quad & f_{end}\big(x(t_{end}), t_{end}\big) + \int_{t_0}^{t_{end}} f_{int}\big(x(t), u(t), t\big) \cdot dt \\ \text{subject to the equality constraints (incl. the dynamic plant model):} \quad & \\ & f_{eq}\big(x(t), u(t), t\big) = 0 \end{aligned} \tag{2.2}$$

The system model relates the state trajectory $x(t)$ to the input trajectory $u(t)$. It is sought after a function $u(t)$ starting at t_0 and ending at the time instant t_{end} that

FIGURE 2.1: Basic principle of a discrete-time MPC. The left part of the picture shows the past with the measured state trajectory and the inputs that have been fed to the real plant. The right side of the picture shows the optimal predicted input at time k and the repeated optimal prediction at time $(k+1)$. Figure edited from [7].



minimises the objective. Since a function is sought after and not a finite number of variables, problem (2.2) is an infinite dimensional or *dynamic* optimisation problem.

The classical approach to solve such a dynamic optimisation problem analytically is based on the *calculus of variations* according to the theories of Euler and Lagrange [103]. By introducing *Lagrange multipliers* $\lambda(t)$, the equality constraints can be inserted into the objective function to convert problem (2.2) into an unconstrained problem [104].

$$\min_{u(t)} \quad f_{end}\big(x(t_{end}), t_{end}\big) + \int_{t_0}^{t_{end}} \Big(f_{int}\big(x(t), u(t), t\big) + \lambda^T(t) \cdot f_{eq}\big(x(t), u(t), t\big) \Big) \cdot dt \tag{2.3}$$

The terms of problem (2.3) can be summarised by the *Hamiltonian function* as follows [104]:

$$H_f\big(x(t), u(t), \lambda(t), t\big) = -f_{int}(x(t), u(t), t) + \lambda^T(t) \cdot f_{eq}\big(x(t), u(t), t\big) \tag{2.4}$$

Then, necessary optimality conditions for the optimal control problem can be derived as follows [104]:

$$\begin{aligned} \frac{\partial H_f}{\partial u} &= 0 \\ \dot{\lambda} &= -\frac{\partial H_f}{\partial x} \end{aligned} \tag{2.5}$$

Additional optimality conditions result from the type of end conditions on the state trajectory [104]. In case the optimality conditions cannot be solved analytically, a discretised numerical solution to this boundary value problem must be aimed at.

If limits on the input trajectory in the form $u_{min} \leq u(t) \leq u_{max}$ are present, a solution can be obtained with the help of *Pontryagin's Maximum Principle* (PMP). Then, the condition $\frac{\partial H_f}{\partial u} = 0$ must be replaced by the following to find the associated optimal control law [104].

$$H_f\big(x(t), u(t), \lambda(t), t\big) = \max_u \tag{2.6}$$

However, for more general inequality constraints on $u(t)$ or for inequality constraints on the states $x(t)$, it is very hard to find analytical solutions using PMP. Then, a discretisation of the optimal control problem and a numerical solution is necessary.

2.1.2 Non-Linear MPC with Discretised System Model, Objective Function and Constraints

For problems that cannot be solved analytically, the controls of problem (2.2) are discretised on a suitable time grid within a prediction horizon of N_p samples to reach to a finite dimensional optimsation problem which can be solved with numerical methods.

The resulting discrete problem formulation is updated and solved at every control sample time step k.

$$\begin{aligned} \min_{u(i)} \quad & f_{end}\big(x(k+Np|k)\big) + \sum_{i=0}^{N_p-1} f_{int}\big(x(k+i|k), u(k+i|k)\big) \\ & \text{subject to the equality constraints:} \\ & f_{eq}\big(x(k+i|k), u(k+i|k)\big) = 0 \quad \forall i \in \{0, 1, ..., N_p - 1\} \\ & \text{subject to the inequality constraints:} \\ & f_{ineq}\big(x(k+i|k), u(k+i|k)\big) \leq 0 \quad \forall i \in \{0, 1, ..., N_p - 1\} \end{aligned} \tag{2.7}$$

A finite dimensional problem can be obtained and solved by numerical discretisation methods like *multiple shooting* [105] or *collocation methods* [106] in combination with fast optimisation solvers like *sequential quadratic programming* or *interior point methods* which eventually allow to implement a non-linear model-predictive control system in real-time.

If some of the control and / or the state variables are constrained to take discrete values, (2.7) turns into a hybrid problem (partially combinatorial and partially continuous) which can be very hard or impossible to solve in real-time. In vehicular applications, such a hybrid problem can arise because of gear-shifting where the transmission ratio can only take discrete values. In this case, search strategies like *branch and bound* and *branch and cut* or rounding strategies can be applied [89].

A general method to solve all kinds of discretised optimal control problems is *dynamic programming* [66]. However, the method is in most cases too slow for real-time implementation.

2.1.3 Model-Predictive Control Based on Convex Optimal Control Problems

In the case where problem (2.7) cannot be solved in real-time, a convex problem formulation can be aimed at. A convex optimisation is performed over a convex feasible set. A real function is convex, if its graph lies below each segment between two arbitrary

points of the function graph. The set of points above the graph (the epigraph of the function) is convex in this case.

A convex feasible set is preferable to non-convex formulations since it is guaranteed that only one global extremum exists, which is an important property to guarantee the convergence and solvability of the optimisation algorithm in practical implementations.

Possible convex formulations are *second-order cone programs* (SOCPs), *geometric programs* (GP) and *semidefinite programs* (SDP) [107]. For implementational reasons, it is worth checking if a non-linear optimal control problem can be formulated in a strictly convex way.

2.1.4 MPC with Discrete Quadratic Objective Function and Linear Constraints

A very commonly used special case of potentially convex discrete problem formulations is a quadratic cost function combined with linear constraints [13]. The states $x(k+i|k)$ and the controls $u(k+i|k)$ are discretised on a suitable grid within a prediction horizon of N_p samples. The cost function penalises the quadratic deviations of the states from the reference trajectory $x_{ref}(k+i|k)$ and the weighted squared values of the controls $u(k+i|k)$. The state trajectory is given by a discrete linear state space model and states and controls are subject to linear inequalities.

$$\begin{aligned}
&\min_{u(i)} \quad ||x(k+N_p|k) - x_{ref}(k+i|k)||_P^2 \\
&+ \sum_{i=0}^{N_p-1} ||x(k+i|k) - x_{ref}(k+i|k)||_Q^2 + ||u(k+i|k) - u_{ref}(k+i|k)||_R^2 \\
&\text{subject to the linear state space plant model:} \\
&x(k+i+1|k) = A_d \cdot x(k+i|k) + B_d \cdot u(k+i|k) \quad \forall i \in \{0,1,...,N_p-1\} \\
&\text{subject to the initial conditions:} \\
&x(k|k) = x_{0|k} \\
&\text{subject to the inequality constraints:} \\
&F_{ineq} \cdot x(k+i|k) + E_{ineq} \cdot u(k+i|k) \leq g_{ineq} \quad \forall i \in \{0,1,...,N_p-1\}
\end{aligned} \tag{2.8}$$

This formulation yields several advantages and is widely used. Linear state space models are widely used in control applications and can straightforwardly be implemented in the above OCP formulation. The limitations of the actuators and the plant can be incorporated by linear inequalities.

Additionally, problem (2.8) can be reformulated as a *quadratic program* (QP) where y are the optimisation variables.

$$\begin{aligned} \min_{y} \quad & \frac{1}{2} y^T H y + w^T y \\ & \text{subject to:} \\ & A_{ineq} \cdot y \leq b_{ineq} \end{aligned} \tag{2.9}$$

The fast and efficient solution of QPs has been studied extensively such that real-time capability is given in most cases by the use of modern computer hardware. Furthermore, if the weighting matrices fulfill the properties $Q \geq 0$ and $R > 0$ and $P > 0$, problems (2.8) and (2.9) are convex and the solvability in polynomial time is guaranteed [108].

Even though reference tracking of a linear state space model can also be considered as an even faster to solve linear program, the quadratic formulation of the cost function often leads to a smoother and more desirable control behaviour [109].

Finally, stability of the closed-loop control can be guaranteed by choosing a suitable length of the prediction horizon and by using a proper weighting on the terminal state $x(k + N_p|k)$ or a terminal constraint set [13, 110].

In summary, all these properties make the discrete quadratic formulation a very advantageous formulation for practical use. However, non-linear plant dynamics cannot straightforwardly be considered in (2.8).

2.1.5 Explicit Model-Predictive Control

To facilitate the implementation of model-predictive control, the off-line solution of optimal control problems as a function of the parameters that can change during the control operation has been investigated. These parameters are mainly the initial state, the reference and eventually a known disturbance.

If an explicit solution can be obtained off-line, the precomputed optimal controls can be read from a lookup table during operation without the need to perform an optimisation on-line. This can reduce the computational burden and facilitate the implementation of the MPC.

For quadratic optimal control problems with linear constraints in the form (2.8) with constant reference, it has been found that the explicit solution takes the form of a continuous piecewise linear function of the actual state of the plant. According to this, the optimal control can be expressed as a gain factor depending on the partition of the state space the current initial state is in. [111, 112]

The main drawback of explicit MPC is that the number of partitions grows rapidly with the number of steps in the prediction horizon and the dimension of the problem. This makes the computation of the piecewise affine control law time consuming and the high-dimensional lookup table operation can be more challenging to perform than the online solution of the quadratic program linked with the MPC task [113]. For this reason, the application of explicit MPC is limited to control tasks with a small model dimension and few prediction steps.

The implementation of explicit MPC can be realised with the help of the *Multiparametric Toolbox* [114].

2.2 Methods to Speed up Discretised Real-Time Model-Predictive Control

In addition to the choice of the problem formulation, the optimisation can further be simplified. Most methods to speed-up the application of discretised MPC focus on a simplification of the underlying optimisation. The most common approaches try to reduce the number of optimisation variables (i. e. the dimension of the problem) by parameterising the input and / or the state trajectory of the system in different ways.

In parallel to these techniques affecting the formulation of the optimal control problem, mathematicians work on how to solve a given optimisation as efficient and fast as possible. In this overview however, only the approaches that concern the problem formulation are considered.

2.2.1 The Use of a Control Horizon Shorter than the Prediction Horizon

In the previous part of the chapter, it has implicitly been assumed that optimised controls are determined for each of the N_p sample steps of the prediction.

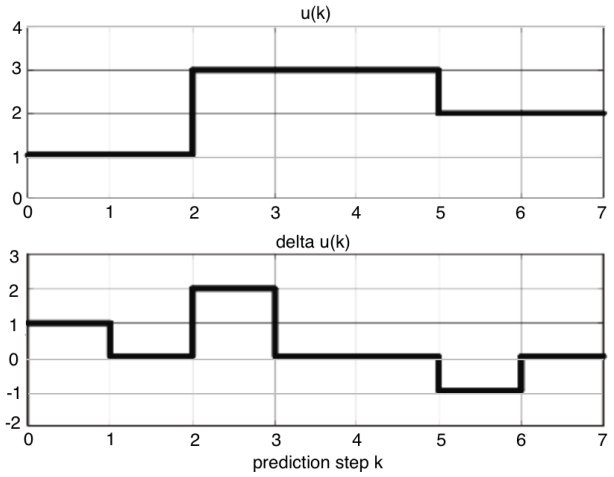
To reduce the number of decision variables in the optimal control problem, it is a common technique to limit variations in the controls to the first N_c samples of the prediction horizon (called the *control horizon*) and keep their values constant for the remainder of the prediction, cf. Fig. 2.1. The controls at the beginning of the prediction horizon are the most / the only important in MPC since only the controls of the very first sample are fed to the real plant.

For the case of a quadratic cost function and linear constraints, problem (2.8) is modified as follows [13]:

$$\begin{aligned}
\min_{u(i)} \quad & ||x(k+N_p|k)||_P^2 + \sum_{i=0}^{N_p-1} ||x(k+i|k) - x_{ref}(k+i|k)||_Q^2 \\
& + \sum_{i=0}^{N_c-1} ||u(k+i|k) - u_{ref}(k+i|k)||_R^2 \\
& \text{subject to the linear state space plant model:} \\
x(k+1) = A_d \cdot x(k+i|k) + B_d \cdot u(k+i|k) \quad & \forall i \in \{0,1,...,N_p-1\} \\
& \text{subject to the equality constraints:} \\
x(k|k) = x_{0|k} & \\
u(k+i|k) = u(k+N_c|k) \quad & \forall i \in \{N_c+1, N_c+2, ..., N_p-1\} \\
& \text{subject to the inequality constraints:} \\
F_{ineq} \cdot x(k+i|k) + E_{ineq} \cdot u(k+i|k) \leq g_{ineq} \quad & \forall i \in \{0,1,...,N_p-1\}
\end{aligned} \tag{2.10}$$

This technique leads to a certain deterioration of the control performance but also to a faster online optimisation since the problem dimension is shrunk to $N_c < N_p$ decision variables for each control input.

FIGURE 2.2: Control trajectory $u(k)$ (top) and the change in the control $\Delta u(k)$ (bottom) in the case of input move blocking, edited from [8].



2.2.2 Input Move Blocking

Another method to reduce the number of decision variables is *input move blocking* where the values of the control are fixed (i. e. kept constant) for several prediction sample steps [115]. This is illustrated in Fig. 2.2. Input move blocking was first proposed in [116] and extensively used in industry [117].

Input move blocking can be incorporated in the optimal control problem by adding an equality constraint fixing the values of the full length vector of controls

$$U = [u(0|k), u(1|k), \ldots, u(N_p - 1|k)]^T$$

to those of the reduced order controls with M degrees of freedom

$$\tilde{U} = [\tilde{u}(0|k), \tilde{u}(1|k), \ldots, \tilde{u}(M - 1|k)]^T \text{ with } M < N_p.$$

The two vectors U and $\tilde{U}$ are linked by the blocking matrix T [118]. For single-input systems, the blocking constraints take the form:

$$U = T \cdot \tilde{U} \tag{2.11}$$

In the case of a multi-input system, the following equation using the Kronecker product of T with the identity matrix is applied, where m is the number of control inputs:

$$U = \big(T \otimes I_{m\times m}\big)\tilde{U} \tag{2.12}$$

The matrix T consists of zeros and ones where each row contains exactly one non-zero element. For example, to incorporate a blocking scheme where $u(0) = u(1)$ and $u(2) = u(3) = u(4)$ in an MPC problem with $N_p = 4$, the blocking matrix T is:

$$T = \begin{bmatrix} 1 & 0 \\ 1 & 0 \\ 0 & 1 \\ 0 & 1 \\ 0 & 1 \end{bmatrix} \tag{2.13}$$

In the case of using a quadratic cost function and linear constraints, the complete optimisation problem is then given by:

$$\begin{aligned}
&\min_{\tilde{U}} \quad ||x(k+N_p|k)||_P^2 \\
&+ \sum_{i=0}^{N_p-1} ||x(k+i|k) - x_{ref}(k+i|k)||_Q^2 + ||u(k+i|k) - u_{ref}(k+i|k)||_R^2 \\
&\text{subject to the linear state space plant model:} \\
&x(k+i+1|k) = A_d \cdot x(k+i|k) + B_d \cdot u(k+i|k) \quad \forall i \in \{0,1,...,N_p-1\} \\
&\text{subject to the blocking constraints:} \\
&U = [u(0|k), u(1|k), \dots u(N_p-1|k)]^T = T \cdot [\tilde{u}(0|k), \tilde{u}(1|k), \dots \tilde{u}(M-1|k)]^T = T \cdot \tilde{U} \\
&\text{subject to the initial conditions:} \\
&x(k|k) = x_{0|k} \\
&\text{subject to the inequality constraints:} \\
&F_{ineq} \cdot x(k+i|k) + E_{ineq} \cdot u(k+i|k) \leq g_{ineq} \quad \forall i \in \{0,1,...,N_p-1\}
\end{aligned} \tag{2.14}$$

2.2.3 Offset Move Blocking

In *offset move blocking*, the optimisation variables are offsets (deviations) from the infinite-horizon unconstrained linear-quadratic control $u(k+i|k) = K_{LQR} \cdot x(k+i|k)$ determined by solving the Riccatti equation. Instead of the control itself, the offsets $o(k+i|k)$ from this control law are subject to move blocking and thus constrained to be constant over several samples. The control trajectory is then given as $u(k+i|k) = K_{LQR} \cdot x(k+i|k) + o(k+i|k)$. [115]

This strategy allows more variations in the control than input blocking since the controls can vary at each sample and only the changes in comparison to the linear quadratic regulator are forced to be constant over several steps. [115]

By using a blocking matrix T in the same fashion as in the input move blocking approach and in the case of a quadratic cost function with linear constraints, the optimal control problem turns to:

$$
\begin{aligned}
&\min_{\tilde{O}} \quad ||x(k+N_p|k)||_P^2 \\
&+ \sum_{i=0}^{N_p-1} ||x(k+i|k) - x_{ref}(k+i|k)||_Q^2 + ||u(k+i|k) - u_{ref}(k+i|k)||_R^2 \\
&\text{subject to the linear state space plant model:} \\
&x(k+i+1|k) = A_d \cdot x(k+i|k) + B_d \cdot u(k+i|k) \quad \forall i \in \{0,1,...,N_p-1\} \\
&\text{subject to the blocking constraints:} \\
&u(k+i|k) = K_{LQR} \cdot x(k+i|k) + o(k+i|k) \quad \forall i \in \{0,1,...,N_p-1\} \\
&\hat{O} = [o(0|k), o(1|k), \dots o(N_p-1|k)]^T = T \cdot [\tilde{o}(0|k), \tilde{o}(1|k), \dots \tilde{o}(M-1|k)]^T = T \cdot \tilde{O} \\
&\text{subject to the initial conditions:} \\
&x(k|k) = x_{0|k} \\
&\text{subject to the inequality constraints:} \\
&F_{ineq} \cdot x(k+i|k) + E_{ineq} \cdot u(k+i|k) \leq g_{ineq} \quad \forall i \in \{0,1,...,N_p-1\}
\end{aligned}
\tag{2.15}
$$

2.2.4 Delta Blocking

In the *delta input blocking* strategy, not the controls $u(k)$ but their increments $\Delta u(k+i|k) = u(k+i+1|k) - u(k+i|k)$ are blocked over several prediction samples. This again allows more variations in the control and can lead to a better control performance than the pure input blocking strategy. [115]

The delta blocking can be combined with offset blocking (*delta offset blocking*) as well such that the increments of the offsets $\Delta o(k+i|k) = o(k+i+1|k) - o(k+i|k)$ are constrained to be constant for several samples. [115]

2.2.5 Stability, Feasibility and Tracking in the Case of Move Blocking

The methods of input parameterisation described above decrease the number of decision variables and thus dimension of the optimal problem to reduce the computational load.

A drawback of these methods is however, that the feasibility of the optimisation as well as the stability and the required reference tracking performance of the control may not be ensured any more. For input move blocking approaches, these issues have been addressed in research.

A relaxation of the constraints on the problem has been proposed in [119] to maintain the feasibility in the case of input move blocking. Both, feasibility and stability are also addressed by the *Moving Window Blocking* strategy presented in [115]. To increase the reference tracking performance of a controller subject to input move blocking, [120] proposes to include the weighted optimal control trajectory of the previous MPC step in the optimal control problem of the current step. Stability and performance for a standard MPC formulation with input blocking (constant reference, no known disturbance) has been investigated in [121]. It must be checked for the given control application case if these new methods are applicable and useful.

2.2.6 Input Parameterisation Using Polynomials

Another kind of input parameterisation is to restrict the input trajectory to take the form of polynomial functions. Some approaches use low-dimensional interconnected

piecewise polynomial functions (*splines*) that are forced to meet collocation points [106]. Alternatively, the control and state trajectory can be expanded as a series of orthogonal functions like Chebyshev [122], Legendre [123] or Laguerre [124] polynomials whose coefficients are then determined by the numerical optimisation.

These techniques are called *pseudospectral methods* and have widely been used in flight and military applications of non-linear MPC due to their efficiency [125]. The name *pseudospectral methods* comes from the similarity of the basic idea with *spectral methods* where a function is assumed to be composed of a series of trigonometric functions.

One advantage of using polynomial parameterisations is that the number of decision variables can be reduced in comparison to the standard piecewise constant control trajectory while allowing a high degree of freedom. Furthermore, the use of polynomial functions guarantees a smooth and thus eventually a technically preferable control trajectory. Finally, analytical derivatives of the cost function can eventually be calculated off-line to avoid a computationally exhaustive numerical derivation online.

2.2.7 Other Approaches

A different way of input parameterisation is the use of *principal control moves*. In this case, the control system is either simulated or run in experiments with a full-degree of freedom MPC controller. During this experimental closed-loop operation, the reactions of the specific control system to set-point changes or disturbances are recorded. These are called *principal control moves*. Then, these control moves are decomposed into orthonormal components. Their weighted sum can reproduce the original controller reactions that occurred during the experimental operation. With the help of these orthonormal components, the MPC problem is reformulated in such a way that the optimisation only determines the weightings on the orthonormal components of the principal control moves. Thus, the number of decision variables is reduced while the control system can still react equivalently to the full degree-of-freedom controller. However, previously unexpected situations require distinct actions of the controller and may not be considered beforehand. [126]

Apart from the aforementioned approaches of input parameterisation, there are also techniques to find suboptimal but faster solutions. One example is *multiplexed model-predictive control* where the different controls of a multi-input multi-output (MIMO) system are updated sequentially [127]. This splitting up leads to a suboptimal solution of the original problem but a faster computation.

2.3 Conclusion of the Chapter

A dynamic plant model, an objective function and optionally a set of inequalities are needed to determine optimised predictions of the behaviour of the real plant and the associated controls. Generally, every real plant behaves in a non-linear way. However, it is essential to check if a general non-linear formulation of the optimal control problem is necessary to achieve suitable predictions or if the problem can be posed in a more preferable way (e. g. using a convex problem formulation). The type of problem formulation determines which solution strategy can be used and finally, whether a real-time implementation is possible and which computational effort is necessary. This is a driving factor regarding the cost of implementation as well.

Together with these different approaches of modelling the predicted system, several methods to simplify the optimisation problem are available. If a proper input parameterisation can be found, the sample time of the MPC and the cost of implementation can significantly be reduced. Thus, the control engineer should investigate if these techniques are applicable in the specific case.

Chapter 3

Dynamic Modelling of the Electric Vehicle

The design and application of a model-predictive cruise controller requires a dynamic model to generate optimised predictions of the system behaviour. In principle, the same model can also be used as a simulation model to simulate the closed-loop operation of the control system.

In practice however, the prediction model must be kept simple such that an optimisation of the model outputs is possible in real-time. On the other hand, it is advantageous to use a more detailed simulation model that reflects the properties of the real plant more in detail. In this case, more realistic closed-loop simulations can be performed and the control performance can be evaluated more accurately. Eventual flaws in the control system can then already be found in the simulation.

Hence, the methodology of using models of two different complexity levels is also followed in this work. First, a more detailed vehicle model is derived which is thereafter simplified such that it can be used within the predictive controller.

To realise an energy-efficient cruise controller, mainly two physical values must be modeled. Firstly, the vehicle speed must be computed to evaluate the driving style regarding the tracking of a desired reference speed. Secondly, an energy consumption model is required to achieve the main goal of the controller - an energy-efficient driving style.

FIGURE 3.1: Electrical and mechanical components of the power train of the Smart Electric Drive, 3^{rd} generation, model 2012 [9].

The modelling of both the vehicle motion and the energy consumption are described in this chapter. In both cases, the models are identified and parameterised especially for the series-production electric vehicle that is used later on for the practical tests, a *Smart Electric Drive*, model 2012. For this reason, the technology of this particular vehicle is presented first.

3.1 Technology of the Smart Electric Drive

The *Smart Electric Drive* (Smart ED) is a battery electric vehicle. The main components of its power train and their mounting position in the vehicle are shown in Fig. 3.1.

The lithium-ion battery is located below the vehicle floor and can be charged at a standard household socket with the help of the integrated charging unit. The battery has a capacity of 52 Ah (17.6 kWh) and a nominal voltage of 390 V. The centre-piece of the Smart ED is a three-phase permanent-magnet synchronous machine with a maximum (peak) power of 55 kW and able to run as a motor or a generator. The electric machine is connected to the battery through power electronics (inverter) that convert the direct current of the battery into a three-phase current and vice versa. The power flow from/to the motor through the inverter is controlled by the accelerator and the brake pedal

TABLE 3.1: Main specifications of the test vehicle *Smart ED* [9]

description	**value**
continuous power rating	35 kW
peak power rating	55 kW
maximum motor torque	130 Nm
maximum driving speed	125 km/h
acceleration time 0 to 100 km/h	11.5 s
cruising range based on the New European Driving Cycle	145 km
energy consumption	15.1 kWh/100 km
battery capacity	17.6 kWh / 52 Ah
kerb weight	975 kg
admissible total weight	1150 kg
overall transmission ratio	9.922
tyre dimensions front	155/60R15T
tyre dimensions rear (driving axle)	175/55R15

positions. The motor torque is transmitted to the rear wheels through a gear box with a fixed transmission ratio. The main vehicle specifications are summarised in Tab. 3.1.

The simulation model of the vehicle must be designed to comply with the characteristics of the vehicle.

3.2 Vehicle Motion Model

To model the longitudinal dynamics of a vehicle (i. e. the driving speed in the main direction of motion), it is common practice to consider the car as a point mass with the mass m in a one-dimensional motion with the speed v [128]. The acceleration of the point mass is then given by Newton's second law of motion $\sum F = m \cdot dv/dt$ where the sum of the forces F represents the difference between the traction or brake force and the driving resistance forces in the direction of movement. The driving resistance forces are generally given as follows.

The air drag resistance force F_d results from the friction and the displacement of the surrounding air and depends on the square of the driving speed v [128]:

$$F_d(t) = \frac{1}{2} \cdot c_d \cdot A_v \cdot \rho_a \cdot v(t)^2 \tag{3.1}$$

The coefficients are related to the shape of the vehicle (projected front surface A_v and air drag coefficient c_d) and the density of the surrounding air ρ_a.

The grade resistance force F_{gr} describes the longitudinal component of the gravitational force acting on the car in up- or down-hill driving depending on the road slope angle α_{sl}. [128]

$$F_{gr}(t) = (m_v + m_l) \cdot g \cdot \sin\big(\alpha_{sl}(t)\big) \tag{3.2}$$

The parameters m_v and m_l represent the mass of the empty vehicle and its current payload. g is the gravitational constant.

The rolling resistance force F_r is an effect of the deformation of the tyres when rolling and depends on the load on the wheels (the normal force acting between the wheel and the road) and thus on the road slope angle α_{sl}. [128]

$$F_r(t) = c_r \cdot (m_v + m_l) \cdot g \cdot \cos\big(\alpha_{sl}(t)\big) \tag{3.3}$$

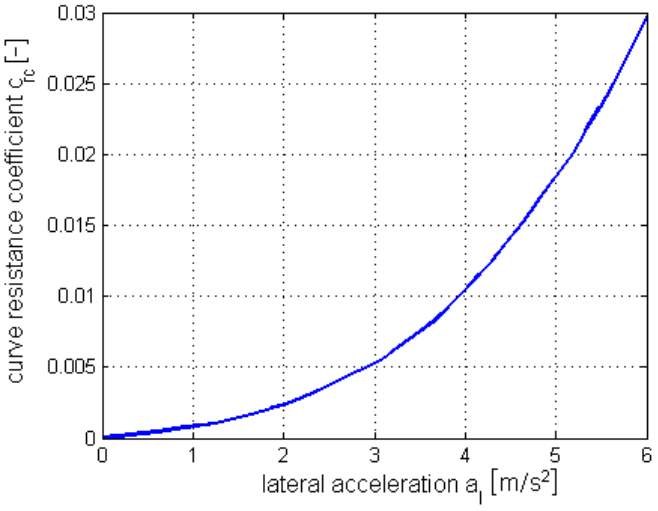
c_r represents the rollings resistance coefficient of the specific tyres and can be extracted from data sheets.

The aforementioned forces are the main resistance forces in longitudinal direction. However, the lateral dynamics influence the longitudinal motion as well. When driving through a curve, the rolling resistance of the tyres increases. This additional curve resistance force can be modeled as [10]:

$$F_c(t) = (m_v + m_l) \cdot g \cdot c_{rc}\Big(a_l\big(v(t), r_c(t)\big)\Big) \tag{3.4}$$

The curve resistance coefficient c_{rc} is a function of the lateral acceleration which itself depends on the driving speed v and the curve radius r_c.

FIGURE 3.2: Relation between the curve resistance coefficient c_{rc} and the lateral acceleration of the vehicle a_l for a tyre of dimensions 195/65/V15. Figure edited from [10].



$$a_l(t) = \frac{v(t)^2}{r_c(t)} \tag{3.5}$$

The relation of c_{rc} and a_l is given in Fig. 3.2 for a tyre of the dimensions 195/65/V15 [10]. This relation is also assumed to be valid, here.

In summary, the motion equation can be posed according to Newton's second law of motion as the difference between the traction force at the wheels F_{trac} applied by the motor or brakes of the vehicle and the aforementioned driving resistance forces, divided by the equivalent mass of the vehicle m_{eq}. The equivalent mass includes the inertia of the rotational parts of the drive train that are subjected to the acceleration of the vehicle (motor, transmission drive shafts, wheels). The equivalent mass can be computed as $m_{eq} = e_i \cdot (m_v + m_l)$, where the value of the gain factor e_i is assumed to be 1.01.

$$\frac{dv(t)}{dt} = \Big(F_{trac}(t) - F_d(t) - F_{gr}(t) - F_r(t) - F_c(t)\Big)/m_{eq} \tag{3.6}$$

Fig. 3.3 shows the forces acting on the vehicle in longitudinal direction.

The parameters of the motion model can mostly be extracted from data sheets of the vehicle manufacturer [9]. In addition, the rolling resistance coefficient of the tyres is

FIGURE 3.3: Main forces acting on the vehicle in longitudinal direction.

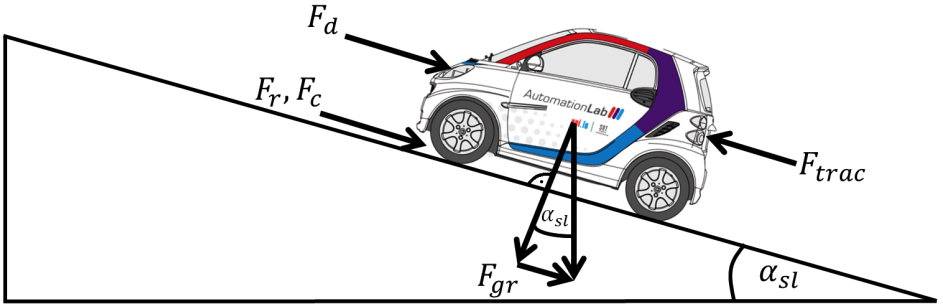


TABLE 3.2: Parameters of the motion model. [9]

symbol	**value**	**symbol**	**value**
A_v	1.95 m^2	m_{eq}	1070 kg
c_d	0.37	m_l	160 kg
c_r	0.01	m_v	900 kg
g	9.81 m/s^2	ρ_a	1.2 kg/m^3
e_i	1.01		

assumed to be $c_r = 0.01$, the gravitational constant $g = 9.81 m/s^2$ and the density of the surrounding air $\rho_a = 1.2 kg/m^3$. All parameters are summarised in Tab. 3.2.

The control input of the motion model (3.6) is the traction force at the wheels F_{trac} which will be determined by the MPC controller. Additional given inputs are the road slope angle α_{sl} and the curve radius r_c that can be read from the digital maps of a navigation system in an anticipatory way if the route is known.

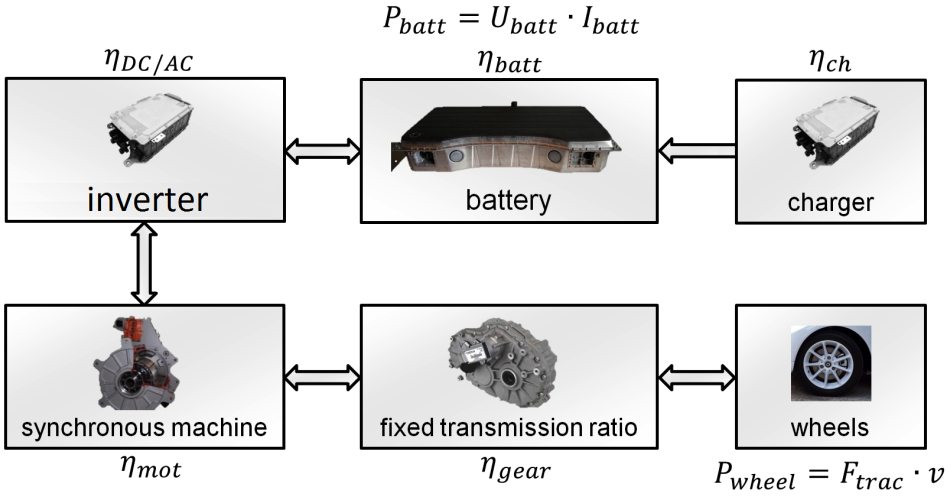
In summary, the main assumptions of the vehicle motion model are:

- The vehicle is considered as a point mass in a one-dimensional motion.
- The mass of the vehicle does not vary.
- The wind speed is zero.
- Tyre slip is disregarded.

3.3 Vehicle Energy Consumption Model

To give an overview on the power flow in the vehicle, Fig. 3.4 shows the main drive train components of the car and their relations.

FIGURE 3.4: Main electrical and mechanical power flow within the vehicle.



Strictly speaking, the energy consumption of the vehicle is the energy withdrawn from the electrical power grid by the charger when charging the battery. Like every technical device, the charger and the battery have a certain efficiency ratio and cannot fully supply all the energy to the vehicle that they have consumed during charging.

However, since the process of charging cannot be influenced by a cruise controller, the energy consumption is considered here as the amount of energy that is discharged from the battery. This value can directly be influenced by actuating the accelerator pedal and controlling the motor torque and is thus chosen as the variable to be modeled.

A battery can in general be seen as an energy storage or a charge storage device. The electrical energy supplied by the battery E_{batt} is the integral of the battery power that is given by the product of the terminal battery voltage U_{batt} and the battery current I_{batt}.

$$E_{batt} = \int P_{batt} \cdot dt = \int U_{batt} \cdot I_{batt} \cdot dt \tag{3.7}$$

In contrast, the battery charge consumption C_{batt} is represented by the integral of the battery current over time.

$$C_{batt} = \int I_{batt} \cdot dt \tag{3.8}$$

The charge consumption is generally used to describe the state of charge SOC of a battery. It is defined as the ratio of remaining charge within the battery (or its momentary capacity) and the total capacity C_{tot}. Hence, SOC can take values between 0 and 1. [128]

$$SOC = \frac{C_{tot} - \int_{t_0}^{t_{end}} I_{batt} \cdot dt}{C_{tot}} \tag{3.9}$$

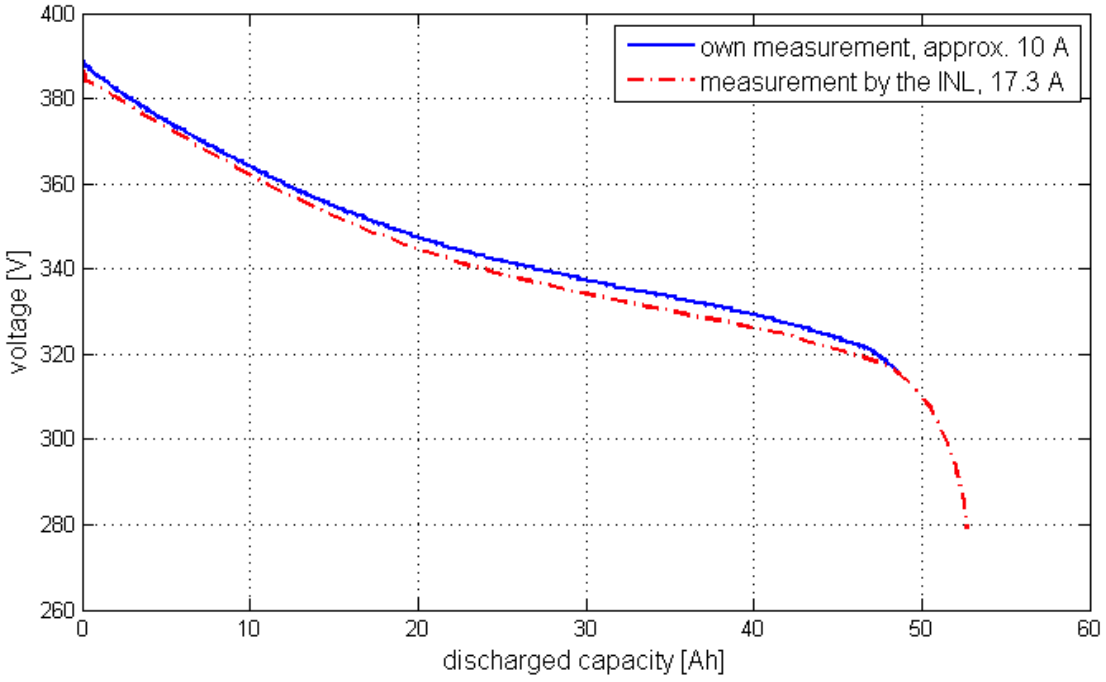
Consequently, a charge or energy consumption model depends on the battery current and (in case of the energy consumption) the voltage which themselves depend on the operating point of the vehicle. As presented in Fig. 3.4, the characteristics of the electric machine together with the power electronics and the battery itself determine the battery current. The battery voltage mainly depends on the discharge current I_{batt}, the state of charge SOC and the cell temperature.

The dependency of the battery voltage U_{batt} on the state of charge at a constant discharge current is called the *discharge curve* of a battery. It is available from data sheets [11] of the *Smart ED* and has been measured in own experiments with the vehicle in the laboratory as well. Since the battery is mounted within the car and the risk of a removal is estimated too high, the battery has been discharged in the mounted position by switching on the air conditioning system of the vehicle which withdraws a nearly constant discharge current of between 8 and 11 A. This way, the battery has been fully discharged within around 5 hrs. The discharge curve provided in data sheets by the *INL* [11] and the one measured in own experiments are given in 3.5. The curve measured at 17.3 A displays a lower voltage level than the curve measured at a discharge rate of 10 A since the higher discharge current leads to a lower terminal voltage. Considering this, both curves show a very good accordance.

From the discharge curve it becomes clear that the battery voltage drops significantly with the state of charge. As a consequence, it requires a higher battery current I_{batt} to obtain the same electrical power $P_{batt} = U_{batt} \cdot I_{batt}$ at lower states of charge where also the terminal voltage U_{batt} is lower.

As a result, the current consumed by the power train increases at lower states of charge when the same electrical power is demanded from the battery. On the other hand, the electrical power consumption remains approximately constant for different states of

FIGURE 3.5: Discharge curves of the lithium-ion battery of the *Smart ED*. The red dash-dotted curve is measured at a constant discharge current of 17.3 A (3 hrs. discharge time) at the *Idaho National Laboratory (INL)* [11]. The blue curve has been obtained in own measurements at a discharge rate of approximately 10 A (approx. 5 hrs. discharge time).



charge. Hence, a charge consumption model (based on the integration of the battery current over time) must be dependent on the state of charge while a power consumption model (based on the integration of the battery current and voltage) can be assumed to be independent of the state of charge.

Thus, it is decided to model the energy consumption of the vehicle given by the integral of the battery power over time to achieve a simple model that is valid for various states of charge.

A detailed model of all physical processes in the power train components is not suitable for the simulation of a cruise control loop due to its complexity. The extremely fast switching times of the power electronics for instance would require very small simulation step sizes and slow down the simulation without benefits regarding the accuracy of the speed and energy consumption prediction.

To find a simple model being able to represent the energy consumption dynamics accurately, another method is chosen. The battery current and voltage are measured on a professional dynamometer test bench (see Fig. 3.6) in different quasi-static operating

FIGURE 3.6: Measurement of the power train characteristics on a professional dynamometer test bench at the company *DELPHI*, Bascharage, Luxembourg.

points of the vehicle (defined by driving speed and traction force) and stored to create a characteristic map of the power train characteristics. This map can then be used in the simulations as a lookup table.

Quasi-static modelling is widely used for modelling of the energy consumption of fuel-powered as well as electric cars and has proven to be more accurate as well as simpler than analytical models in many cases [128].

In the case of modelling the energy consumption of an electric vehicle, the dynamics of the current and voltage changes can be neglected since the dynamics of the vehicle motion are by far slower than the dynamics of the electrical variables. Thus, the delays in the electrical variables can be neglected in comparison.

The operating point of the electric machine and the inverter is mainly defined by the (demanded and actual) torque of the electrical machine T_{eng} and its rotational speed ω_{eng}. Since the *Smart ED* regarded in this work like many electric vehicles has only one fixed transmission ratio i_{tot}, the operating point can directly be related to the driving speed and the traction force at the wheels which are the variables used in the motion model (3.6) by using the following relations.

$$\begin{aligned} F_{trac} &= i_{tot} \cdot \frac{T_{eng}}{r_{dyn}} \\ v &= \frac{\omega_{eng} \cdot r_{dyn}}{i_{tot}} \end{aligned} \tag{3.10}$$

Herein, r_{dyn} is the dynamic wheel radius which represents the loaded radius of the wheel during driving. It can be read from data sheets according to the size of the rear tyres of the *Smart ED*.

To set up the lookup table for the quasi-static consumption model, the battery power $P_{batt} = U_{batt} \cdot I_{batt}$ is assumed to be a non-linear function of the traction force at the wheels F_{trac} and the driving speed v.

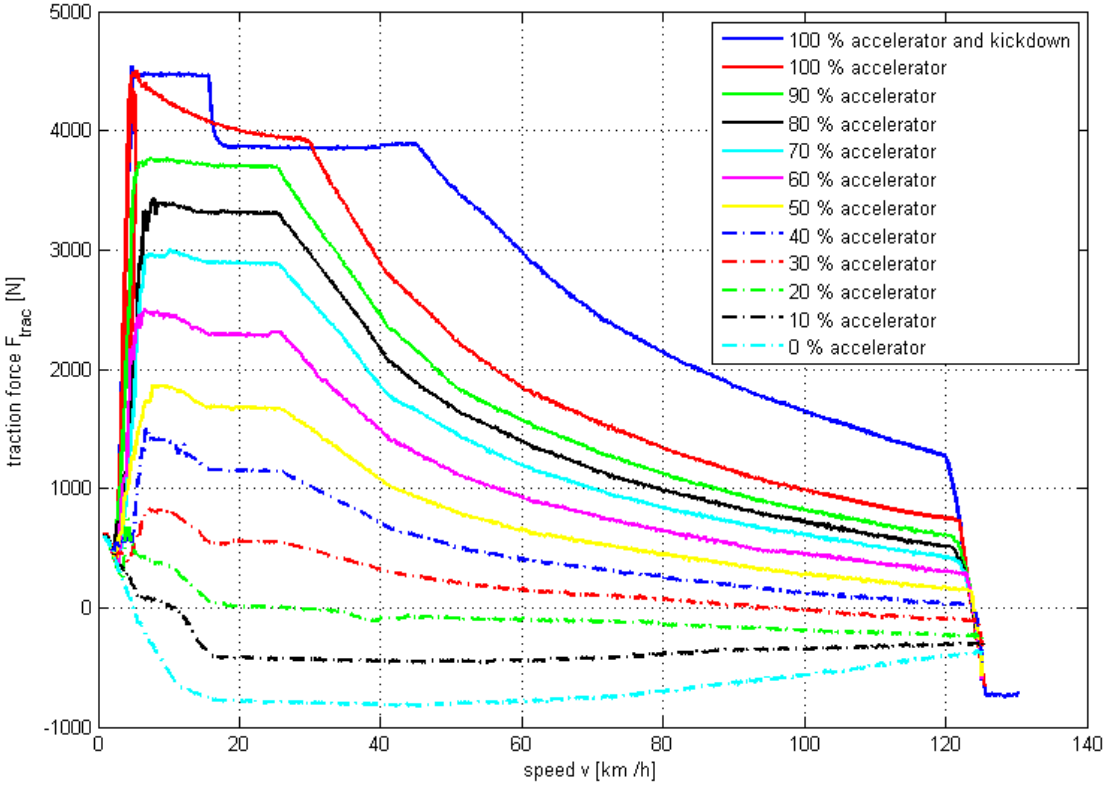
In order to cover a suitable grid of operating points, the dynamometer test bench starts at an initial speed of 5 km/h and increases the speed linearly up to 125 km/h, the maximum speed of the *Smart ED*. This linear increase is performed slowly (quasi-statically) within two minutes. This procedure is repeated with different fixed accelerator pedal positions between 0 % (released pedal) and 100 % (pedal fully pushed) in 5 % increments. The *Smart ED* features a boost switch (*kickdown*) below the accelerator pedal which can be pushed for a temporary use of the maximum acceleration of the vehicle (e. g. for overtaking). The traction force at boosting is also measured and referred to as "100 % accelerator and kickdown".

During the test runs, the dynamometer test bench measures and logs the speed at the wheel v and the traction force F_{trac}. At the same time, the battery current I_{batt}, the battery voltage U_{batt} and the driving speed v are read through the on-board diagnosis (OBD) interface of the vehicle and logged on a PC. The redundant speed measurement is used afterwards to synchronise the measurements of the dynamometer and those of the vehicle.

The load curves of all test runs measured by the dynamometer are given in Fig. 3.7. To avoid the possibility of limited power supply, the battery state of charge has been kept above 70 % throughout all measurements.

It should be noted that electrical energy is recovered when the accelerator pedal is released or only slightly pushed. This results in negative traction forces. (Here, only coasting is considered when the accelerator is released and the brake pedal is not pushed. Higher energy recovery is possible when the brake pedal is actuated. This however goes along with friction losses due to the mechanical brakes.)

FIGURE 3.7: Traction force (F_{trac}) measurements of the dynamometer test bench at different accelerator pedal positions plotted over the driving speed v.



By combining the traction force and speed data with the battery variables current and voltage, the power consumption map and the current consumption map can be set up. The interpolated characteristics are given in Figs. 3.8 and 3.9.

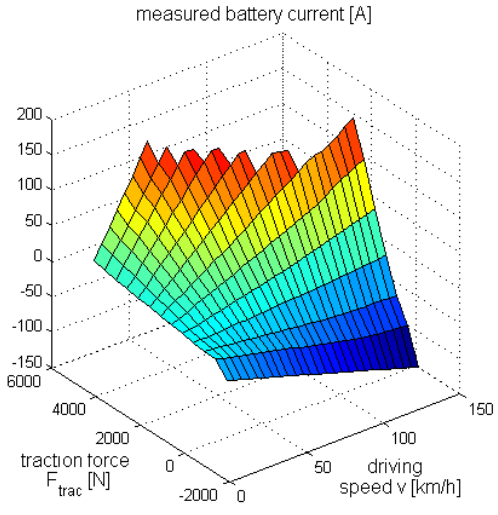


FIGURE 3.8: Interpolated current consumption map, valid for high SOCs above 70 %.

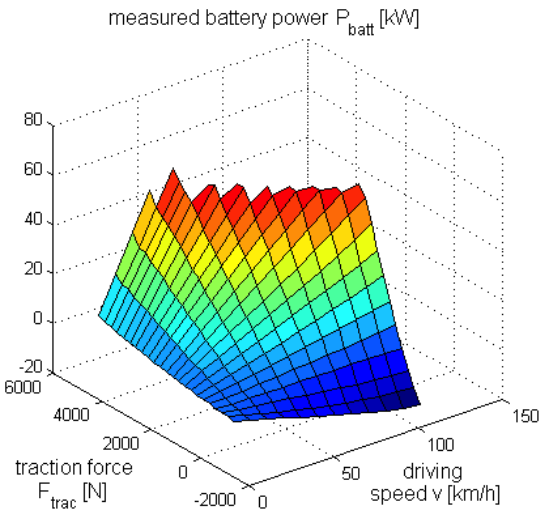


FIGURE 3.9: Interpolated power consumption map.

This data includes the characteristics of the inverter, the synchronous machine, the transmission and the wheels and thus, all types of electrical, thermal and friction energy losses within these components are considered in this model.

All auxiliary electrical devices have been disabled (lights, climatisation, etc.) throughout the measurements. This must be kept in mind and eventual auxiliary loads must be added to the total consumption when using the map data for a prediction of the energy consumption.

The energy consumption map has been measured in an environment of approximately 20 $°C$ at normal operating temperature of the vehicle (the vehicle was cooled with an air stream during the measurements, see Fig. 3.6). The presented power train model assumes that the energy consumption characteristics are not influenced by thermal effects. This assumption is admissible since the vehicle uses a controlled cooling water circuit to maintain a constant temperature of the electrical components during operation. If the vehicle is operated in very cold environments or the normal operating temperature is exceeded / cannot be maintained, the power train characteristics may vary which could then lead to a model mismatch.

Furthermore, it has been assumed that the requested power can always be supplied by or charged into the battery. This assumption can be validated by the following measurements published by the *INL* [11] showing the maximum power that can be charged into or discharged from the battery depending on the remaining capacity of the battery.

According to Figs. 3.10 and 3.11, the assumption that the requested power can always be charged into or discharged from the battery is admissible because a power of 50 kW, can always be maintained by the battery if a discharge pulse of a duration of 30 s or a charge pulse of 10 s is considered. If the temporary boosting feature (kickdown) is not used, this is more than the electric machine and the inverter require in nominal operation (cf. Fig. 3.9).

Other effects like the *Peukert effect* (dependency of the capacity on the discharge current) and aging of the battery are disregarded. These assumptions are admissible for modern lithium-ion batteries.

In summary, the main assumptions of the charge consumption model are:

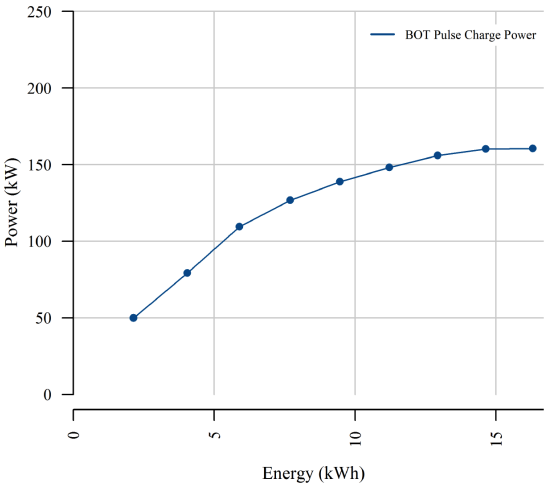


FIGURE 3.10: Maximum 10 s charge pulse power of the vehicle battery based on measurements by the *Idaho National Laboratory (INL)* [11].

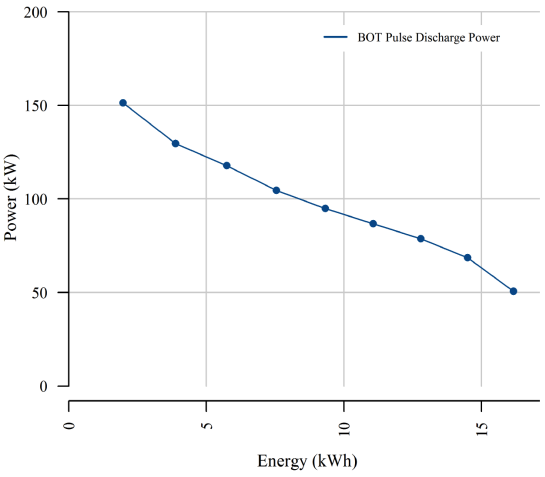


FIGURE 3.11: Maximum 30 s discharge pulse power of the vehicle battery based on measurements by the *Idaho National Laboratory (INL)* [11].

- The power consumption is modeled by a static map as a function of the traction force at the wheels and the driving speed. Dynamic effects are disregarded.
- The static power consumption map does not depend on temperature effects.
- The power consumption map does not depend on the state of charge of the battery.
- The demanded amount of electrical power can always be charged into or discharged from the battery.

Measurement data of the battery and the vehicle technology configuration validate these assumptions. A further validation of the simulation model is provided in the sequel by driving tests.

3.4 Model Validation

The simulation model described in the previous sections is validated by on-road driving experiments. Throughout the test drives, the accelerator pedal position, the battery current I_{batt}, the battery voltage U_{batt} and the driving speed v are measured through the OBD interface reading from the CAN bus of the vehicle. Since the simulation model requires the traction force as an input, it needs to be derived from the available measurements. This is achieved by using a lookup table to estimate the traction force

FIGURE 3.12: Characteristic map to determine the traction force at the wheel from the measured battery power P_{batt} and the measured driving speed v.

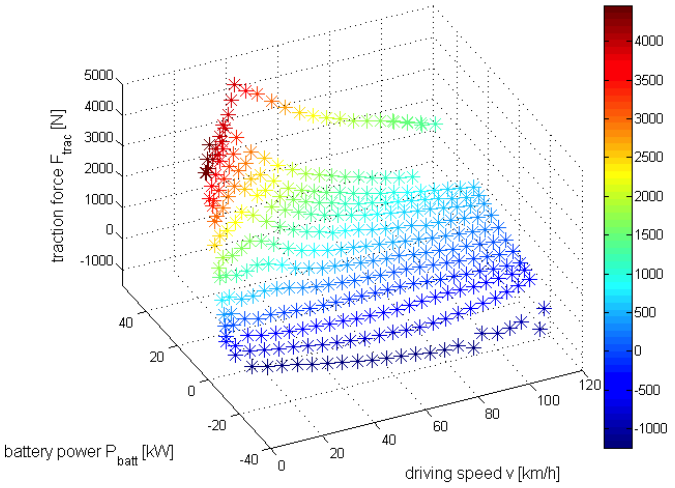
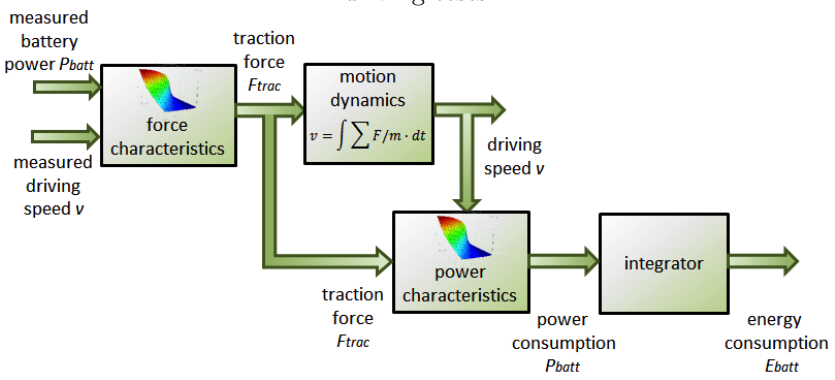


FIGURE 3.13: Model structure for the validation of the simulation results by on-road driving tests.

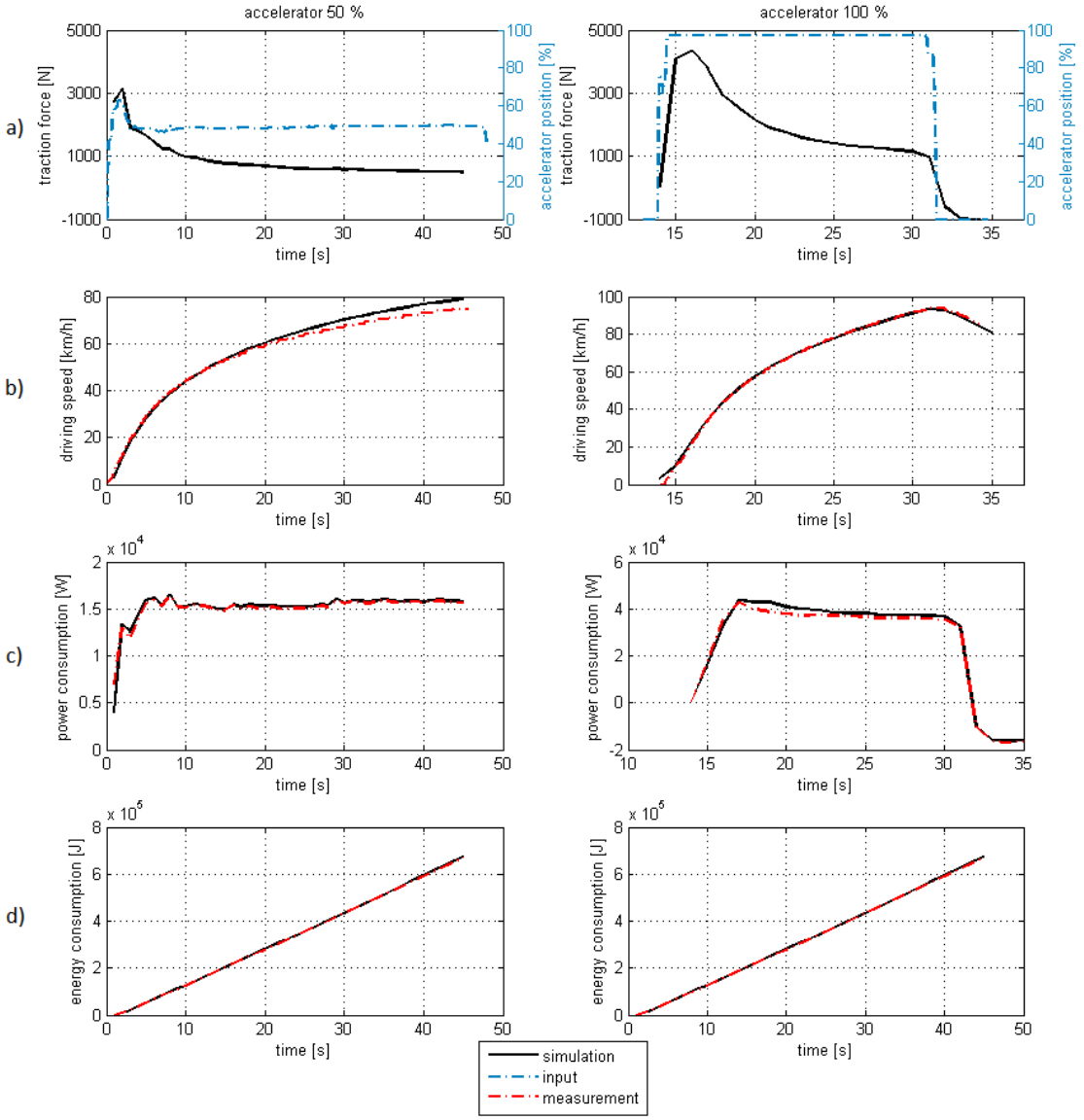


from the battery power and the driving speed measurement. The table uses the same data as the map in Fig. 3.9, however with a different assignment to the axes as presented in Fig. 3.12.

The battery power (as a product of the battery current and voltage) is chosen as an indicator to determine the traction force because the characteristics here do not depend to a large extent on the state of charge of the vehicle battery. The complete model structure for the validation is given in Fig. 3.13.

The lookup table used to compute the traction force and the one used to obtain the power consumption are based on the same data. Nevertheless, the model validation can be carried out in this way since the traction force is read from the lookup table by using the *measured speed* as an input while the power consumption is obtained by using the

FIGURE 3.14: Results of the model validation by on-road driving test data in acceleration maneuvers with constant pedal position. The blue dash-dotted curves show the measured accelerator position input. The red dash-dotted curves show the measurements, the black solid curve the simulated data.



simulated speed. If the characteristic map giving the traction force was imprecise, this would result in an imprecise simulation of the driving speed and then consequently lead to an imprecise battery power simulation. Thus, both the energy consumption model as well as the motion model can be validated according to Fig. 3.13.

Firstly, accelerations from standstill on an even road with constant accelerator positions are considered. Fig 3.14 presents the results from two acceleration maneuvers, one with 50 % accelerator position and one at full load (100 % accelerator and kickdown).

Fig. 3.14a shows the measured accelerator position throughout the experiment (blue

dash-dotted line). It is the input for the real system and applied by the human driver. The resulting battery current and voltage are measured to compute the battery power $P_{batt}(t) = U_{batt}(t) \cdot I_{batt}$. This in turn serves as the input to the simulation model. The traction force is determined with the help of a lookup table from the measured battery power and driving speed, see Fig. 3.14a, black solid line. With this traction force, the simulated driving speed can be computed by integration of (3.6). All simulations are carried out within *MATLAB / Simulink*. Fig. 3.14b shows the comparison of the measured and the simulated speed. The slight deviations can result from variations in the road slope angle which could not be measured in the experiment and is assumed to be zero. The battery power is again read from a lookup table including the characteristics of Fig. 3.9 in dependence on the instantaneously simulated traction force and the simulated driving speed. Fig. 3.14c-d shows that the computation of the battery power provides a very good fit to the measured data in all operating points. This allows to conclude that the determination of the traction force is also accurate.

Finally, a scenario with a wide range of driving speeds, traction forces and fast changes of the battery power is chosen for an on-road validation. Fig. 3.15 shows the comparison between the measurements and the simulation results.

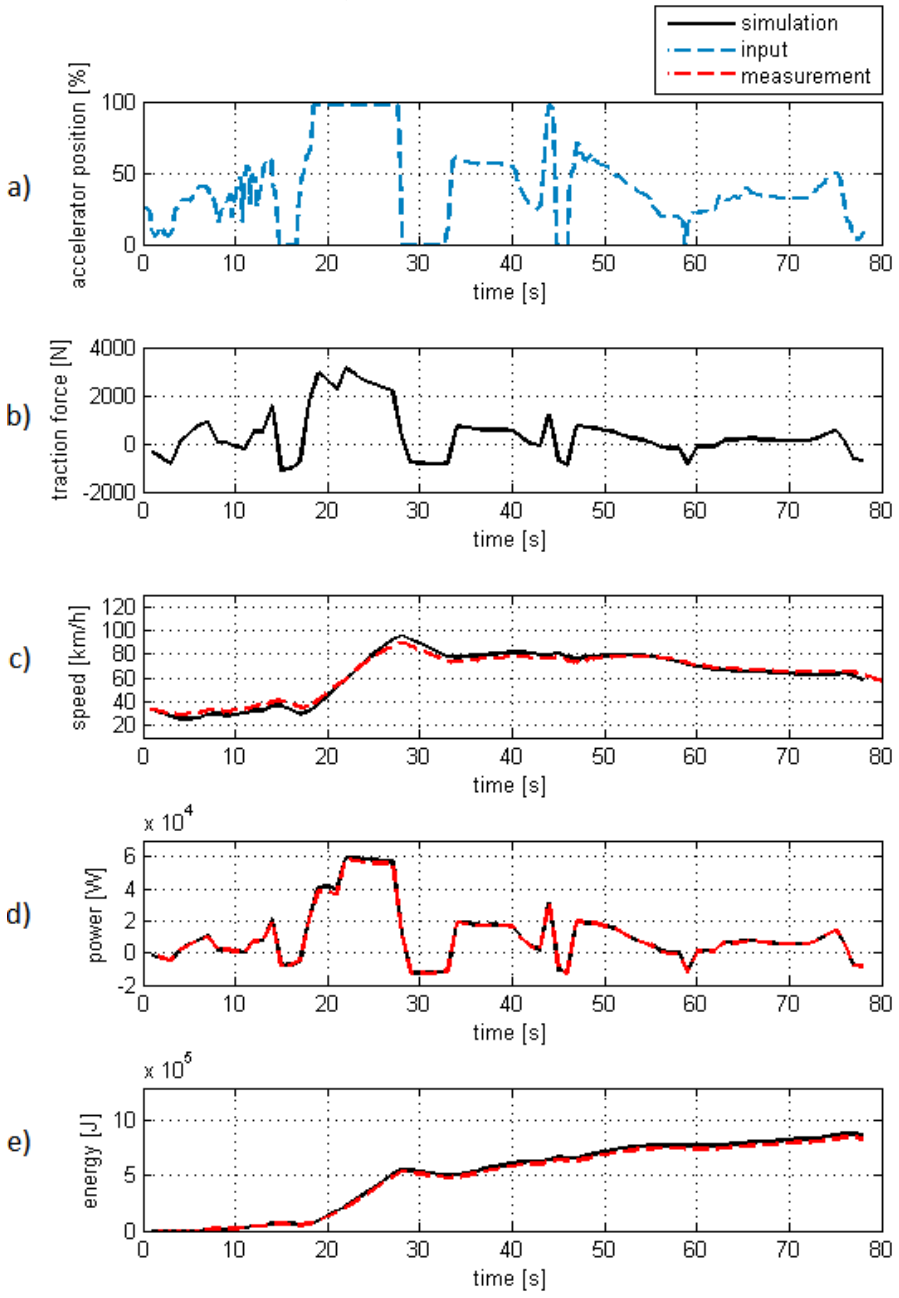
All simulation results in Fig. 3.15 again show a very good accordance with the measurements. The results show that the proposed model provides accurate results over a wide range of operating points in a realistic scenario with a highly varying input. This validates the quasi-static modelling of the drive train by lookup tables. This experiment was carried out with a battery state of charge of approximately 40 % while the characteristic maps were measured at a charging level of above 70 %. This shows that the model is also valid over a wide range of battery charge levels.

The scenarios above include energy recovery (negative traction force and power consumption) by releasing the accelerator pedal, however without engaging the mechanical brakes. The simulated energy recovery also matches the measurements very well.

3.5 Conclusion of the Chapter

The preceding sections have presented a simulation model of the driving speed and the charge consumption of the vehicle identified by measurements with the real *Smart*

FIGURE 3.15: Results of the model validation by on-road driving test data with varying input. The blue dashed curve shows the accelerator pedal input, the red dashed curves show the measurements, the black solid curve the simulated data.



ED. This consists of a dynamic motion model based on Newton's second law and an energy consumption model based on characteristic maps. By quasi-static modelling using measured data, the characteristics of the whole power train can be considered in a relatively simple model structure.

Measurements of the vehicle behaviour in on-road driving tests have validated the simulation model and have shown a very good match of the measured and simulated data. Furthermore, the model could be validated for different states of charge.

The presented simulation model is later-on used as a basis for the prediction model of the MPC controller for energy-efficient cruise control.

The main results of this chapter have been published in [129, 130].

Chapter 4

Controller Design

The design of an MPC controller includes the prediction model, the setup of the cost function including the weighting factors, the definition of inequality constraints representing actuator and plant limitations and the choice of a proper prediction horizon. As mentioned in Chapter 2, the way the prediction model, the cost function and the constraints are formulated determines the properties of the resulting optimal control problem and thus the necessary computational effort and available techniques to solve it.

Against this background, the choice of a prediction model that is accurate enough to represent the main characteristics of the vehicle on the one hand, but simple enough to be solvable in real-time on the other hand is of particular importance.

4.1 Prediction Model

The motion and consumption models presented in Chapter 3 are taken as a basis for the prediction model of the MPC controller. However, these models contain non-linearities such as a characteristic map implemented as a lookup table and is hence not given in a preferable form for a prediction model within an optimal control formulation. The following sections present simplifications and reformulations to achieve a more favourable formulation of the motion and energy consumption model.

4.1.1 Controller Motion Model

The motion model (3.6) is taken as a basis for the controller prediction model and is repeated at this point for the ease of reading.

$$\frac{dv(t)}{dt} = \Big(F_{trac}(t) - F_d(t) - F_{gr}(t) - F_r(t) - F_c(t)\Big)/m_{eq} \tag{4.1}$$

It includes the driving resistance forces:

$$\begin{aligned} F_d(t) &= \frac{1}{2} \cdot c_d \cdot A_v \cdot \rho_a \cdot v(t)^2 \\ F_{gr}(t) &= g \cdot (m_v + m_l) \cdot \sin\big(\alpha_{sl}(t)\big) \\ F_r(t) &= c_r \cdot (m_v + m_l) \cdot g \cdot \cos\big(\alpha_{sl}(t)\big) \\ F_c(t) &= (m_v + m_l) \cdot g \cdot c_{rc}\Big(a_l\big(v(t), r_c(t)\big)\Big) \end{aligned} \tag{4.2}$$

and the lateral acceleration of the car:

$$a_l(t) = \frac{v(t)^2}{r_c(t)} \tag{4.3}$$

In this formulation, (4.1) comprises several non-linearities.

The road slope angle α_{sl} is a known input of the model. As a first step towards a simpler model, this known input can directly be replaced by the sine and the cosine functions of α_{sl} which can be precomputed. Consequently, the rolling resistance and grade resistance force become linear functions of the new known inputs $d_1 = \cos(\alpha_{sl})$ and $d_2 = \sin(\alpha_{sl})$.

After the aforementioned substitution, the resulting model is still non-linear due to the dependency of the air drag resistance force F_d on the square of the driving speed and the dependency of the curve resistance on the lateral acceleration which itself depends on the squared driving speed.

An additional challenge in the implementation is that the model inputs road slope angle and road curvature as well as the speed limits are functions of the position on the road

- and not directly the time. Thus, it is favourable to change the domain (i. e. the independent variable) of the model from the time t to the position s. This domain change can generally be achieved by multiplying both sides of (4.1) with the differential quotient

$$\frac{dt}{ds} = \frac{1}{v(t)} \tag{4.4}$$

which is just the inverse of the driving speed v. However, this would lead to a highly non-linear model depending on the inverse of a state variable and is thus not preferable in a prediction model of a fast model-predictive controller. In this context, it is advantageous to model the kinetic energy of the moving car

$$e_{kin}(t) = \frac{1}{2} \cdot m_{eq} \cdot v(t)^2 \tag{4.5}$$

as a measure of the driving speed v. With the restriction to non-negative speeds ($v \geq 0$), the kinetic energy is a unique function of the driving speed such that the speed values can be computed from the kinetic energy and vice versa. This change of variables has been inspired by others works [76, 85]. To obtain a model as a function of the position on the road, the derivative of (4.5) is taken with respect to s as follows:

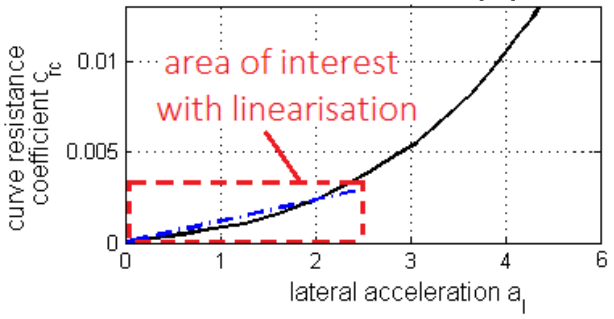
$$\frac{de_{kin}}{ds} = m_{eq} \cdot v \cdot \frac{dv}{ds} = m_{eq} \cdot \frac{ds}{dt} \cdot \frac{dv}{ds} = m_{eq} \cdot \frac{dv}{dt} \tag{4.6}$$

A closer look at (4.6) shows that the derivative of the kinetic energy of the moving car e_{kin} with respect to the position s is just the same relation as the original model of the driving speed with respect to the time multiplied by the equivalent mass of the vehicle m_{eq}:

$$\frac{de_{kin}(s)}{ds} = F_{trac}(s) - F_d(s) - F_{gr}(s) - F_r(s) - F_c(s) \tag{4.7}$$

Herein, all terms related to the driving speed v are substituted by $v = \sqrt{\frac{2 \cdot e_{kin}}{m_{eq}}}$ and all inputs can now be given as a function of the position on the road s:

FIGURE 4.1: Relation between the curve resistance coefficient c_{rc} and the lateral acceleration a_l of the vehicle for tyre dimensions of 195/65/V15 with linearisation in the relevant area. Figure edited from [10].



$$
\begin{aligned}
F_d(s) &= \frac{1}{m_{eq}} \cdot c_d \cdot A_v \cdot \rho_a \cdot e_{kin}(s) \\
F_{gr}(s) &= (m_v + m_l) \cdot g \cdot d_1\big(\alpha_{sl}(s)\big) \\
F_r(s) &= c_r \cdot (m_v + m_l) \cdot g \cdot d_2\big(\alpha_{sl}(s)\big) \\
F_c(s) &= (m_v + m_l) \cdot g \cdot c_{rc}\Big(a_l\big(e_{kin}(s), r_c(s)\big)\Big) \\
a_l(s) &= \frac{2}{m_{eq} \cdot r_c(s)} \cdot e_{kin}(s)
\end{aligned}
\tag{4.8}
$$

Apart from the advantage that (4.7) is now formulated depending on the position, the air drag resistance force F_d and the lateral acceleration of the vehicle a_l are now represented as linear functions of the kinetic energy e_{kin}. Hence, the non-linearities in these equations related to the square of the driving speed have been eliminated without any loss of model accuracy.

The only remaining non-linear expression in (4.7) is the dependency of the curvature coefficient on the lateral acceleration $c_{rc}\Big(a_l\big(e_{kin}(s), r_c(s)\big)\Big)$ that is given in Fig. 3.2. A closer look at this relation shows that it covers areas with extremely high accelerations that are only relevant in race driving and never take effect in comfortable driving. By limiting the graph to a maximum of 2.5 m/s^2, the relation can be linearised with high accuracy. This is illustrated in Fig. 4.1.

Using the presented approximation, the curve resistance force can be expressed as a linear function of the kinetic energy e_{kin} of the moving car with the help of the linearised curve

resistance coefficient c_{rc}^*:

$$\begin{aligned} c_{rc} &\approx c_{rc}^* \cdot a_l(s) \\ F_{rc} &\approx (m_v + m_l) \cdot g \cdot c_{rc}^* \cdot \frac{v(t)^2}{r_c(s)} = \frac{2 \cdot (m_v + m_l) \cdot g}{m_{eq} \cdot r_c(s)} \cdot c_{rc}^* \cdot e_{kin}(s) \end{aligned} \tag{4.9}$$

The approximate curve resistance force is a linear function of the kinetic energy e_{kin}, however with the position dependent parameter $r_c(s)$. Since the curve radius $r_c(s)$ is known in advance from digital maps and since the model is formulated in dependency on the position s, this can be handled by a position variant but still linear state space model.

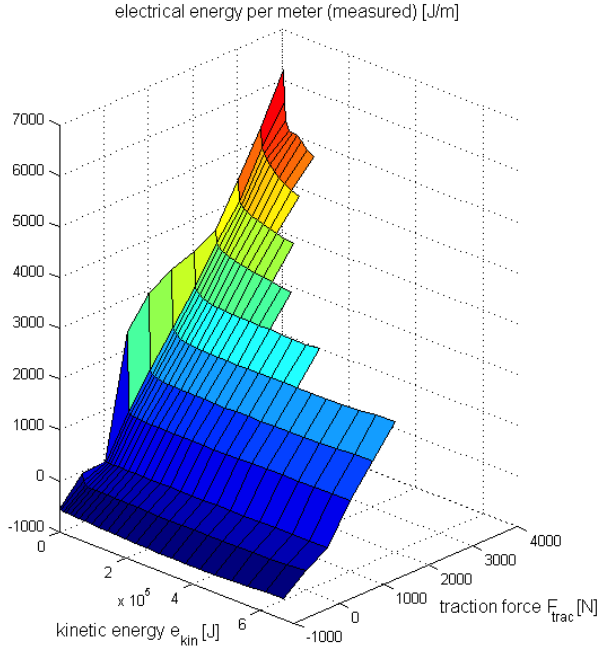
Finally, the motion model of the vehicle can be formulated as a linear model of the kinetic energy e_{kin} in the position domain which is a huge advantage in terms of computation time of the MPC input. This change of variables implies that references and constraints on the driving speed must be converted into the equivalent kinetic energy values before the optimisation and that the optimised speed trajectory must be calculated from the optimised kinetic energy values after the optimisation. However, this has no relevant influence on the computation time of the MPC control.

4.1.2 Controller Energy Consumption Model

The energy consumption model presented in Section 3.3 is based on a static characteristic map represented by a lookup table. However, the characteristic map of the power consumption, given in Fig. 3.9, is formulated as a function of the driving speed and the time (with the electrical power being the consumed energy per second). As a consequence, the presented energy consumption model does not comply with the prediction model of the vehicle motion (4.7) because it does not use the same variables.

To resolve this issue, the x-axis of the power consumption map in Fig. 3.9 is rescaled in terms of kinetic energy by using (4.5). Additionally, each data point of the map is divided by its associated driving speed to obtain the energy consumption per travelled distance according to the reformulation (4.4). This is possible since the transmission of the electric vehicle has only one fixed gear ratio such that the rotational speed of the electric machine has a constant relation to the rotational speed of the wheels (and thus the driving speed in the absence of tyre slip).

FIGURE 4.2: Energy consumption of the vehicle per travelled distance as a function of the kinetic energy e_{kin} and the traction force at the wheel F_{trac}



These transformations finally lead to a non-linear relation of the energy consumption per meter depending on the kinetic energy e_{kin} of the moving car and the traction force F_{trac} in the form of a lookup table $E_{batt} = f\big(e_{kin}(s), F_{trac}(s)\big)$. It is given in Fig. 4.2.

A lookup table however cannot easily be implemented in an MPC formulation and would moreover lead to a general non-linear model-predictive controller with high computation time for the optimisation and no guarantees on convergence and stability.

Hence, it is desirable to convert the lookup table into an analytical relation. In many related works, this is achieved by using polynomial approximations of the power consumption map e. g. [60, 131]. A closer look at Fig. 4.2 however shows that the given behaviour can hardly be captured by one single lower order polynomial that would be suitable for the use within a fast real-time capable MPC formulation. The characteristic map in Fig. 4.2 shows a steep ascent of the energy consumption per meter in the areas of low kinetic energy values and a large almost linear zone in the area of intermediate traction force and kinetic energy values. This shape is thus more suitable for a piecewise linear approximation.

Further advantages of this approximation type can be exploited if a *convex* piecewise linear approximation is chosen instead of a general piecewise linear function. Firstly, a

convex approximation of the map allows an overall convex problem formulation which provides the guarantee to find the global minimum solution of the problem. Secondly, a convex piecewise linear energy model can be included in an efficiently solvable linear or a quadratic program by making use of the so-called *separable programming* technique [132]. Lateron, this will be explained more in detail. At this point, the convex piecewise linear fitting of the energy consumption map is considered, first.

Finding a piecewise approximation offers the freedom to choose the number of approximating functions as well as the definition of the regions of the original map data to which each of these functions is fitted to.

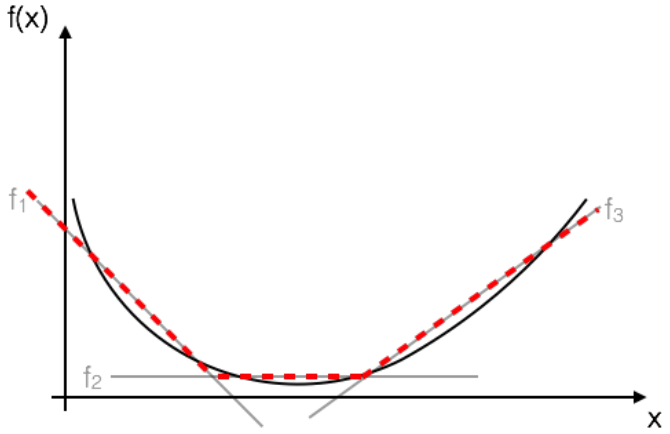
If the number of approximating functions is not limited, the best convex fit $y = [y_1, y_2, \ldots, y_n]$ of n given data points $\hat{y} = [\hat{y}_1, \hat{y}_2, \ldots, \hat{y}_n]$ at grid points $r = [r_1, r_2, \ldots, r_n]$ can be obtained by solving the following quadratic optimisation where the inequality constraints ensure the convexity and the vector $m = [m_1, m_2, \ldots, m_n]$ contains the slopes of the interpolating functions between the grid points [133]:

$$\begin{aligned} &\min_{y,m} \sum_{i=1}^{n} (\hat{y}_i - y_i)^2 \\ &\text{subject to:} \\ y_j \geq y_i + m_i \cdot (r_j - r_i) \quad &i, j = 1...n \end{aligned} \tag{4.10}$$

Solving (4.10) can generally result in defining different linear functions for each of the original data points since the number of approximating functions is not limited. A high number of approximating functions increases the accuracy of the fit. In the intended application however, this will also increase the number of constraints in the optimisation problem and thus the computational complexity to solve it. Thus, a compromise between the number of linear functions and the fitting accuracy must be aimed at and the solution of (4.10) is not suitable for the model design in practice.

However, finding the best convex piecewise linear approximation to given data points with a limited number of functions is a computationally hard combinatorial problem [134]. The heuristic algorithm presented in [134] is applied instead. Here, it is important

FIGURE 4.3: The function $f(x)$is approximated by a *convex* set of linear functions $f_1(x)$ to $f_3(x)$. This convex piecewise linear fitting can be seen as the maximum (or the *upper envelope*) of the set of linear functions as highlighted by the red dashed lines in the figure.



to note that a convex piecewise linear approximation can be represented by the maximum of a set of linear functions, as illustrated in Fig. 4.3 by the red dashed lines.

The heuristic algorithm in [134] starts with an initial set of linear functions $\mathcal{P}_{fun}$ and the data points to be fitted. Firstly, it is determined in which region of the data grid which function represents the maximum value of $\mathcal{P}_{fun}$. These grid points are then related to the corresponding function.

After this, each of the functions is fitted by using the least-squares algorithm to its assigned grid points (where it represented the maximum of $\mathcal{P}_{fun}$ in the previous step). This procedure is repeated iteratively. If the algorithm converges, it ends up with linear functions that are exactly fitted to those regions of the data grid where they represent the maximum of $\mathcal{P}_{fun}$, i.e. where they become relevant for the approximation. It is referred to [134] for further details on this algorithm.

To achieve a good fitting accuracy on the one hand and a tractable number of functions in the optimisation problem on the other hand, a set of six linear functions is chosen, here, since a higher number of approximating functions could not increase the fitting accuracy significantly.

In the case of the energy consumption map shown in Fig. 4.2, the approximating functions take the form:

$$P_i = a_i \cdot e_{kin} + b_i \cdot F_{trac} + c_i \quad i = 1...6 \tag{4.11}$$

An initial starting point for the fitting algorithm is obtained by assigning six regions of the energy consumption map manually and fitting one linear function to each of these regions with a least squares approximation. With this initial assignment, the above described algorithm can be used to improve the fit. The following procedure describes the methodology in the case of the energy consumption map.

1. Start with the characteristic map of the energy consumption per meter defined at the grid points $(e_{kin,i}, F_{trac,i})$ and an initial set of 6 approximating linear functions $\mathcal{P}_{fun} = \{P_1 \dots P_6\}; \quad P_i = a_i \cdot e_{kin} + b_i \cdot F_{trac} + c_i$, defined by the coefficients a_i, b_i and c_i.

2. Determine the data grid points $(e_{kin,i}, F_{trac,i})$ where each of the linear functions P_i yields the maximum function value of the set $\mathcal{P}_{fun}$ and assign these grid points to the related function.

3. Update the coefficients of the linear functions by least-squares fitting of the functions to those data points that have been assigned to it in the previous step.

4. **If** the updated coefficients are the same as those of the previous iteration, return a_i, b_i and c_i as the result of the converged algorithm (i. e. the linear are functions are fitted to those data points, where they represent the maximum of the set of functions $\mathcal{P}_{fun}$).

5. **Else:** go to step 2.

As a result of the converged algorithm, the piecewise linear fit presented in Fig. 4.5 is obtained.

The absolute fitting error between the original data and the convex piecewise linear fit has been evaluated and given in Fig. 4.6. According to Fig. 4.6, the highest fitting errors are given at very low driving speeds and intermediate positive traction forces F_{trac}. As visible in Fig. 4.4, this results from the very high consumption values in these areas of the map. This cannot be solved by introducing an additional linear function, since the

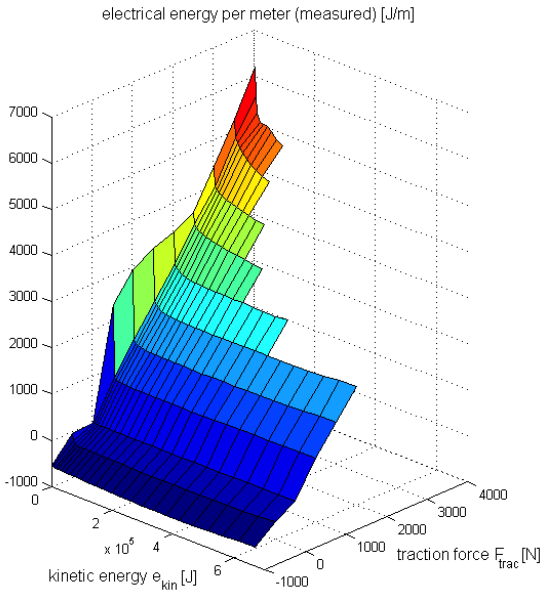


FIGURE 4.4: The reformulated measured energy consumption per meter characteristics are repeated here for the sake of comparison.

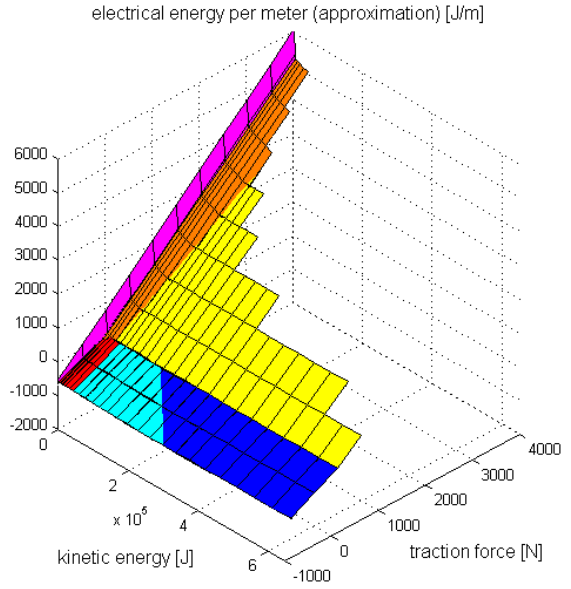
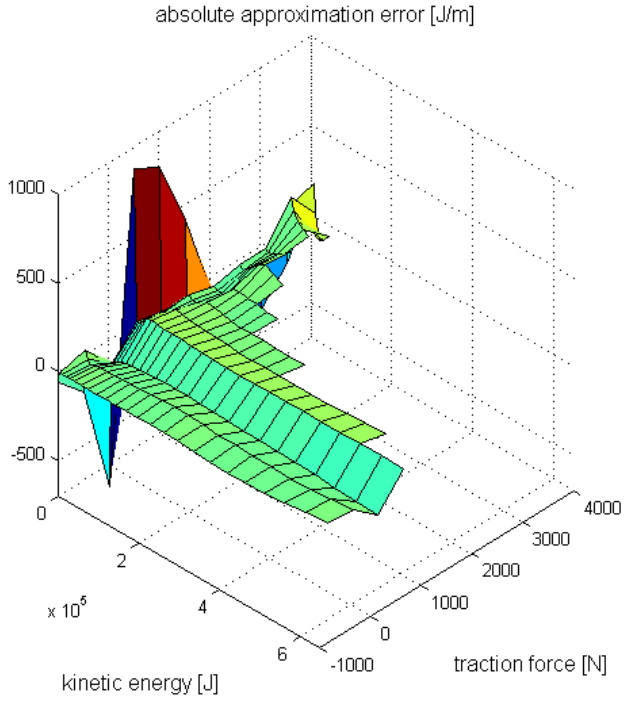


FIGURE 4.5: Convex piecewise linear fit of the characteristic energy consumption per meter of the *Smart ED*

FIGURE 4.6: Absolute fitting error between the measured energy consumption per meter of the *Smart ED* (cf. Fig. 4.4) and the convex piecewise linear approximation (cf. Fig. 4.5) in [J/m].



high error levels are only present at intermediate positive traction forces and not in the full range of forces.

However, since the proposed controller is mainly intended for the use in overland and highway driving, the prediction accuracy at these extremely low speeds is of minor importance. In all other regions, the absolute and percental fitting error is very low and a realistic prediction can be expected from the piecewise linear model. Thus, the presented fit is suitable to be used in the prediction model of the MPC controller.

The optimisation problem of the MPC controller (as discussed later in Section 4.3) is then set up in a way that the maximum of the constraint functions (P_1 to P_6) is determined implicitly and guarantees that the dummy decision variable $u_{cons}(s)$ represents the energy consumption per meter of the vehicle according to the following piecewise linear model:

$$\begin{aligned} u_{cons} &= a_i \cdot e_{kin} + b_i \cdot F_{trac} + c_i \quad \text{if } P_i \text{ is active} \\ u_{cons} &\geq a_j \cdot e_{kin} + b_j \cdot F_{trac} + c_i \quad \text{for } j \neq i \end{aligned} \tag{4.12}$$

This technique is called *separable programming* and explained in more details in Section 4.3.1.

In other words, it is realised that one of the inequality constraints (4.11) is always active depending on the actual operating point (specified by the driving speed v and the traction force F_{trac}). Given this information, the energy consumption of the vehicle $E_{batt}(s)$ can simply be modeled by integrating the dummy decision variable $u_{cons}(s)$ (representing the charge consumption per meter) with respect to the position.

$$\frac{dE_{batt}}{ds} = u_{cons}(s) \tag{4.13}$$

The continuous dynamic motion model (4.7) and the model of the energy consumption (4.13) can be summarised in state space form and discretised straightforwardly by assuming zero-order hold for the input and the disturbance. The following discrete linear

state space representation (with the position-variant system matrix $A_d(k+i|k)$ depending on the curve radius r_c) is obtained for $k \in \{0....N-1\}$, where the symbol "ˆ" indicates the discrete counterparts of the continuous variables:

$$\underbrace{\begin{bmatrix} \hat{e}_{kin}(k+i+1|k) \\ \hat{E}_{batt}(k+i+1|k) \end{bmatrix}}_{\hat{x}(k+i+1|k)} = \underbrace{\begin{bmatrix} a_{11,i} & 0 \\ 0 & 1 \end{bmatrix}}_{A_d(k+i|k)} \cdot \underbrace{\begin{bmatrix} \hat{e}_{kin}(k+i|k) \\ \hat{E}_{batt}(k+i|k) \end{bmatrix}}_{\hat{x}(k+i|k)} + \underbrace{\begin{bmatrix} b_{11} & 0 \\ 0 & b_{22} \end{bmatrix}}_{B_d} \cdot \underbrace{\begin{bmatrix} \hat{F}_{trac}(k+i|k) \\ \hat{u}_{cons}(k+i|k) \end{bmatrix}}_{\hat{u}(k+i|k)} + \underbrace{\begin{bmatrix} e_{11} & e_{12} \\ 0 & 0 \end{bmatrix}}_{E_d} \cdot \underbrace{\begin{bmatrix} \sin(\hat{\alpha}_{sl}(k+i|k)) \\ \cos(\hat{\alpha}_{sl}(k+i|k)) \end{bmatrix}}_{\hat{d}(k+i|k)} \tag{4.14}$$

This discrete linear state space model together with the inequalities (4.12) can now be included in a quadratic optimal control problem with linear constraints that can efficiently be solved by *quadratic programming* (QP) methods. An MPC formulation based on a (convex) quadratic program can be expected to be running in real-time and is much more suitable for the intended safety-critical real-time application than a general non-linear optimisation.

The parameter values of the discrete controller prediction model for a discretisation step size of 10 m are summarised in Tab. 4.1.

4.2 Additional Constraints on the Optimisation Problem

Since (4.7) is only valid for positive kinetic energy values, the inequality constraint

$$\hat{e}_{kin}(k+i|k) \geq 0 \tag{4.15}$$

is added to the optimisation problem.

To stay within the limitations of the vehicle drive train, further inequality constraints need to be included in the problem formulation. The control input of the prediction model (4.14) is the traction force at the wheel F_{trac}. The maximum traction force at a fully pressed accelerator pedal but without using the kick-down / boosting ability of the

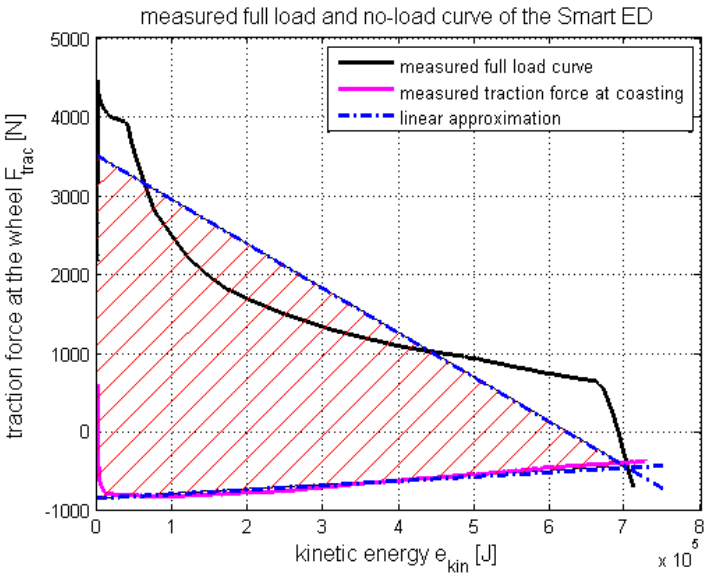


FIGURE 4.7: Measured traction force of the *Smart ED* at full load (100 % accelerator position without kick-down / boosting, black line) and coasting (pedals released, pink line). The linear approximation is given in blue. The hatched area is the feasible region of the traction force F_{trac}.

car depends on the driving speed v (and thus on the kinetic energy e_{kin} of the moving car) and has been measured at the dynamometer test bench.

Since an actuation of the brake pedal is not considered in this control approach, the minimum traction force at the wheel is achieved at coasting when the accelerator pedal is released. As already mentioned before, the *Smart ED* recovers electrical energy in this state which leads to negative traction forces. The minimum traction force has again been measured at the dynamometer test bench and both, the minimum and the maximum force at the wheel are plotted in Fig. 4.7.

The maximum traction force (black line in the figure) is determined by the maximum torque of the synchronous machine. At low rotational speeds (i. e. at low driving speeds), the electric motor runs in *nominal* operation. Here, the maximum torque is limited by the maximum current that the coils of the rotor can bear. At higher rotational speeds, the induced voltage in the coils would increase to values above the supply voltage such that an increase of the rotational speed is only possible if the excitation field is weakened (*field weakening operation*). This leads to the decrease in the motor torque and hence the maximum traction force at the wheel decreases at higher kinetic energy values of the moving car according to the full-load curve in Fig. 4.7.

The measured minimum and maximum traction force limits are non-linear with the kinetic energy of the moving car. However, since the prediction model described in Section 4.1 could be reformulated in a linear form, it is desirable to keep the inequality constraints linear as well to enable the use of a quadratic problem formulation with linear constraints.

Hence, the limits are linearised using a least-squares fit of the measured data. The linearised constraints are plotted in blue in Fig. 4.7 and take the form:

$$g_1 \cdot e_{kin} + g_2 \leq F_{trac} \leq g_3 \cdot e_{kin} + g_4; \quad e_{kin} \geq 0 \tag{4.16}$$

The measured minimum traction force can be captured by the linearised constraint with a high accuracy. The upper limit however cannot fully be represented by a linear function. In the region of medium kinetic energies where the linear constraint allows higher traction forces than the measured full-load curve, it is possible that the traction force planned by the MPC is above maximum realisable traction force in the car. This is considered later in the closed-loop simulation and the experimental implementation by saturating the output of the controller according to the measured full-load curve to represent the behaviour of the real car. The effect of this saturation might lead to a mismatch between the predicted and the actual vehicle behaviour but can be compensated by the controller due to the feedback. Furthermore, since the controller will be used for energy-efficient driving, very high traction forces close to the upper limit can be expected to be avoided to exclude the associated high energy consumption. Thus, the constraint mismatch at high traction forces will rarely take effect in the control operation.

The inequality constraints on the problem determine if the problem may turn infeasible, i. e. that no solution exists that fulfills all constraints. This is critical since then, no control action is determined, in this case.

In this application case, the constraints are chosen such that this can hardly happen. All constraints on the inputs ($\hat{F}_{trac}$, $\hat{u}_{cons}$) can always be fulfilled by the optimiser by choosing these decision variables appropriately. Apart from these, only the state kinetic energy e_{kin} is constrained to take values between zero and the kinetic energy related to the maximum driving speed of 125 km/h according to Fig. 4.7. Theoretically, it

TABLE 4.1: Parameters of the controller prediction model.

par.	**value**	**par.**	**value**	**par.**	**value**
a_1	-0.202	a_2	-0.00447	a_3	6.146E-5
a_4	-9.76E-4	a_5	-2.256E-4	a_6	1.138E-4
b_1	1.318	b_2	1.276	b_3	1.196
b_4	0.674	b_5	0.676	b_6	0.728
c_1	1060.05	c_2	155.52	c_3	46.83
c_4	114.95	c_5	87.91	c_6	26.86
$0.97 \leq a_{11,i} \leq 0.99$		b_{11}	9.9595	b_{22}	10
e_{11}	9.5218e4	e_{12}	952.17	g_1	5.538E-4
g_2	-841.1	g_3	-0.0056	g_4	3505

could happen that the vehicle is running in a down-slope such that it accelerates above 125 km/h and regenerative braking is not enough to keep this limit. In practice, it is almost impossible that these limits cannot be kept by the actuators and the cruise control system should not be used in such extreme cases.

4.3 Cost Function

Generally, the cost function of an optimal control problem consists of terms related to the states at the end of the prediction horizon $x(s_{end})$ (the so-called *Mayer term*) as well as terms related to the integral (or the sum in the discrete case) of the states x and inputs u throughout the horizon (the so-called *Lagrange term*) [135].

$$\min_{u(s)} \quad \underbrace{f_{end}\big(x(s_{end}), s_{end}\big)}_{\text{Mayer term}} + \int_{s_0}^{s_{end}} \underbrace{f_{int}\big(x(s), u(s), s\big)}_{\text{Lagrange term}} \cdot ds \tag{4.17}$$

Together with the equality and inequality constraints on the problem, the cost function determines whether the optimal control problem is linear, quadratic or generally non-linear and thus which effort is necessary to solve it. Since the prediction model (4.14) has been formulated in a way that allows the use of only linear constraints, it is desirable to use a linear or a quadratic cost function to achieve a linear or a quadratic optimisation problem that can efficiently be solved. MPC formulations with linear cost function and linear constraints have shown undesirable non-smooth control behaviour [109] and thus, a (discrete) quadratic cost function is chosen:

$$\min_{\hat{u}(k+i|k)} \quad ||\hat{x}(k+N_p|k) - \hat{x}_{ref}(k+N_p|k)||_P^2 + \sum_{i=0}^{N_p-1} ||\hat{x}(k+i|k) - \hat{x}_{ref}(k+i|k)||_Q^2 + ||\hat{u}(k+i|k) - \hat{u}_{ref}(k+i|k)||_R^2 \tag{4.18}$$

The cost function defines the aim of the control. This can be bringing the states to a static set-point or reference tracking in combination with minimising the control effort. Furthermore, the cost function terms as well as the length of the prediction horizon determine whether the control loop is stable [13, 110].

In the specific case here, the cost function must be chosen such that the piecewise linear modelling approach of the energy consumption map (Fig. 4.4 and (4.13)) according to the technique of *separable programming* is valid.

4.3.1 The Method of Separable Programming

Generally, the application of separable programming aims at converting a non-linear program to a linear program by approximating the non-linear functions by (in the standard case convex) piecewise linear functions [132].

These linear functions are included as inequality constraints in the new optimisation problem and form a piecewise linear boundary of the feasible region which is approximating the original non-linear function. Then, a new dummy decision variable is introduced that is minimised subject to these inequalities. Its optimal value will consequently lie on the boundary of the feasible region which is approximation of the original function. This in illustrated in Fig. 4.8.

In the special case of the given application the energy consumption per meter is approximated by the linear inequalities (4.12). The additional dummy variable representing the energy consumption per meter is u_{cons}.

The application of separable programming requires

- the objective function and the constraints of the resulting linear problem to be *separable* and

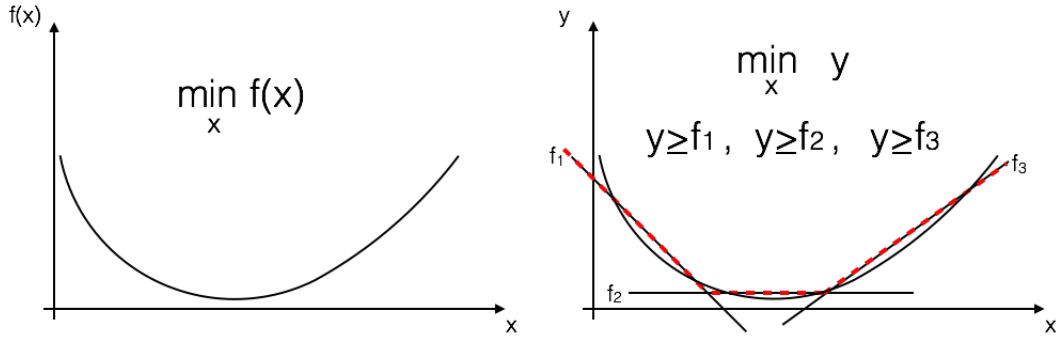


FIGURE 4.8: The principle of separable programming. The original non-linear optimisation problem on the left is replaced by the linear optimisation problem on the right of the Figure. A new dummy decision variable y is introduced which is minimised subject to linear constraints that approximate the original function. It is important, that eventually other cost function terms do not depend on y (i. e. they must be separable).

- the decision variables approximating the original functions in the problem to be bounded.

If these properties are fulfilled, the (convex) piecewise linear model of the originally non-linear function takes effect.

Here, a quadratic objective is intended to be used to achieve a smoother behaviour of the controls. In this case, the prerequisites for the application of separable programming must be generalised and adapted according to the following proposition.

Proposition 1. Consider an optimisation problem with the following properties:

- Property 1: The cost function summands $f(U)$ and $g(W)$ are separable with regard to the vector optimisation variables U and W with the elements $U(l), W(l)$, where $l \in \{0...N-1\}$. In other words, $g(W)$ does not depend on U:

  $$\min_{U,W} \quad f(U) + g(W)$$

- Property 2: The decision variable U is subject to convex lower bounds. Here, it is enough to ensure that U is bounded below since linear constraint sets can only form a convex bound.

  $$\mathcal{P}: \quad A \cdot \begin{bmatrix} U \\ W \end{bmatrix} \geq b$$

- Property 3: $f(U)$ is strictly monotonously decreasing for all $U(l), \quad l \in \{0...N-1\}$ in the feasible region: $f(U_1) < f(U_2) \quad \forall U_1(l), U_2(l) : U_1(l) < U_2(l)$

Let $\{U^*, W^*\}$ be the solution of this optimisation problem. Then, U^* will lie on the boundary of the feasible region, i.e. at least one of the inequality constraints will be equally fulfilled /active.

Proof. Suppose, for the sake of contradiction, that for a feasible pair $\{U, W\}$ no inequality constraint is active, i.e.

$$A \cdot \begin{bmatrix} U \\ W \end{bmatrix} > b.$$

Then, the value of the cost function could be reduced because there exists a vector $U' \neq U$ where $U'(l) \leq U(l) \quad \forall l \in \{0...N-1\}$ such that $f(U') + g(W) < f(U) + g(W)$ and

$$A \cdot \begin{bmatrix} U' \\ W \end{bmatrix} = b.$$

This follows from Property 3 and the fact that the constraint set on U represents a lower bound and is convex. Thus, the pair $\{U, W\}$ cannot be the minimal solution. Hence, the solution must always lie on the boundary of the feasible set and at least one inequality constraint $P_i \in \mathcal{P}$ must be equally fulfilled. □

Consequently, the cost function of the predictive energy-saving cruise controller must be designed such that Properties 1 - 3 of Proposition 1 are fulfilled. In the following, several cost function candidates are presented that fulfill the requirements of separable programming and will later be investigated and compared in closed-loop simulations regarding the control performance.

4.3.2 Basic Cost Function Enabling Energy-Efficient Speed Reference Tracking

The main goal of the intended control approach is energy-efficient driving and hence, the energy consumption must be considered in the cost function of the MPC controller. However, considering only the energy consumption would lead to zero or very low driving speeds at low traction forces to decrease the driving resistance forces. Thus, the trip time must be considered as well in the cost function or the constraints of the optimal control problem to achieve a reasonable trade-off.

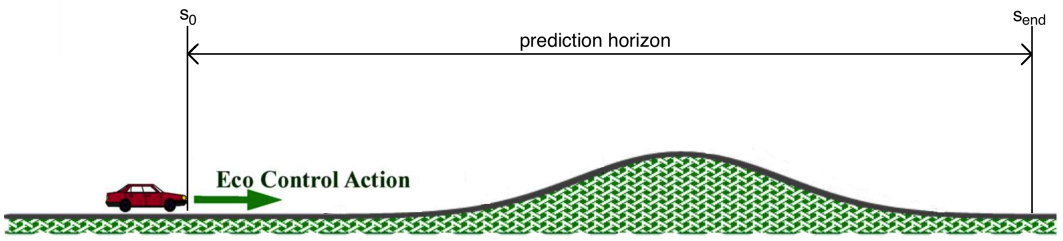


FIGURE 4.9: Exemplary driving situation with the vehicle in front of an up-slope within the prediction horizon. Figure edited from [12]

The direct inclusion of the trip time in the cost function has been used in several related works to achieve a compromise between driving fast and energy-efficient e. g. [136]. This simple methodology is not sufficient for the use in this work because driving at a compromise between speed and energy consumption does not respect speed limits, the desired speed of the driver, the maximum allowable speeds in curves or keeping the distance to preceding cars. Hence, a reference speed or a maximum speed depending on the position on the route should be included, here. Since the prediction model (4.14) describes the kinetic energy e_{kin} as a measure of the driving speed, the speed reference must be converted into a kinetic energy reference to comply with the variables available.

Assuming this kinetic energy reference to be known, the quadratic tracking error can directly be included in the Lagrange term (i. e. the integral / summed up term) of the cost function to penalise deviations from the reference at each prediction step.

The energy consumption can either be included in the Mayer or the Lagrange term - with different implications for the closed-loop control behaviour. To illustrate this, the driving situation where the vehicle shall overcome a small hill within the prediction horizon is considered. The reference speed is assumed to be constant and the terrain profile starts with flat road, followed by an up-slope and a subsequent down-slope, see Fig. 4.9.

Depending on the characteristics of the power train, especially when higher traction forces lead to a disproportionately higher energy consumption, it can be beneficial to speed up in front of the hill and then overcome the slope using the momentum while driving at lower traction forces.

The inclusion of the accumulated energy consumption $\sum_{i=0}^{N_p} \hat{E}_{batt}(k+i|k)$ in the Lagrange term penalises the consumption at each step of the prediction. Hence, a speeding up in front of the hill may be suppressed in order to keep the accumulated consumption over

all prediction steps low. This can lead to a higher consumption at the end of the horizon / the trip.

In contrast, including the terminal energy consumption $\hat{E}_{batt}(k+N_p|k)$ as a Mayer term (resulting from the sum of the power consumption throughout the horizon) leads to an "intelligent" predictive controller behaviour with the freedom to increase the consumption at any position of the prediction horizon if there is the benefit to save energy later at the end of the horizon as a result of this anticipatory action. Thus, this methodology is chosen, here.

In conclusion, the simplest cost function for an energy-saving cruise controller that considers a given kinetic energy reference would be the sum of the energy consumption at the end of the prediction horizon and the accumulated squared kinetic energy tracking error. To enforce driving speeds close to the reference at the end of the prediction horizon, a penalty term considering the terminal kinetic energy tracking error is included additionally. In the discrete case, this yields:

$$\min_{\hat{F}_{trac}(k+i|k),\hat{u}_{cons}(k+i|k)} \quad l_{e,b}\cdot\hat{E}_{batt}(k+N_p|k)+q_{e,b}\cdot\left(\hat{e}_{kin}(k+N_p|k)-\hat{e}_{kin,ref}(k+N_p|k)\right)^2 + \sum_{i=0}^{N_p-1} q_{s,b}\cdot\left(\hat{e}_{kin}(k+i|k)-\hat{e}_{kin,ref}(k+i|k)\right)^2 \tag{4.19}$$

When minimising (4.19), the weightings $l_{e,b}$, $q_{e,b}$ and $q_{s,b}$ determine how closely the kinetic energy reference $\hat{e}_{kin,ref}$ is tracked and to which extent deviations are allowed in order to save energy. This directly leads to the question, which combination of weightings leads to a desirable compromise between (speed) reference tracking and energy savings. This however depends on the specific driving situation and on the preferences of the user and will be investigated in simulations.

Finally, it is checked whether the cost function (4.19) is suitable for the separable programming approach of modelling the energy consumption as mentioned before.

Corollary 1. The cost function (4.19) fulfills the prerequisites of Proposition 1 and hence, the piecewise linear energy consumption model according to (4.12) takes effect.

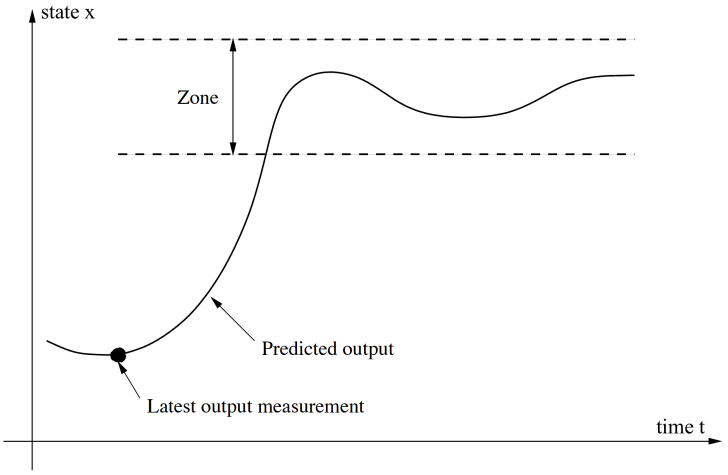


FIGURE 4.10: Exemplary *zone objective*. Constraints on the outputs specify an admissible zone for the output trajectory, edited from [13]

Proof. The proof consists of showing that Properties 1-3 of Proposition 1 are fulfilled.

Property 1 is fulfilled for the assignment $f(U) = l_{e,b} \cdot \hat{E}_{batt}(k + N_p|k)$ where $U(l) = \hat{u}_{cons}(k + i|k)$ since none of the other cost function terms depends on the variables $\hat{u}_{cons}(k + i|k)$ and vice versa.

Property 2 is respected since the decision variables $\hat{u}_{cons}(k + i|k)$ are subject to the convex lower bounds (4.12).

Property 3 is finally fulfilled since $f(U) = l_{e,b} \cdot \hat{E}_{batt}(k+N_p|k)$ is just the weighted sum of the decision variables $\hat{u}_{cons}(k + i|k)$ according to (4.14) and thus strictly monotonously decreasing with each of the decision variables.

Consequently, the requirements of the separable programming approach are fulfilled. □

4.3.3 Zone Objective Using Soft Constraints

Another approach is to use a cost function that only aims at the minimisation of the energy consumption. However, as already mentioned, the driving speed must be reflected somehow in the optimisation problem which can then be realised by imposing constraints on the minimum and maximum kinetic energy of the moving car. This is equivalent to specifying a *zone* for the admissible driving speed, see also Fig. 4.10 [13].

Imposing constraints on the driving speed can render the optimisation problem infeasible. This could happen for instance when the reference changes faster than the vehicle

can follow or when the actual initial measurement of the state already lies beyond the zone, as illustrated in Fig. 4.10. Because of this, the constraints of the zone objective should be implemented as *soft constraints*.

Soft constraints include a *slack variable* $\hat{s}_l$ that allows violations of the respective inequality constraint. However, these violations are highly penalised in the cost function while the penalty terms equal zero, when the inequality constraints are fulfilled. In the specific case of constraints on the kinetic energy of the moving car, the soft constraints take the form:

$$\hat{e}_{kin,min}(k+i|k) + \hat{s}_{l,1}(k+i|k) \leq \hat{e}_{kin}(k+i|k) \leq \hat{e}_{kin,max} + \hat{s}_{l,2}(k+i|k) \tag{4.20}$$

The squares of the slack variables $\hat{s}_{l,1}(k+i|k)^2$ and $\hat{s}_{l,2}(k+i|k)^2$ are included in the cost functions with comparatively high weightings to enforce the slack variables to be zero whenever it is possible and only allow non-zero values if the problem would be infeasible otherwise.

In the case of using a zone specified by soft constraints to include the kinetic energy reference, the cost functions only includes the terminal energy consumption and the slack variables and takes the form:

$$\min_{\hat{F}_{trac}(k+i|k),\hat{u}_{cons}(k+i|k),\hat{s}_l(i|k)} \quad l_{e,z} \cdot \hat{E}_{batt}(k+N_p|k) + \sum_{i=0}^{N_p} q_{s,z} \cdot \left(\hat{s}_l(k+i|k)\right)^2 \tag{4.21}$$

Minimising (4.21) subject to (4.20) enforces the kinetic energy to be between the lower and upper limit. As long this is fulfilled (and consequently $\hat{s}_l(k+i|k)^2 = 0$), the electrical energy consumption is minimised within these limits without compromising.

The advantage here is that the control designer can clearly define how much deviation from the desired speed is allowed in order to save energy by defining the zone limits accordingly. This can avoid unsatisfactory deviations for the user. Further, the trade-off question regarding weightings on the kinetic energy reference tracking and the electrical energy consumption does not arise in this case.

A disadvantage of the zone objective is the fact that the weighting on the slack variables in the zone constraints must be very high to enforce the constraints to be fulfilled. Disproportionate weighting factors can cause an ill-conditioned / unbalanced optimisation problem that requires more time for the numerical solution on-line.

The following investigation shows that the zone objective (4.21) complies with the separable programming approach as proposed above. The same reasoning as in Corollary 1 can be applied here.

Corollary 2. Cost function (4.21) fulfills the prerequisites of Proposition 1 and hence, the piecewise linear energy consumption model according to (4.12) takes effect.

Proof. The proof again consists of showing that Properties 1-3 of Proposition 1 are fulfilled.

Property 1 is fulfilled for the assignment $f(U) = l_{e,z} \cdot \hat{E}_{batt}(k+N_p|k)$ where $U(l) = \hat{u}_{cons}(k+i|k)$ since none of the other cost function terms depends on the variables $\hat{u}_{cons}(k+i|k)$ and vice versa.

Property 2 is respected since the decision variables $\hat{u}_{cons}(k+i|k)$ are subject to the convex lower bounds (4.12).

Property 3 is finally fulfilled since $f(U) = l_{e,z} \cdot \hat{E}_{batt}(k+N_p|k)$ is just the sum of the decision variables $\hat{u}_{cons}(k+i|k)$ according to (4.14) and thus strictly monotonously decreasing with each of the decision variables.

Consequently, the requirements of the separable programming approach are fulfilled. □

4.3.4 Cost Function Ensuring Stability of the Closed-Loop Control

The design of the two aforementioned cost function candidates has purely been driven by the intended goals of the control approach but they do not provide stability guarantees for the closed loop. In the following, the classical way of designing a stable MPC is presented. The approach is then adapted to the energy-saving cruise controller afterwards.

The key of a stability proof in MPC is to show that the cost function value decreases from MPC sample step to sample step and that a settling of the system states to their

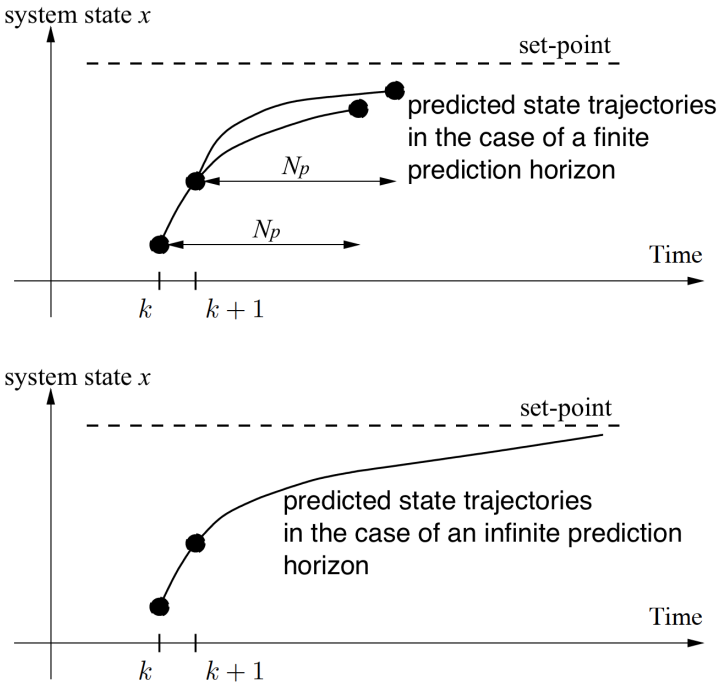


FIGURE 4.11: The upper half of the image shows that the optimisation problem changes from MPC sample step k to step $k+1$ in the case of a finite prediction horizon since new information enters the problem from step to step. In contrast, this is not the case when using infinite horizons (bottom half of the picture). Here, the new optimisation problem is the "*tail*" of the problem of the previous step according to Bellmann's principle of optimality. Figure edited from [13]

set-point is associated with this decrease. Then, the cost function can be used as a *Lyapunov function* of the controller. [13, 110]

One way of achieving this is to incorporate an infinite prediction and control horizon which implies that in the case of a perfect prediction model and the absence of disturbances, the predicted trajectories of the subsequent optimisation problem are a part of the predicted trajectories (Bellmann's principle of optimality) at the previous sample instant as the initial state approaches the set-point. This ensures the cost associated with the tracking error to decrease from sample step to sample step. This is illustrated in Fig. 4.11. [13]

In order to be able to solve an optimisation problem with inequality constraints and an infinite prediction horizon, the cost function must be reformulated. It is started with a general formulation where the system state $\hat{x}$ shall be driven to zero and the inputs $\hat{u}$ and also input moves $\Delta\hat{u}$ are penalised. Note that the sum here extends to an infinite number of steps. [13]

$$\min_{\hat{u}} \sum_{i=1}^{\infty} \left(||\hat{x}(k+i|k)||_Q^2 + ||\Delta\hat{u}(k+i-1|k)||_S^2 + ||\hat{u}(k+i-1|k)||_R^2 \right) \tag{4.22}$$

Herein, it is assumed that only the first N_u control moves are non-zero since the states must reach the origin in N_u steps in order to prevent an infinite cost. It should be noted that the control horizon must be chosen accordingly long to make this possible. [13]

In the sequel, it is assumed that $R > 0$, $S \geq 0$ and $Q \geq 0$, the plant is stable, the plant model fits the real plant without mismatch, full state measurement is given, there are no disturbances, the optimisation problem is always feasible and the true optimum is obtained at each sample of the MPC. [13]

If the required value of the input $\hat{u}$ to keep the states $\hat{x}$ at the origin is zero, it follows from these assumptions that $\hat{u}(k+i|k) = 0$ for $i \geq N_u$ in order to prevent an infinite cost. The cost function can then be reformulated as: [13]

$$\min_{\hat{u}} \sum_{i=1}^{\infty} ||\hat{x}(k+i|k)||_Q^2 + \sum_{i=1}^{N_u} \left(||\Delta\hat{u}(k+i-1|k)||_S^2 + ||\hat{u}(k+i-1|k)||_R^2 \right) \tag{4.23}$$

It is assumed that the prediction model is perfect and there are no disturbances. Hence, the predicted states will exactly match the measured states and it can be concluded that the optimal controls $\hat{u}$ remain the same for the future sample steps. Due to the infinite horizon formulation, the optimal cost function value $J^o(k+1)$ at MPC sample step $(k+1)$ can then be computed from the optimal value $J^o(k)$ at step k by subtracting the summands related to the very first step k: [13]

$$J^o(k+1) = J^o(k) - ||\hat{x}^o(k+1)||_Q^2 - ||\Delta\hat{u}^o(k|k)||_S^2 - ||\hat{u}^o(k|k)||_R^2 \tag{4.24}$$

Because of the assumptions on the weightings Q, S and R, the sum of the subtracted terms is positive and hence it follows that $J^0(k+1) < J^0(k)$ and $x^0(k+1) < x^0(k)$ such that J^o is a *Lyapunov function* of the closed loop which ensures stability. [13]

However, to enable the solution of the constrained predictive control problem based on cost function (4.23), the problem must be reformulated. The cost function can be split up and rewritten as: [13]

$$\begin{aligned} &\min_{\hat{u}} \sum_{i=N_u+1}^{\infty} ||\hat{x}(k+i-1|k)||_Q^2 \\ &+ \sum_{i=1}^{N_u} \left(||\hat{x}(k+i-1|k)||_Q^2 + ||\Delta\hat{u}(k+i-1|k)||_S^2 + ||\hat{u}(k+i-1|k)||_R^2 \right) \end{aligned} \tag{4.25}$$

Since the control input $\hat{u}(k+i) = 0$ for $i \geq N_u$ to avoid an infinite cost, the first term of (4.25) can be rewritten according to the state propagation $\hat{x}(k+i+1|k) = A_d \cdot \hat{x}(k+i|k)$: [13]

$$\sum_{i=N_u+1}^{\infty} ||\hat{x}(k+i|k)||_Q^2 = \hat{x}(k+N_u|k)^T \cdot \underbrace{\left[\sum_{i=0}^{\infty} \left(A^T\right)^i Q A^i \right]}_{\bar{Q}} \cdot \hat{x}(k+N_u|k) \tag{4.26}$$

If the plant is stable, the series $\sum_{i=0}^{\infty} \left(A^T\right)^i Q A^i = \bar{Q}$ will converge and can be computed by solving the *matrix Lyapunov equation*. Then, the cost function can be written as follows: [13]

$$\begin{aligned} &\min_{\hat{u}} \quad \hat{x}(k+N_u|k)^T \cdot \bar{Q} \cdot \hat{x}(k+N_u|k) \\ &+ \sum_{i=1}^{N_u} \left(||\hat{x}(k+i-1|k)||_Q^2 + ||\Delta\hat{u}(k+i-1|k)||_S^2 + ||\hat{u}(k+i-1|k)||_R^2 \right) \end{aligned} \tag{4.27}$$

This formulation can finally be solved as a finite dimensional constrained quadratic optimisation problem to obtain a predictive controller with guaranteed stability.

These results can be adapted to the given energy-saving cruise controller. The above described methodology to prove the stability is valid for a set-point equal to zero (at the origin of the state space). For non-zero reference values, the terms $||\hat{x}(k+i-1|k)||_Q^2$ need to be replaced by $||\hat{x}(k+i-1|k) - \hat{x}_{ref}(k+i-1|k)||_Q^2$ and also $||\hat{u}(k+i-1|k)||_R^2$

must be replaced by $||\hat{u}(k+i-1|k) - \hat{u}_{ref}(k+i-1|k)||_R^2$ to achieve a bounded cost. Herein, $\hat{u}_{ref}$ is the steady input value that keeps the state at x_{ref}. However, if $R = 0$, this is not required. [13]

In the case of the regarded predictive cruise controller, the kinetic energy reference (i. e. the desired speed) varies with the position on the track. However the reference can be assumed to take a fix constant value for $i \geq N_u$ (corresponding to a nominal speed) to overcome this issue.

Furthermore, the stability of the plant itself (i. e. the motion and energy consumption dynamics) must be tested. This can easily be done by computing the eigenvalues of the system matrix A_d in (4.14) with $a_{11} = 0.99$ (the maximum value). The eigenvalues of A_d with $a_{11} = 0.99$ are 0.99 and 1. An eigenvalue of 1 indicates a state at the boundary of stability and refers to the battery energy consumption $\hat{E}_{batt}(i+1|k) = \hat{E}_{batt}(i|k) + \hat{u}_{cons}(i+1|k)$.

Consequently, the plant can then not be considered as stable in the classical sense and moreover, the series $\bar{Q}$ does not converge to a finite value in the methodology to prove the stability as sketched above.

The classical methodology to treat such a case would be to decompose the system matrix into its stable and unstable modes and include equality constraints into the problem formulation that force all unstable modes to be controlled to zero (or to the related set-points, respectively) at the end of the control horizon N_u. Then, the unstable modes do not contribute to the cost function terms where $i > N_u$ and can be excluded from the series $\bar{Q}$. [13]

The very same method cannot be applied here, since the energy consumption is not a system state to be controlled to a certain set-point. Rather, the energy consumption $\hat{E}_{batt}$ directly results from the system state kinetic energy $\hat{e}_{kin}$ and the control input traction force $\hat{F}_{trac}$ according to the piecewise linear model (4.12) and is not an independently controlled system state to be regulated.

Here, the system can be considered as stable when the kinetic energy of the moving car (and thus the driving speed) settles to a constant value for $i \to \infty$. The electrical energy consumption $\hat{E}_{batt}$ (being the unstable mode in the given plant system) is thus excluded

from the cost function by assigning a zero weighting for $i > N_u$ (hence only the driving speed is considered to compute the weighting on the terminal cost).

When summarising the aforementioned points, a cost function for the energy-efficient predictive cruise controller can take the following form:

$$\begin{aligned}\min_{\hat{F}_{trac}(k+i|k),\hat{u}_{cons}(k+i|k)} \quad & l_{e,s} \cdot \hat{E}_{batt}(k+N_u) + \bar{q}_{e,s} \cdot \left(\hat{e}_{kin}(k+N_u|k) - \hat{e}_{kin,ref}(k+N_u|k)\right)^2 \\ & + \sum_{i=0}^{N_u-1} \Big(q_{s,s} \cdot \left(\hat{e}_{kin}(k+i|k) - \hat{e}_{kin,ref}(k+i|k)\right)^2 \\ & + r_{s,s} \cdot \left(\hat{F}_{trac}(k+i|k) - \hat{F}_{trac,ref}(k+i|k)\right)^2 \Big)\end{aligned} \tag{4.28}$$

Herein, $\bar{Q} = \begin{bmatrix} \bar{q}_{e,s} & 0 \\ 0 & 0 \end{bmatrix}$ is computed as the solution of the matrix Lyapunov equation:

$$\begin{aligned} A_d^T \bar{Q} A_d - \bar{Q} - Q = 0 \\ \text{where} \quad Q = \begin{bmatrix} q_{s,s} & 0 \\ 0 & 0 \end{bmatrix} \\ \text{and} \quad \bar{Q} = \begin{bmatrix} \bar{q}_{e,s} & 0 \\ 0 & 0 \end{bmatrix} \end{aligned} \tag{4.29}$$

As already mentioned, the energy consumption at the end of the control horizon $\hat{E}_{batt}(k+N_u|k)$ is not intended to be driven to a certain reference point, is excluded from the stability consideration and thus can be included as a linear term in the cost function without raising stability issues. Furthermore, the control horizon must be long enough to enable the system state to reach the reference. If there are no extreme up- or down-slopes on the road, this is given for the chosen horizon of 40 steps of 10 m.

In addition to the consideration of stability, the modelling approach of *separable programming* must be kept in mind and is reflected in the subsequent corollary.

Corollary 3. The cost function (4.28) fulfills the prerequisites of Proposition 1 and hence, the piecewise linear energy consumption model according to (4.12) takes effect.

Proof. The proof again consists of showing that the Properties 1-3 of Proposition 1 are fulfilled.

Property 1 is fulfilled for the assignment $f(U) = l_{e,s} \cdot \hat{E}_{batt}(k + N_p|k)$ where $U(l) = \hat{u}_{cons}(k + i|k)$ since none of the other cost function terms depends on the variables $\hat{u}_{cons}(k + i|k)$ and vice versa.

Property 2 is respected since the decision variables $\hat{u}_{cons}(k + i|k)$ are subject to the convex lower bounds (4.12).

Property 3 is finally fulfilled since $f(U) = l_{e,s} \cdot \hat{E}_{batt}(k + N_p|k)$ is just the sum of the decision variables $\hat{u}_{cons}(k + i|k)$ according to (4.14) and thus strictly monotonously decreasing with each of the decision variables.

Consequently, the requirements of the separable programming approach are fulfilled. □

Finally, it can be pointed out that it is possible to formulate the OCP in a way that guarantees the stability of the control loop in the case of a steady operating point and constant reference and disturbance. However, it must be noted that it is not the aim of the developed controller to regulate the system states to a steady state. Rather, the main aim of the eco-cruise controller is to plan a traction force trajectory that reflects a good compromise between driving close to the desired speed and energy saving with a varying speed reference. Therefore, a controller formulation without guaranteed nominal stability is acceptable as well.

The determination of the prediction horizon and the weightings for the three cost functions is partially based on simulations and thus described in the simulations section (Chapter 5).

In the sequel of the report, it is evaluated in simulations which of the three cost function types given above is the most suitable for the intended control approach.

4.4 The Complete Optimal Control Problem

Based on the prediction model, the inequality constraints and the cost function candidates described in the previous sections, the related complete discrete optimisation

problems that define the MPC control input can be composed. Depending on the cost function type that is implemented, the optimisation problem takes a different form. For the three cost function types as described above, the resulting optimisation problems are given in the following.

Using the basic cost function (4.19) including kinetic energy reference tracking, the problem takes the following form:

$$\begin{aligned}\min_{\hat{F}_{trac}(k+i|k),\hat{u}_{cons}(k+i|k)} \quad & l_{e,b}\cdot\hat{E}_{batt}(k+N_p|k)+q_{e,b}\cdot\left(\hat{e}_{kin}(k+N_p|k)-\hat{e}_{kin,ref}(k+N_p|k)\right)^2 \\ & +\sum_{i=0}^{N_p-1} q_{s,b}\cdot\left(\hat{e}_{kin}(k+i|k)-\hat{e}_{kin,ref}(k+i|k)\right)^2\end{aligned} \tag{4.30a}$$

subject to the model of the system dynamics:

$$\begin{aligned}&\begin{bmatrix}\hat{e}_{kin}(k+i+1|k)\\ \hat{E}_{batt}(k+i+1|k)\end{bmatrix}=A_d(k+i|k)\cdot\begin{bmatrix}\hat{e}_{kin}(k+i|k)\\ \hat{E}_{batt}(k+i|k)\end{bmatrix}+\\ &B_d\cdot\begin{bmatrix}\hat{F}_{trac}(k+i|k)\\ \hat{u}_{cons}(k+i|k)\end{bmatrix}+E_d\cdot\begin{bmatrix}\sin(\hat{\alpha}_{sl}(k+i|k))\\ \cos(\hat{\alpha}_{sl}(k+i|k))\end{bmatrix}\quad \forall i\in\{0,1,...,N_p-1\}\end{aligned} \tag{4.30b}$$

subject to the initial conditions:

$$\hat{E}_{batt}(k|k)=E_{batt}(0);\quad \hat{e}_{kin}(k|k)=e_{kin}(0) \tag{4.30c}$$

subject to the limitations of the traction force:

$$\begin{aligned}&0\le\hat{e}_{kin}(k+i|k)\quad\forall i\in\{0,1,...,N_p-1\}\\ &g_1\cdot\hat{e}_{kin}(k+i|k)+g_2\le\hat{F}_{trac}\le g_3\cdot\hat{e}_{kin}(k+i|k)+g_4\quad\forall i\in\{0,1,...,N_p-1\}\end{aligned} \tag{4.30d}$$

subject to the approximations of the energy consumption map:

$$\begin{aligned}&\hat{u}_{cons}(k+i|k)\ge a_j\cdot\hat{e}_{kin}(k+i|k)+b_j\cdot\hat{F}_{trac}(k+i|k)+c_j\\ &\forall j\in\{1,2,...,6\}\quad\forall i\in\{0,1,...,N_p-1\}\end{aligned} \tag{4.30e}$$

When applying the zone objective (4.21), the soft constraints on the kinetic energy (4.20) are added to the optimisation problem as given below.

$$\min_{\hat{F}_{trac}(k+i|k),\hat{u}_{cons}(k+i|k),\hat{s}_l(k+i|k)} \quad l_{e,z}\cdot\hat{E}_{batt}(k+N_p|k)|| + \sum_{i=0}^{N_p} q_{s,z}\big(\hat{s}_l(k+i|k)\big)^2 \tag{4.31a}$$

subject to the model of the system dynamics:

$$\begin{bmatrix}\hat{e}_{kin}(k+i+1|k)\\ \hat{E}_{batt}(k+i+1|k)\end{bmatrix} = A_d(k+i|k)\cdot\begin{bmatrix}\hat{e}_{kin}(k+i|k)\\ \hat{E}_{batt}(k+i|k)\end{bmatrix} + B_d\cdot\begin{bmatrix}\hat{F}_{trac}(k+i|k)\\ \hat{u}_{cons}(k+i|k)\end{bmatrix} + E_d\cdot\begin{bmatrix}\sin(\hat{\alpha}_{sl}(k+i|k))\\ \cos(\hat{\alpha}_{sl}(k+i|k))\end{bmatrix} \quad \forall i\in\{0,1,...,N_p-1\} \tag{4.31b}$$

subject to the initial conditions:

$$\hat{E}_{batt}(k|k) = E_{batt}(0); \quad \hat{e}_{kin}(k|k) = e_{kin}(0) \tag{4.31c}$$

subject to the limitations of the traction force:

$$\begin{gathered}0\le\hat{e}_{kin}(k+i|k) \quad \forall i\in\{0,1,...,N_p-1\}\\ g_1\cdot\hat{e}_{kin}(k+i|k)+g_2\le\hat{F}_{trac}\le g_3\cdot\hat{e}_{kin}(k+i|k)+g_4 \quad \forall i\in\{0,1,...,N_p-1\}\end{gathered} \tag{4.31d}$$

subject to the approximations of the energy consumption map:

$$\begin{gathered}\hat{u}_{cons}(k+i|k)\ge a_j\cdot\hat{e}_{kin}(k+i|k)+b_j\cdot\hat{F}_{trac}(k+i|k)+c_j\\ \forall j\in\{1,2,...,6\} \quad \forall i\in\{0,1,...,N_p-1\}\end{gathered} \tag{4.31e}$$

subject to the soft constraints defining a zone for the kinetic energy:

$$\begin{gathered}\hat{e}_{kin,min}(k+i|k)+\hat{l}_{s,1}(k+i|k)\le\hat{e}_{kin}(k+i|k)\le\hat{e}_{kin,max}+\hat{l}_{s,2}(k+i|k)\\ \forall i\in\{0,1,...,N_p-1\}\end{gathered} \tag{4.31f}$$

For the controller formulation with guaranteed stability including cost function (4.28), the optimisation problem to determine the eco-cruise control action is:

$$\min_{\hat{F}_{trac}(k+i|k),\hat{u}_{cons}(k+i|k)} \quad l_{e,s}\cdot\hat{E}_{batt}(k+N_u)+\bar{q}_{e,s}\cdot\left(\hat{e}_{kin}(k+N_u|k)-\hat{e}_{kin,ref}(k+N_u|k)\right)^2 + \sum_{i=0}^{N_u-1}\Big(q_{s,s}\cdot\left(\hat{e}_{kin}(k+i|k)-\hat{e}_{kin,ref}(k+i|k)\right)^2 + r_{s,s}\cdot\left(\hat{F}_{trac}(k+i|k)-\hat{F}_{trac,ref}(k+i|k)\right)^2\Big) \tag{4.32a}$$

subject to the model of the system dynamics:

$$\begin{bmatrix}\hat{e}_{kin}(k+i+1|k)\\ \hat{E}_{batt}(k+i+1|k)\end{bmatrix} = A_d(k+i|k)\cdot\begin{bmatrix}\hat{e}_{kin}(k+i|k)\\ \hat{E}_{batt}(k+i|k)\end{bmatrix} + \tag{4.32b}$$

$$B_d\cdot\begin{bmatrix}\hat{F}_{trac}(k+i|k)\\ \hat{u}_{cons}(k+i|k)\end{bmatrix} + E_d\cdot\begin{bmatrix}\sin(\hat{\alpha}_{sl}(k+i|k))\\ \cos(\hat{\alpha}_{sl}(k+i|k))\end{bmatrix} \quad \forall i\in\{0,1,...,N_p-1\} \tag{4.32c}$$

subject to the initial conditions:

$$\hat{E}_{batt}(k|k)=E_{batt}(0);\quad \hat{e}_{kin}(k|k)=e_{kin}(0) \tag{4.32d}$$

subject to the limitations of the traction force:

$$\begin{gathered}0\le\hat{e}_{kin}(k+i|k)\quad\forall i\in\{0,1,...,N_p-1\}\\ g_1\cdot\hat{e}_{kin}(k+i|k)+g_2\le\hat{F}_{trac}\le g_3\cdot\hat{e}_{kin}(k+i|k)+g_4\quad\forall i\in\{0,1,...,N_p-1\}\end{gathered} \tag{4.32e}$$

subject to the approximations of the energy consumption map:

$$\begin{gathered}\hat{u}_{cons}(k+i|k)\ge a_j\cdot\hat{e}_{kin}(k+i|k)+b_j\cdot\hat{F}_{trac}(k+i|k)+c_j\\ \forall j\in\{1,2,...,6\}\quad\forall i\in\{0,1,...,N_p-1\}\end{gathered} \tag{4.32f}$$

The three above controller formulations are tested and compared in simulations in the sequel of this report to determine the most suitable among the three for the given use case.

4.5 Controller Formulation as Quadratic Program

The three controller formulations proposed in Section 4.4 are all based on a discrete / finite dimensional quadratic cost function and linear constraints. Consequently, all three optimisation problems can be reformulated as a *quadratic program* according to (2.9).

4.5.1 Setup of the Matrices of the Quadratic Program

To achieve this, the first step is to expand the state propagation for the discrete state space prediction model (4.14). It must be noted that this model is position variant and thus the system matrix A_d changes with the prediction step i at time k.

$$\underbrace{\begin{bmatrix} \hat{e}_{kin}(k+1|k) \\ \hat{E}_{batt}(k+1|k) \\ \hat{e}_{kin}(k+2|k) \\ \hat{E}_{batt}(k+2|k) \\ \vdots \\ \hat{e}_{kin}(k+N|k) \\ \hat{E}_{batt}(k+N|k) \end{bmatrix}}_{\hat{X}} =$$

$$\underbrace{\begin{bmatrix} B_d & 0 & \dots & 0 \\ A_d(k+1|k)B_d & B_d & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots \\ \prod_{i=1}^{N}\big(A_d(k+i|k)\big)B_d & \prod_{i=1}^{N-1}\big(A_d\big)B_d & \dots & B_d \end{bmatrix}}_{G} \cdot \underbrace{\begin{bmatrix} \hat{F}_{trac}(k|k) \\ \hat{u}_{cons}(k|k) \\ \hat{F}_{trac,1}(k+1|k) \\ \hat{u}_{cons,1}(k+1|k) \\ \vdots \\ \hat{F}_{trac}(k+N-1|k) \\ \hat{u}_{cons}(k+N-1|k) \end{bmatrix}}_{\hat{U}}$$

$$+ \underbrace{\begin{bmatrix} E_d & 0 & \dots & 0 \\ A_d(k+1|k)E_d & E_d & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots \\ \prod_{i=1}^{N}\big(A_d(k+i|k)\big)E_d & \prod_{i=1}^{N-1}\big(A_d(k+i|k)\big)E_d & \dots & E_d \end{bmatrix}}_{W} \cdot \underbrace{\begin{bmatrix} \sin(\hat{\alpha}_{sl}(k|k)) \\ \cos(\hat{\alpha}_{sl}(k|k)) \\ \sin(\hat{\alpha}_{sl}(k+1|k)) \\ \cos(\hat{\alpha}_{sl}(k+1|k)) \\ \vdots \\ \sin(\hat{\alpha}_{sl}(k+N-1|k)) \\ \cos(\hat{\alpha}_{sl}(k+N-1|k)) \end{bmatrix}}_{\hat{D}}$$

$$+ \underbrace{\begin{bmatrix} A_d(k|k) \\ \prod_{i=0}^{2}\big(A_d(k+i|k)\big) \\ \vdots \\ \prod_{i=0}^{N}\big(A_d(k+i|k)\big) \end{bmatrix}}_{P_0} \cdot \underbrace{\begin{bmatrix} \hat{e}_{kin}(k|k) \\ \hat{E}_{batt}(k|k) \end{bmatrix}}_{\hat{X}_0} \tag{4.33}$$

Accordingly, the states $\hat{X} = [\hat{x}(k+1|k), \dots + \hat{x}(k+N_u|k)]$ in the cost function can

be substituted by the following relation where the only unknown parameters are the discrete control inputs $\hat{U} = [\hat{u}(k+0|k), \ldots + \hat{u}(k+N_u|k)]$ which are the optimisation variables / decision variables of the quadratic program.

$$\hat{X} = G \cdot \hat{U} + W \cdot \hat{D} + P_0 \cdot \hat{X}_0 \tag{4.34}$$

All three cost function candidates introduced in Section 4.3 can be generalised and expressed as penalties on linear and quadratic terms of the state tracking error and the inputs:

$$\begin{aligned} \min_{\hat{U}} \quad & L_{QP} \cdot \left(\hat{X} - \hat{X}_{ref}\right) \\ & + \left(\hat{X} - \hat{X}_{ref}\right)^T \cdot Q_{QP} \cdot \left(\hat{X} - \hat{X}_{ref}\right) \\ & + \left(\hat{U} - \hat{U}_{ref}\right)^T \cdot R_{QP} \cdot \left(\hat{U} - \hat{U}_{ref}\right) \end{aligned} \tag{4.35}$$

Substituting 4.34 into 4.35 finally delivers:

$$\begin{aligned} & \hat{U}^T \cdot \underbrace{\left(G^T Q_{QP} G + R_{QP}\right)}_{\frac{1}{2} H_{QP}} \cdot \hat{U} \\ & + \underbrace{\left(L_{QP} G + 2 \cdot \left(\hat{D}^T W^T Q_{QP} G + \hat{X}_0^T P_0^T Q_{QP} G - \hat{X}_{ref}^T Q_{QP} G - \hat{U}_{ref}^T R_{QP}\right)\right)}_{w_{QP}^T} \cdot \hat{U} \\ & + L_{QP} W \hat{D} + L_{QP} P_0 \hat{X}_0 - L_{QP} \hat{X}_{ref} \\ & + \hat{X}_0^T P_0^T Q_{QP} P_0 \hat{X}_0 + \hat{D}^T W T Q_{QP} W \hat{D} + \hat{X}_{ref}^T Q_{QP} \hat{X}_{ref} + \hat{U}_{ref}^T R_{QP} \hat{U}_{ref} \\ & + 2 \cdot \left(\hat{D}^T W^T Q_{QP} P_0 \hat{X}_0 - \hat{D}^T W^T Q_{QP} \hat{X}_{ref} - \hat{X}_0^T P_0^T Q_{QP} \hat{X}_{ref}\right) \end{aligned} \tag{4.36}$$

For the formation of the quadratic program, only the terms that depend on the optimisation variables $\hat{U}$ are of interest. All others are constants and do not influence the optimum input sequence $\hat{U}^*$. Consequently, the weighting matrices H_{QP} und w_{QP} can be derived according to the definitions in (4.36).

The matrices L_{QP}, Q_{QP} and R_{QP} contain the weightings on the states and inputs for each step of the control horizon:

$$
\begin{aligned}
L_{QP} &= \begin{bmatrix} L & 0 & \dots & 0 \\ 0 & L & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & \dots & L_t \end{bmatrix} \\
Q_{QP} &= \begin{bmatrix} Q & 0 & \dots & 0 \\ 0 & Q & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & \dots & \bar{Q} \end{bmatrix} \\
R_{QP} &= \begin{bmatrix} R & 0 & \dots & 0 \\ 0 & R & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & \dots & R \end{bmatrix}
\end{aligned}
\tag{4.37}
$$

The inequality constraints of the quadratic program are derived accordingly from the constraints on the inputs and states. In a generalised form, all constraints on the inputs and states used in the OCP formulation in Section 4.4 can be formulated as follows:

$$
F_{ineq,QP} \cdot \hat{X} + E_{ineq,QP} \cdot \hat{U} \leq g_{ineq,QP} \tag{4.38}
$$

By substituting $\hat{X}$ according to (4.34), the constraints can be reparameterised to constraints only on the decision variables $\hat{U}$ of the quadratic program.

$$
F_{ineq,QP} \cdot G \cdot \hat{U} + F_{ineq,QP} \cdot W \cdot \hat{D} + F_{ineq,QP} \cdot P_0 \cdot \hat{X}_0 + E_{ineq,QP} \cdot \hat{U} \leq g_{ineq,QP} \tag{4.39}
$$

It follows:

$$\left(E_{ineq,QP} + F_{ineq,QP} \cdot G\right) \cdot \hat{U} \leq g_{ineq,QP} - F_{ineq,QP} \cdot W \cdot \hat{D} - F_{ineq,QP} \cdot P_0 \cdot \hat{X}_0 \quad (4.40)$$

Herein, $E_{ineq,QP}$, $F_{ineq,QP}$ and $g_{ineq,QP}$ are defined as:

$$\begin{aligned} E_{ineq,QP} &= \begin{bmatrix} E_{ineq} & 0 & \dots & 0 \\ 0 & E_{ineq} & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & \dots & E_{ineq} \end{bmatrix} \\ F_{ineq,QP} &= \begin{bmatrix} F_{ineq} & 0 & \dots & 0 \\ 0 & F_{ineq} & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & \dots & F_{ineq} \end{bmatrix} \\ g_{ineq,QP} &= \begin{bmatrix} g_{ineq} \\ g_{ineq} \\ \vdots \\ g_{ineq} \end{bmatrix} \end{aligned} \quad (4.41)$$

With the above described methodology, all three OCP formulations given in Section 4.4 can be converted into a quadratic program according to (2.9) and then solved with available quadratic programming solvers.

The given way of formulation is called *condensed* formulation since the vector of optimisation variables only consists of the control input variables. The system state propagation is directly considered within the matrices H_{QP} and w_{QP}. An alternative way to formulate the optimisation problem is to consider the discrete control inputs as well as the system states as optimisation variables which are then linked by additional equality constraints. This leads to a much larger, but *sparse* and band structured Hessian matrix H_{QP} of the resulting quadratic program. By using special algorithms for sparse computation, this structure can be beneficial even though the dimensions are larger [137, 138]. However, a special solver is needed to make use of these advantages.

4.5.2 Solution of the Quadratic Program

For convex quadratic programs, it can be guaranteed that the solver converges to the global optimum and that this solution is found in polynomial time [108].

This is an extremely desirable property for a real-time application since the computation of the control signal can then be guaranteed to be successful and the required sample time can reliably be estimated.

Whether a quadratic program is convex only depends on the weighting matrix on the squared decision variables H, its *Hessian matrix*. If H is positive semidefinite, the quadratic program is convex. [13]

$$\begin{aligned} \min_{y} \quad & \frac{1}{2} y^T H y + w^T y \\ & \text{subject to:} \\ & A_{ineq} \cdot y \leq b_{ineq} \end{aligned} \tag{4.42}$$

A comparison to (4.36) shows that the Hessian H_{QP} of the regarded controller formulation only depends on the matrices A_d and B_d of the prediction model (4.14) (that have an influence on G), the weightings on the squared state tracking error Q and Q_t as well as the weightings on the inputs R.

Consequently, if the problem is convex, it will remain convex for any reference and disturbance trajectory $\hat{X}_{ref}$, $\hat{U}_{ref}$ and $\hat{D}_{ref}$. In other words, if the control formulation is proven to be convex, it will also be convex under any operating conditions.

A more detailed look on the definition

$$H_{QP} = 2 \cdot (G^T Q_{QP} G + R_{QP}) \tag{4.43}$$

reveals that defining $Q \geq 0$ and $\bar{Q} \geq 0$ is enough to ensure that $G^T Q_{QP} G \geq 0$. If the weightings $R(i)$ are also positive for each i, the matrix H will be positive (semi)-definite and the optimisation problem will be convex. [13]

This is given for the above specified cost function definitions and hence, a favourable formulation for real-time capability is achieved.

Since there are standard algorithms for solving feasible convex QPs in real-time, the solution strategy for the quadratic programs is not further discussed in this and it is relied on available solvers like the function *quadprog()* within the software *MATLAB* and the solver package *QPOASES* [139] in the C-code based experimental implementation.

It shall only be mentioned that two resolution strategies have shown to be the most powerful for the solution of (constrained) quadratic programs, namely *active-set* [140] and *interior-point* [141, 142] methods [13].

Active-set strategies solve a purely equality constrained problem at each iteration by making a guess which of the inequality constraints are equally fulfilled (i. e. active) (those are treated like equality constraints) and which of them are inactive (those are ignored). This way, each subproblem is easier to solve. Consequently, the working points at each iteration lie at the border of the feasible region if there are active constraints. The guess of the active set of inequality constraints is updated at each iteration until the optimum solution is found. On the other hand, interior-point methods generate iterates that lie inside the feasible region by using so-called *barrier functions*. The barrier functions are added to the cost function and increase its value drastically if constraints are approached. [13]

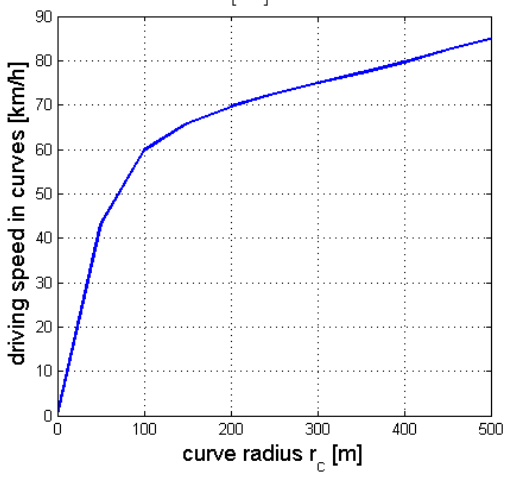
Both resolution strategies have in common that inequalities are treated by reformulating the problem to another one without inequality constraints.

For the details on the algorithms, it is referred to the literature [140, 141, 142].

4.6 Speed Reference Generation

In the preceding chapters it has been assumed that a speed (or kinetic energy) reference trajectory is given throughout the prediction horizon. This section explains how to generate the reference trajectory from the information of the digital map about the road in front of the host vehicle and the headway distance measurements of the radar sensor.

FIGURE 4.12: Relation between the average driving speed in curves and the curve radius r_c based on field experiments with drivers of passenger cars. Figure edited from [14].



4.6.1 Extraction of the Reference Speed from the Available Data

The driving speed reference for the cruise controller can be considered as the desired speed and depends on several influence factors. First of all, the legal speed limits $v_{lim}(s)$ should be followed and can be taken as a first guess for targeting the driving speed.

Furthermore, the curve radius of the road limits the drivable speeds for safety and comfort reasons. One possible way to consider this is to model the lateral acceleration in the prediction model of the controller and to impose constraints on it. However, in order to keep the optimal control problem as simple as possible, maintain the real-time capability and feasibility, curves are considered here by defining the speed reference accordingly without imposing additional constraints.

Hence, the maximum curve speed is considered within the speed reference in the at-hand approach. To associate the curve radius $r_c(s)$ with a safe and comfortable driving speed, the results of a study investigating the relation $v(r_c)$ in field experiments have been applied [14]. In this study, the drivers of passenger cars have run their cars at the average curve speed given in Fig. 4.12.

The illustrated relation between the driving speed in curves $v_{cur}(s)$ and the curve radius $r_c(s)$ has been assumed as a speed reference in curves:

$$v_{cur} = f\big(r_c(s)\big) \tag{4.44}$$

A second task of an adaptive cruise controller is to maintain a certain distance to preceding cars. A prerequisite for this is the measurement of the distance to the car ahead and the relative speed. This is achieved by using an automotive radar sensor like in most adaptive cruise control realisations providing both distance and relative speed of the preceding car. While the actual distance can be measured, the future evolution over the prediction horizon is unknown and depends on the speed of both cars. Since no information about the preceding car is available, its driving speed is assumed to remain constant at the current value throughout the prediction horizon.

A straightforward approach to incorporate a headway control of the distance to the preceding car with a predictive cruise controller is to model the distance in the prediction model such that cost function penalties or constraints can be imposed on the predicted distance. Using the time as independent variable, the change rate of the distance to the preceding car d_h is simply described by the difference between the speed of the preceding vehicle v_p and the speed of the host vehicle v_h.

$$\frac{dd_h}{dt} = v_p - v_h(t) \tag{4.45}$$

By converting the model to the position domain according to the prediction model (4.14), equation (4.45) has to be divided by the host vehicle speed which would lead to a non-linear prediction model and make the optimisation much more complicated:

$$\frac{dd_h}{ds} = \frac{v_p - v_h(s)}{v_h(s)} \tag{4.46}$$

To avoid this, the functionality of headway control is included in the speed reference instead by pre-computing the speed-trajectory that leads to the desired headway distance trajectory. Here, several cases must be distinguished according to the relation of the

current speed and distance measurements. To assess the given case, a safety distance d_{safe} is defined here as a function of the speed of the preceding vehicle v_p according to the heuristic rule of thumb:

$$d_{h,safe}[m] = \frac{1}{2} \cdot v_p[km/h] \tag{4.47}$$

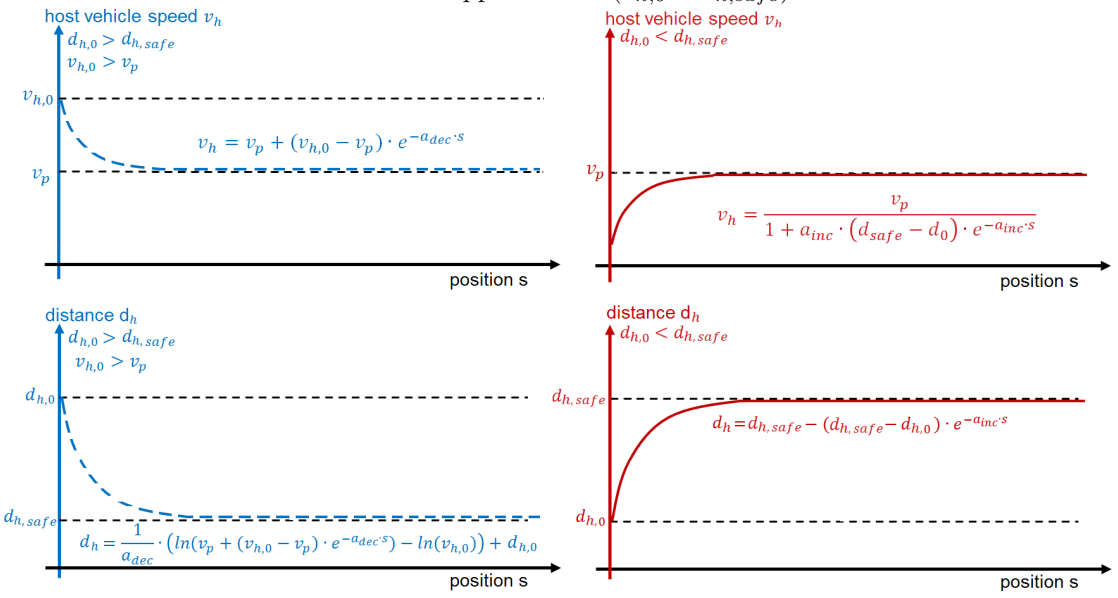
Case 1: $v_h > v_p$ **and** $d_{h,0} > d_{h,safe}$: The host vehicle is approaching a preceding car that is running at a lower speed while the current distance $d_{h,0}$ is above the safety distance $d_{h,safe}$. In this case, a safe and smooth approach of the speed of the preceding car ($v_h \rightarrow v_p$) and the safety distance ($d_h \rightarrow d_{h,safe}$) is demanded. In order to obtain a strong decrease of the speed at the beginning of the prediction horizon (to be safe) and then subsequently a smooth approach of the speed of the preceding vehicle, an exponential decay function beginning at $v_h = v_h(s = 0)$ and approaching $v_h = v_p$ for $s \rightarrow \infty$ is defined, see Fig. 4.13 (dash-dotted blue curves). In all cases, the speed of the preceding vehicle v_p is supposed to remain constant at the current measurement throughout the whole prediction horizon.

$$v_h(s) = v_p + (v_h(s = 0) - v_p) \cdot e^{-a_{dec} \cdot s} \tag{4.48}$$

The parameter a_{dec} determines the rate of the exponential decay and is determined in such a way that the safety distance $d_{h,safe}$ is reached just when the speed of the host vehicle v_h almost reaches the speed of the preceding car ($v_h = 1.01 \cdot v_p$) which results in following the preceding car with the same speed at the safety distance. To derive an equation for a_{dec} accordingly, (4.48) is inserted into the formula of the derivative of the distance between the two cars with respect to the host vehicle position (4.46).

Inserting (4.48) into (4.46) and integration with respect to s leads to the relation for the distance between two cars throughout the prediction horizon:

FIGURE 4.13: Speed reference and resulting distance to the preceding car in the case of car-following. The left half of the figure (blue curves) shows the case that the current distance $d_{h,0}$ is greater than the safety distance $d_{h,safe}$. The right half (red curves) shows the opposite case ($d_{h,0} < d_{h,safe}$)



$$d_h(s) = \frac{1}{a_{dec}} \cdot ln(\underbrace{v_p + (v_h(s=0) - v_p) \cdot e^{-a_{dec} \cdot s}}_{v_h}) \underbrace{- \frac{1}{a_{dec}} ln\big(v_h(s=0)\big) + d_{h,0}}_{\text{integration const.}} \tag{4.49}$$

Demanding and setting $d_h := d_{h,safe}$ and $v_h := 1.01 \cdot v_p$, gives the required relation to determine a_{dec} such that the safety distance is reached just when the cars run with almost the same speed:

$$a_{dec} = \frac{ln(1.01 \cdot v_p) - ln\big(v_h(s=0)\big)}{d_{h,safe} - d_{h,0}} \tag{4.50}$$

Case 2: $d_{h,0} < d_{h,safe}$: In the case that the actual distance is below the safety distance ($d_{h,0} < d_{h,safe}$), an exponential increase of the distance $d_{h,0} \rightarrow d_{h,safe}$ is demanded according to (see Fig. 4.13, continuous red curves):

$$d_h(s) = d_{h,safe} - (d_{h,safe} - d_{h,0}) \cdot e^{-a_{inc} \cdot s} \tag{4.51}$$

With the help of (4.46), the resulting speed profile can be derived:

$$v_h(s) = \frac{v_p}{1 + a_{inc} \cdot (d_{h,safe} - d_{h,0}) \cdot e^{-a_{inc} \cdot s}} \tag{4.52}$$

a_{inc} is tuned with a simple rule of thumb according to $d_{h,safe}$ as follows: $a_{inc} = 1/d_{h,safe}$.

Case 3: $d_{h,0} \geq d_{h,safe}$ **and** $d_{h,0} < 200m$ **and** $v_{h,0} \leq v_p$: When the preceding car is faster than the host vehicle and the current distance is larger than the safety distance but below 200 m, the speed reference is constantly set to the speed of the preceding car $v_h(s) = v_p$.

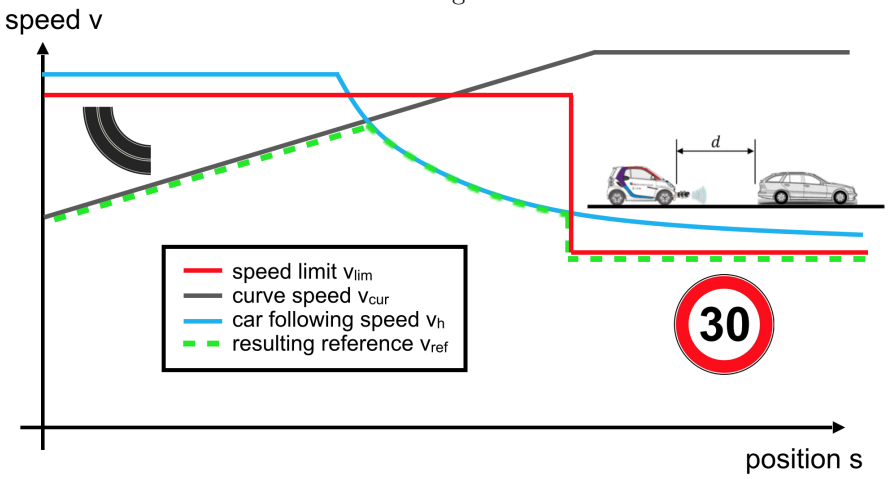
Case 4: $d_0 \geq 200\ m$: The detection range of the automotive radar sensor foreseen for the practical implementation has a maximum range of 200 m. Hence, the speed reference trajectory is not influenced by preceding cars in a distance above 200 m.

The equations and speed profiles related to the car-following behaviour are summarised in Fig. 4.13. In summary, there is a maximum speed at each step of the prediction horizon resulting from the legal speed limits ($v_{lim}(s)$), one resulting from the road curvature ($v_{cur}(s)$) and finally a speed trajectory resulting from the car-following ($v_h(s)$). From these three maximum speeds, none should be disobeyed and thus the lowest is decisive. Consequently, the minimum of these three maximum speeds is taken as the speed reference to be followed by the cruise controller. This procedure is illustrated exemplary in Fig. 4.14.

$$v_{ref}(s) = min\{v_{lim}(s), v_{cur}(s), v_h(s)\} \tag{4.53}$$

Finally, a kinetic energy reference is derived from the speed reference according to (4.5) to comply with the prediction model of the controller.

FIGURE 4.14: Generation of the reference speed as the minimum of the legal speed limit, the admissible speed in curves and the speed trajectory resulting from the car-following task.



In the case of a reference tracking controller, the use of the reference is straightforward. In the case of a zone objective, the generated kinetic energy trajectory is the upper limit of the admissible zone and a parallel shift of it is taken as the lower limit.

4.6.2 Reference Smoothing

The speed reference generation as described above produces reference trajectories including stepwise changes. This may be infavourable in terms of driving comfort and energy consumption since it necessitates strong reactions of the controller to bring the vehicle quickly to the new set-speed.

To generate a discrete reference trajectory with bounded changes from one sample step to the next that do not exceed the speed profile specified above, the following linear optimisation problem can be solved. The changes in the reference speed between two discretisation steps are bounded by the maximum value of r_{max}:

$$\min_{\hat{v}_{ref,smooth}(k+i|k)} \quad \sum_{i=1}^{N_p} (\hat{v}_{ref}(k+i|k) - \hat{v}_{ref,smooth}(k+i|k))$$

subject to: (4.54)

$$\hat{v}_{ref,smooth}(k+i+1|k) - \hat{v}_{ref,smooth}(k+i|k) \leq r_{max} \quad \forall i \in \{1,2,...,N_p\}$$

$$\hat{v}_{ref,smooth}(k+i|k) \leq \hat{v}_{ref}(k+i|k) \quad \forall i \in \{1,2,...,N_p\}$$

The smoothened reference speed values $v_{ref,smooth}(k+i|k)$ at the discretisation steps i are the optimisation variables in problem (4.54). By minimising (4.54), the smoothened reference $\hat{v}_{ref,smooth}(k+i|k)$ approximates the original one $\hat{v}_{ref}(k+i|k)$ (containing step changes). The inequality constraints ensure that the smoothened values never lie above the original ones (to maintain a safe driving) and that a maximum reference change of r_{max} is kept from one discretisation step to another.

The linear program (4.54) can be solved with low computational effort and ensures to maintain the requirements on the speed reference at best.

4.7 Conclusion of the Chapter

The chapter has presented how the dynamic motion and energy consumption model of the electric vehicle is formulated in a way that the optimal control problem can be stated as a quadratic optimisation problem, namely using a linear discrete state space model subject to linear inequality constraints and using a quadratic cost function.

This has been achieved by using a change of variables to predict the kinetic energy of the moving car as a measure of the driving speed, a position-variant model and the technique of separable programming.

The traffic situation is considered within the reference speed trajectory generation and thus without increasing the complexity of the optimisation problem. For this problem formulation it is possible to derive stability guarantees of the MPC controller system and to achieve real-time capability with guaranteed convergence.

Consequently, the presented formulation has many advantages compared to a lot of existing methods in the literature using dynamic programming which are not real-time capable. Furthermore, the presented reformulations are not based on an overall linearisation in one working point such that the non-linear nature of the problem is still maintained.

The main results of this chapter have been published in [143].

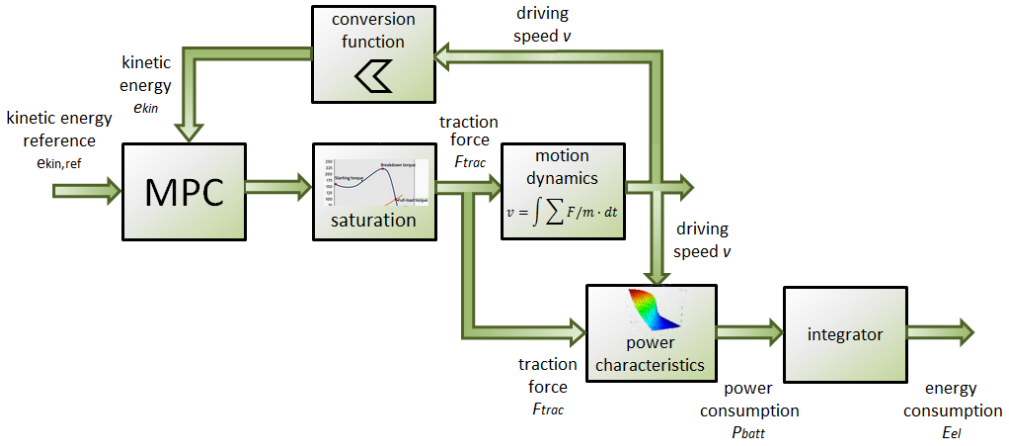
Chapter 5

Simulations of the Closed-Loop Energy-Efficient Control

This chapter presents simulations of the closed-loop control with the MPC controller described in Chapter 4 together with the simulation model derived in Chapter 3 according to Fig. 5.1. The current state of the system is taken from the non-linear simulation model and based on this, the MPC controller computes an optimised traction force input. This input is saturated according to the measured full-load curve (Fig. 4.7) and fed to the simulation.

The simulations serve several aims. First, the influence of different prediction / control horizon lengths and different weighting factors on the closed-loop performance is investigated in order to find a proper tuning of the control parameters. Secondly, the

FIGURE 5.1: Structure of the closed-loop simulation.



performance of the tuned controller is studied to evaluate the energy-saving potential and the speed reference tracking performance in standard situations and mixed driving scenarios. The investigated scenarios comprise up and down-slopes, reference changes and the car following behaviour. The simulation parameters are set according to Tabs. 4.1 and 3.2.

5.1 The Influence of the Prediction Horizon Length

This section investigates the reactions of the MPC controller with the basic cost function (i. e. optimisation problem (4.30)) to known disturbance and reference changes with different prediction horizon lengths. The results of this study are used to choose a reasonable horizon length as a trade-off between computational effort and control performance.

The simulation scenario is chosen to contain harsh changes in the driving speed reference and the road slope angle that necessitate early reactions of the controller and consequently put controllers with long prediction horizons in favour.

The decelerating force is limited to slight regenerative braking according to Fig. 4.7 such that changes in the driving speed reference requiring deceleration benefit from early reactions of the controller. Consequently, a harsh reference change from 120 km/h to 20 km/h is incorporated in the scenario at the position of 1000 m. Another 1000 m at this speed reference further, the reference changes to 70 km/h and remains at this value. At a position of 3000 m, a harsh change in the road slope from 0 to -5 ° is introduced and kept for 500 m. 500 m further on even road without slope, the road slope changes abruptly to an up-slope of +5 ° and is again kept constant for 500 m before it changes back to flat road. In summary, these abrupt changes in the reference and (known) disturbance are chosen to provoke early reactions of the controller that can show the benefit of a longer prediction horizon.

As an initial guess for the weightings in the basic cost function of (4.30)), the weightings on the accumulated as well as terminal tracking error $q_{e,b}$ and $q_{s,b}$ are set to the value 1. The weighting on the terminal energy consumption $l_{e,b}$ is adjusted differently for each horizon length such that

- energy-saving effects are clearly visible and
- the constant driving speed levels are equal despite for all of the prediction horizon length.

An adjustment for different horizon lengths is necessary for the following reason. When the prediction horizon length is increased, the (linear) terminal energy consumption grows linearly with the number of prediction steps while the accumulated tracking error on the other hand grows with the sum of squares. This necessitates an adjustment of the weightings to achieve a comparable control behaviour throughout the different simulation runs.

The overall view on the simulation results with the applied values of $l_{e,b}$ are given in Fig. 5.2 for MPC controllers with prediction horizon lengths between 50 and 800 m, discretised in 10 m steps, as well as the global optimum solution (one-step optimisation for a horizon of 5000 m). The control horizons are equal to the prediction horizons, i.e. $N_c = N_p$.

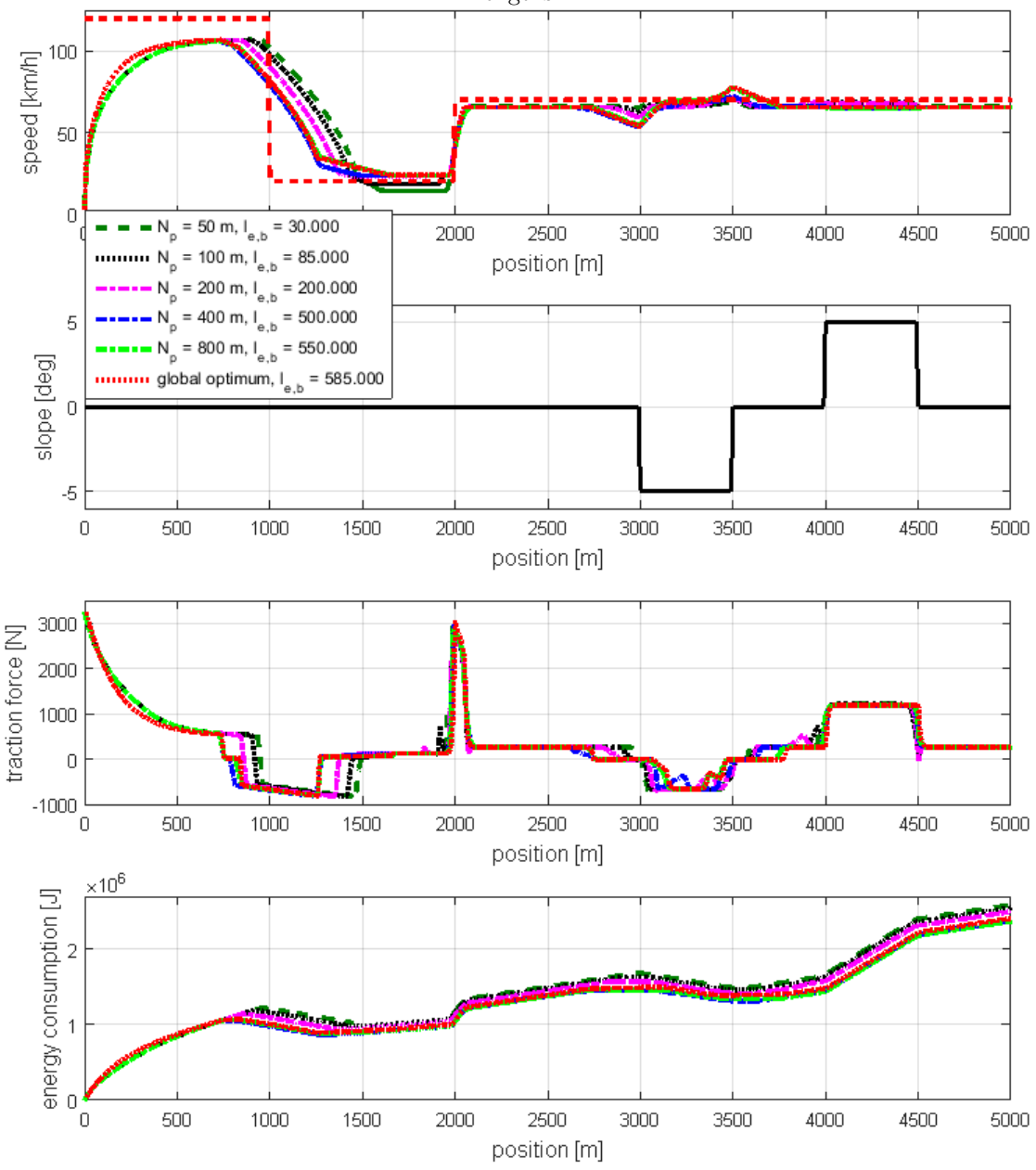
The differences between the controllers become mainly visible in the area of the harsh deceleration reference change at the position of 1000 m and during the down-slope driving between 3000 and 3500 m. For this reason, these two areas are zoomed and plotted in Fig. 5.3 and Fig. 5.4 for a more detailed investigation.

Generally, the MPC controllers with a longer prediction horizon can show earlier reactions to a reference change. This intuition is confirmed by the simulation results in Fig. 5.3 and can be seen from the control input trajectory (traction force) and the driving speed trajectory. The controller with a prediction horizon of 400 m (blue dash-dotted line) is an exception here with an earlier reaction than the other controllers.

Furthermore, the figure shows that the reactions of the controllers with larger prediction horizons converge to the global optimum one-step solution. This result shows that prediction horizons longer than 800 m do not yield benefits regarding the control performance.

Fig. 5.4 shows the reactions of the controllers to down-slopes. Here, it can again be seen that the controllers with longer prediction horizons tend to earlier control reactions to the slope. Further, it can be noticed that this also leads to a higher saving potential by

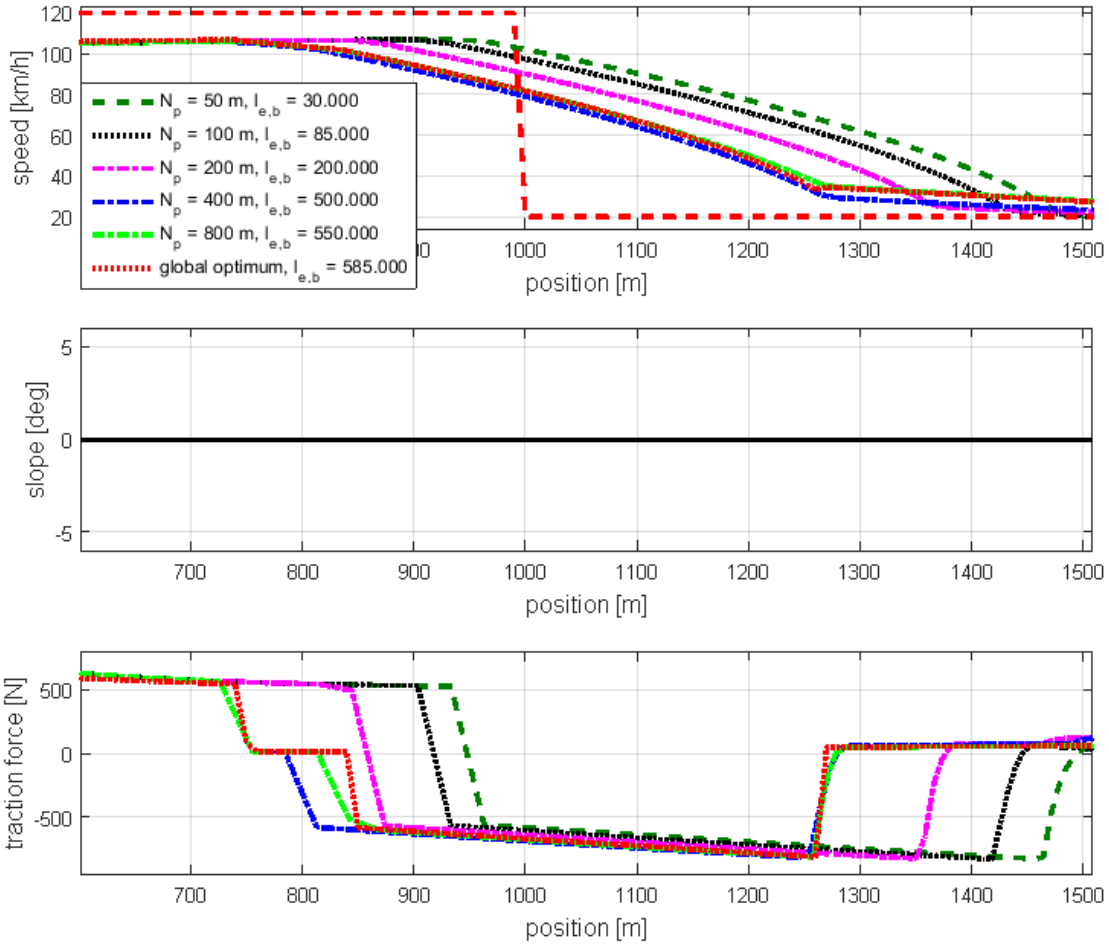
FIGURE 5.2: Results of the closed-loop simulation with different prediction horizon lengths.



- decelerating earlier and
- gaining a higher speed (and thus momentum and kinetic energy) throughout the down-slope.

Temporarily, this leads to a larger deviation from the speed reference trajectory but in an eco-cruise control formulation, this behaviour is intended to save energy as long as the average speed remains acceptable. The overall simulation results in Fig. 5.2 confirm that the longer the prediction horizon, the higher the energy savings.

FIGURE 5.3: Zoom of the simulation results with the deceleration manoeuvre with different prediction horizons.



So far, it can be said that longer prediction (and control) horizons lead to a more favourable control performance. However, this is linked with a higher computational effort. To finally pick a reasonable prediction horizon length, the average driving speed, the energy consumption and the computation time to solve the MPC problem with different horizons are compared to find a good compromise. Since it is not possible to compute the global optimum solution for an entire trip in reasonable time, an MPC with a prediction horizon of 1600 m is included in this part of the analysis, instead. The computation time is measured within *MATLAB* using the *quadprog* function as a solver on an *Intel i7 2.2 GHz quad-core* desktop PC and represents the time to solve one optimisation problem. The results are given in Fig. 5.5.

According to this comparison, the computation time increases drastically for prediction horizons of 800 m and more. On the other hand the energy consumption does not change significantly for prediction horizons of 400 m and longer in this scenario. Thus,

FIGURE 5.4: Zoom of the simulation results with the down-slope driving with different prediction horizons.

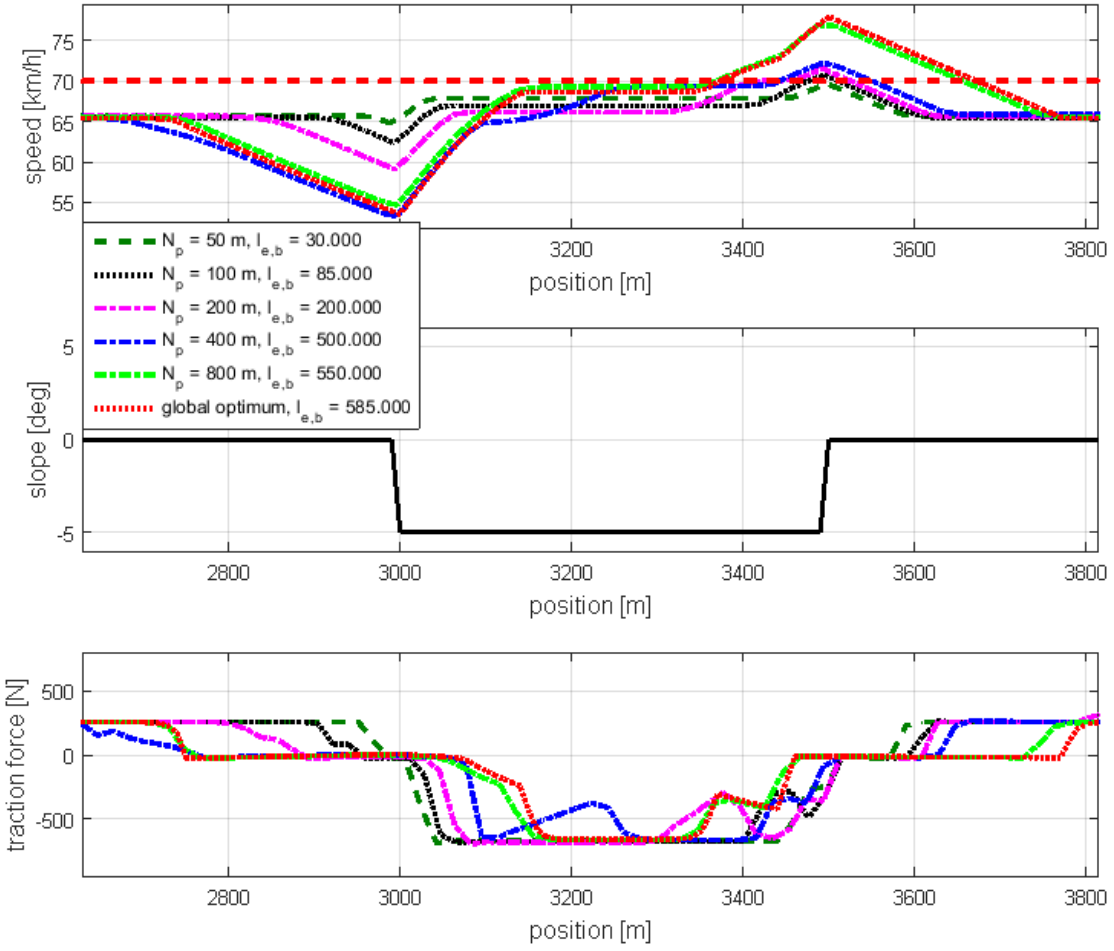
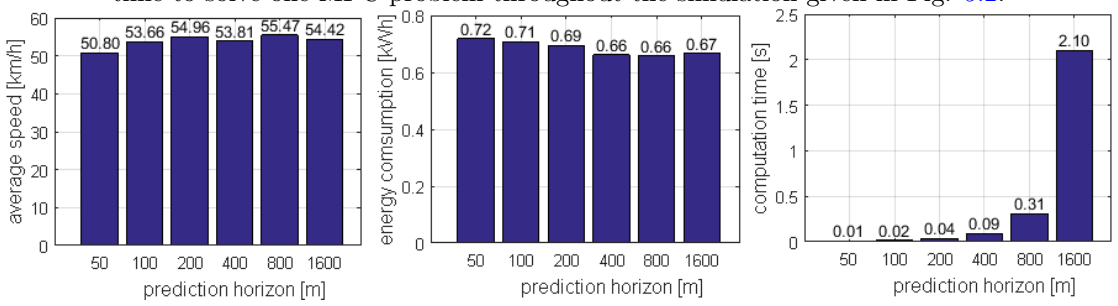


FIGURE 5.5: Comparison of the average speed, energy consumption and computation time to solve one MPC problem throughout the simulation given in Fig. 5.2.



a prediction horizon of 400 m is chosen for the application of the energy-saving cruise controller in the sequel of this thesis and shows to be suitable in the following simulation studies.

5.2 The Influence of the Weighting Factors

Besides the length of the prediction horizon, the weighting factors are tuning parameters for the performance and behaviour of an MPC controller. This section investigates the influence of different weightings on the energy consumption term $\hat{E}_{batt}(k+N_p|k)$ in (4.30). The prediction horizon is now constantly kept at a length of 40 steps of 10 m (400 m in total), based on the results of the previous section. Although there exist theoretical approaches for the tuning of linear MPC controllers, they are not applicable in the given application case. This is because the theoretical approaches consider the classical controller design targets such as stability and regulation to a constant set-point. These are not applicable here since the main aim of an energy-saving cruise controller is optimised high-level planning to find an acceptable compromise between proper speed reference tracking and energy saving. Consequently, a more appropriate way of finding a proper tuning are closed-loop simulations.

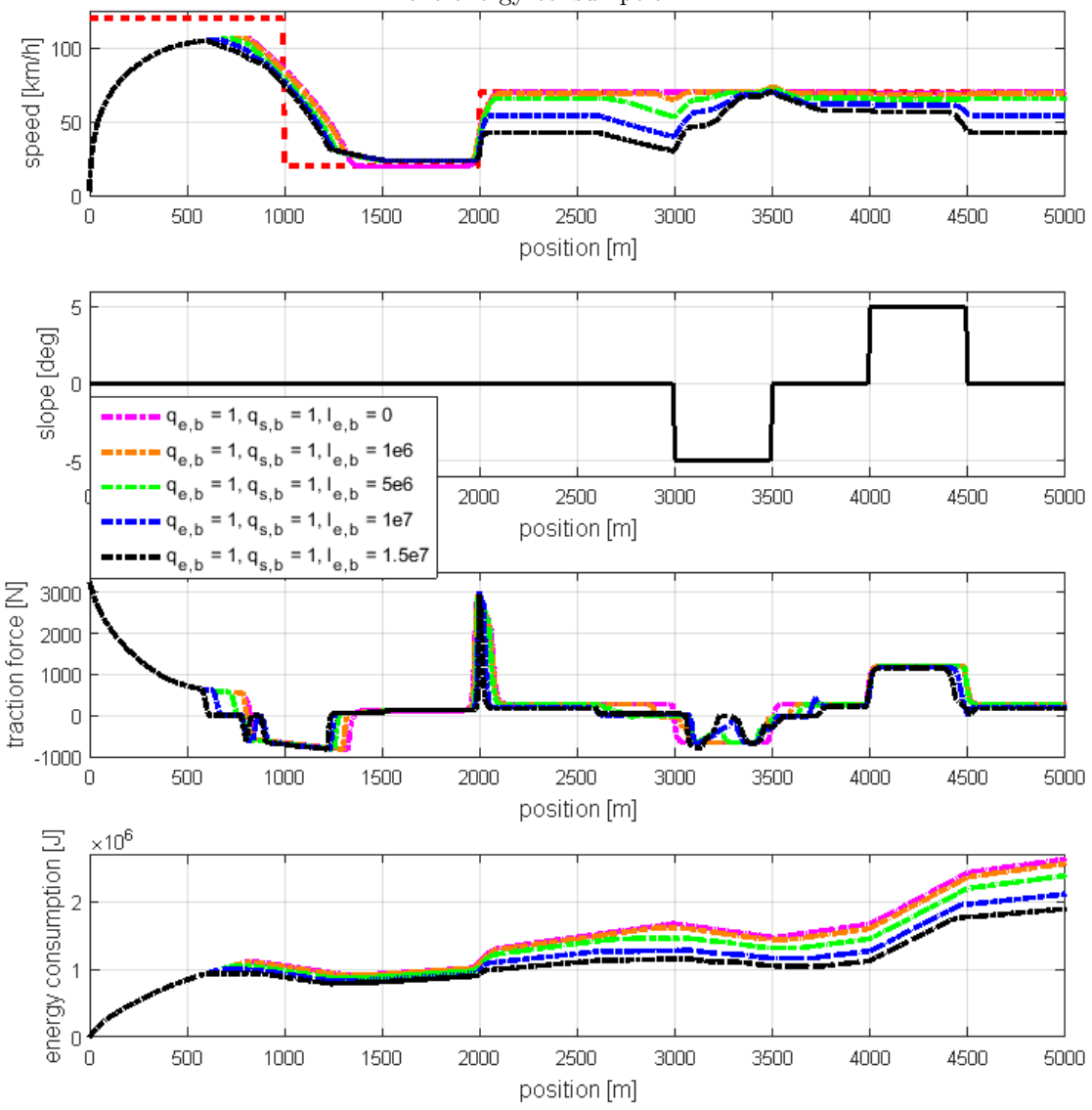
The performance of the controller is mainly driven by the ratio of the weighting on the terminal energy consumption $\hat{E}_{batt}(k+N_p|k)$ and the accumulated tracking error $\sum_{i=0}^{N_p-1}\left(\hat{e}_{kin}(k+i|k)-\hat{e}_{kin,ref}(k+i|k)\right)^2$.

The weightings on the terms related to the speed reference tracking $q_{e,b}$ and $q_{s,b}$ are kept at the value one, here, while the weighting $l_{e,b}$ on the terminal energy consumption is varied in the different simulation runs. The simulation scenario is the same as previously defined in Section 5.1. The overview on the results is given in Fig. 5.6.

Fig. 5.6 shows that a higher weighting $l_{e,b}$ on the terminal energy consumption indeed leads to higher energy savings. On the other hand, the driving speed is reduced as well. This already indicates that the choice of weightings leads to the question how much speed reduction is acceptable in favour of energy saving.

Fig. 5.7 provides a closer look on the reactions of the differently tuned controllers to the speed reference changes in the scenario. The detailed look on the deceleration between

FIGURE 5.6: Results of a closed-loop simulation results with different weightings on the energy consumption.

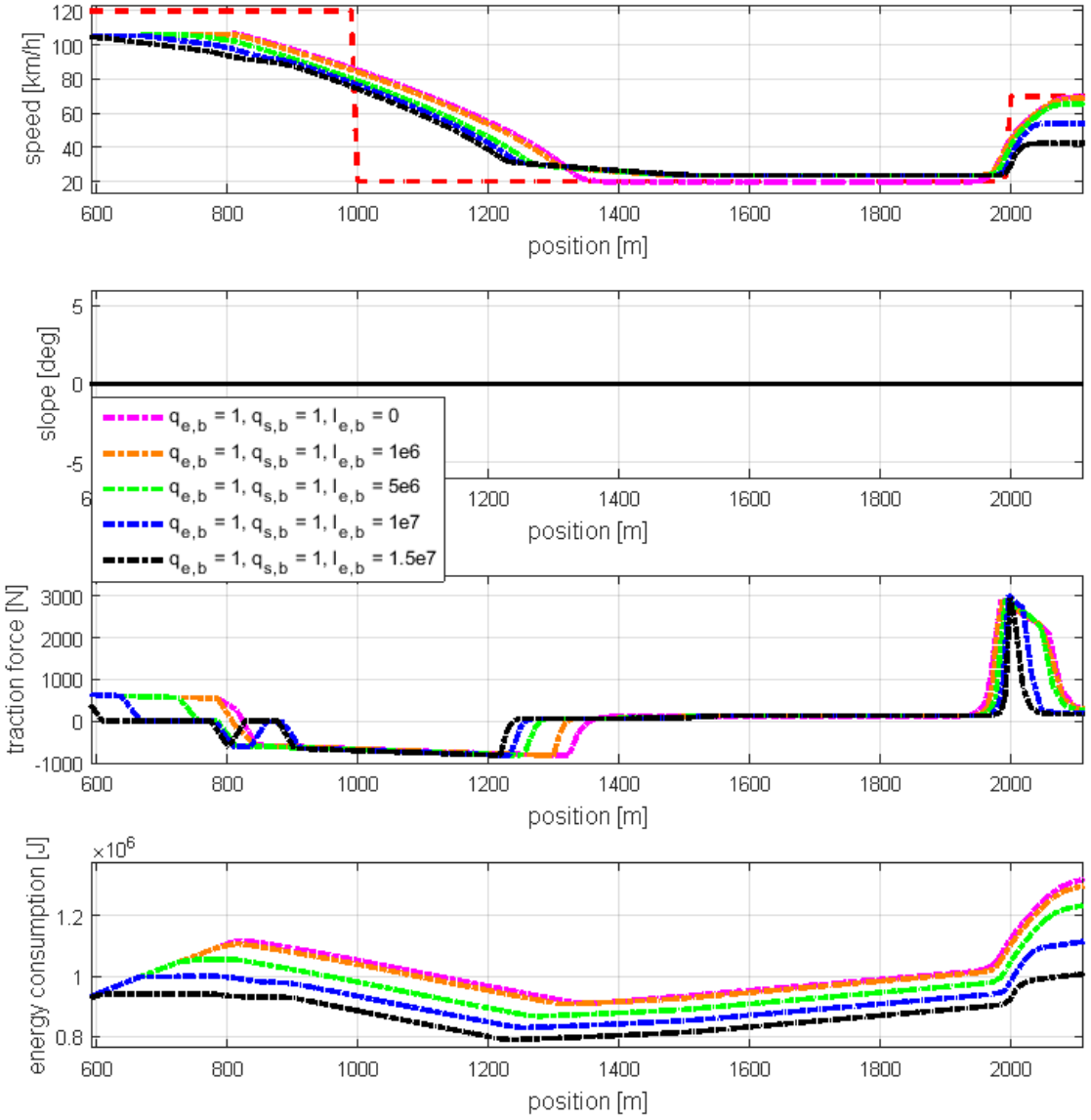


the positions 700 and 1000 m reveals that the controllers with higher weighting on the terminal energy consumption reduce the traction force earlier which leads to an earlier deceleration.

Furthermore, the speed reference of 20 km/h is exceeded by the energy-saving controllers and the subsequent acceleration with a reference of 70 km/h is performed with a shorter traction force peak by the controllers with higher weighting factors $l_{e,b}$. The higher this weighting, the lower is the constant driving speed after this final reference change at the position of 2000 m.

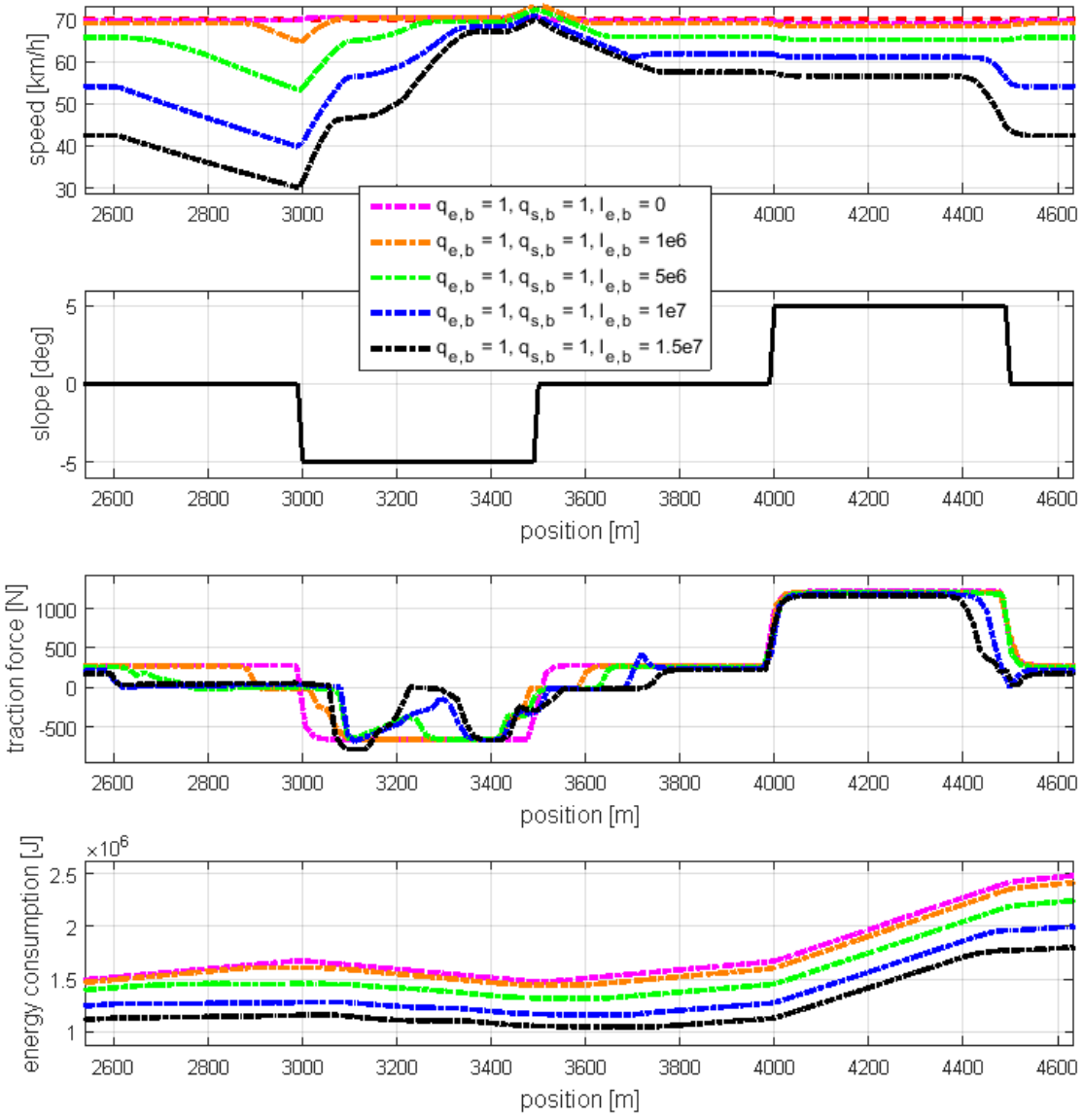
The reactions of the controllers to down-slopes and up-slopes is given in Fig. 5.8.

FIGURE 5.7: Zoom of the simulation results with reference changes and different weightings on the terminal energy consumption.



The controller with zero weighting on the energy consumption (magenta-coloured dash-dotted line) tracks the reference almost exactly despite the down and up-slope. All of the energy-saving controllers show the same saving mechanism: the speed is reduced before the down-slope is reached in order to save energy and then recover the speed again without traction force effort during the down grade. The driving speed increases throughout the down-slope until it finally exceeds the reference, before the simulated vehicle decelerates again after the end of the slope. The weighting on the energy consumption influences how early the controller decelerates in front of the down-slope and to which extent.

FIGURE 5.8: Zoom of the simulation results with up and down-slope driving and different weightings on the terminal energy consumption.

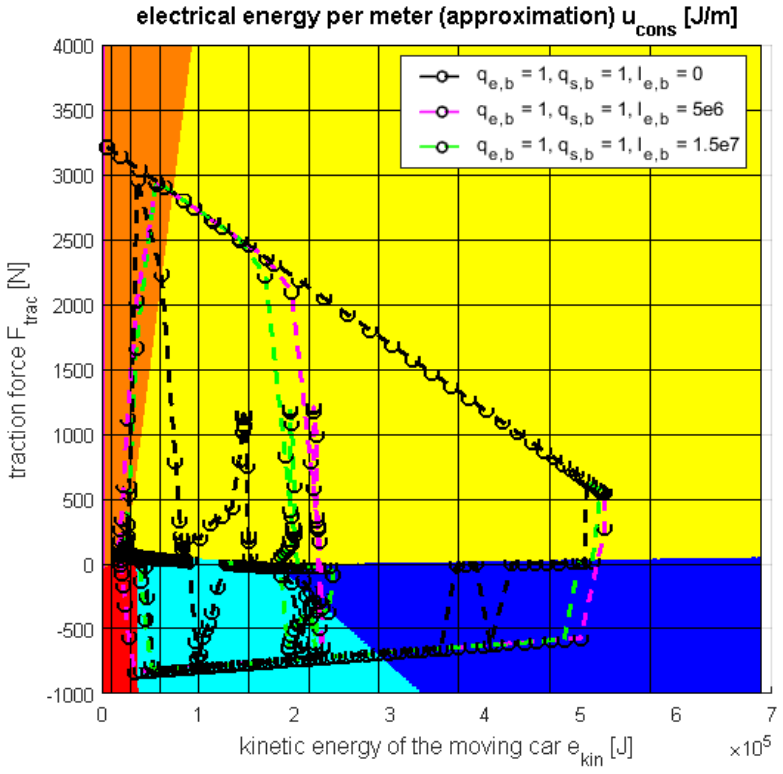


During the up-hill driving (between the positions 4000 and 4500 m), the energy-saving controllers slightly reduce the driving speed compared to the sections with even roads. This leads to a reduced traction force and thus a reduced energy consumption.

To understand the behaviour of the controllers in detail, the operating point trajectories (the pairs of kinetic energy and traction force values) of three of the above simulated controllers are projected on the piecewise linear approximation of the energy consumption map which is part of the controller prediction model and depicted in Fig. 5.9.

Obviously, the operating points are more and more shifted to the intersection lines of the approximating linear functions and the minimum and maximum of the traction force

FIGURE 5.9: Projected top view on the approximated energy consumption map with the operating point trajectories for different weighting factors.



for higher weightings on the energy consumption. Each of the linear approximating functions represents a certain efficiency ratio (energy consumption per meter) The intersection lines represent the minimum or maximum traction force that can be realised with a certain efficiency. Thus, these points are preferably chosen by the optimiser.

Finally, the average tracking error, the average driving speed and the energy consumption are compared among the control responses with different weightings. These are given in Fig. 5.10. It should be noted that the average values are computed with respect to the trip time (not the travelled distance).

According to Fig. 5.10, the tracking performance decreases with increasing weight on the terminal energy consumption $l_{e,b}$. However, the average driving speed even increases for $l_{e,b} = 1e6$ which results from exceeding the reference speed of 20 km/h (between positions 1000 and 2000 m in the simulation). Since the driving speed is low in that segment, this part contributes significantly to the average result. Furthermore, the energy consumption decreases with increasing weight $l_{e,b}$ as it can be expected. To assess the losses of tracking performance and the gains in the energy-savings more easily, the values are scaled with respect to the performance with a zero weight on the energy

FIGURE 5.10: Comparison of the average tracking error, driving speed and energy consumption throughout the simulation given in Fig. 5.6 with different weighting on the energy consumption.

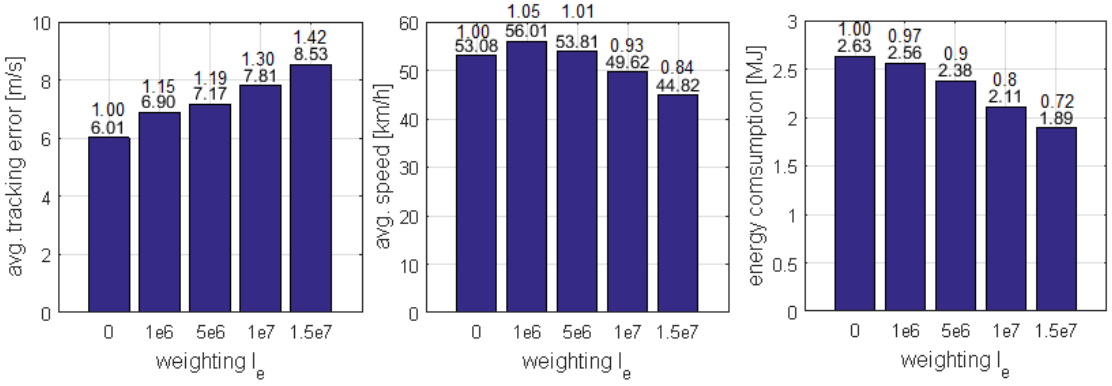
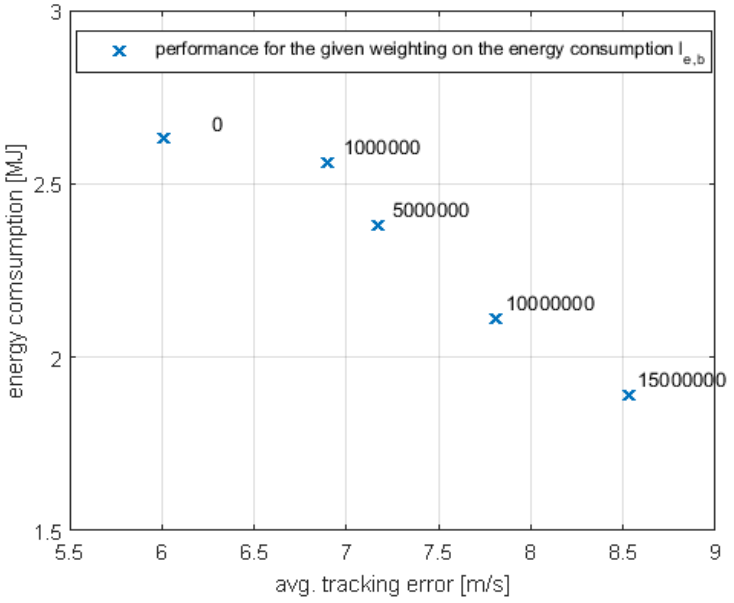


FIGURE 5.11: Tradeoff between the average tracking error and the energy consumption throughout the simulation given in Fig. 5.6 with different weighting on the energy consumption $l_{e,b}$.



consumption $l_{e,b} = 0$ and written on top of the bars in the plot. The tradeoff between tracking performance and energy consumption for the given simulation scenario and the given weighting factors is additionally visualised in Fig. 5.11.

Finally, it is a design decision how much tracking performance can be sacrificed to save energy. As a compromise, the weighting $l_{e,b} = 5e6$ is chosen for further simulations in the sequel.

5.3 Reactions of the Closed-Loop Control in Standard Manoeuvres

In this section, the behaviour of the closed-loop control in different standard driving situations, such as speed reference changes, up and down-slopes and car following is investigated. This investigation is performed with the three different cost functions described in Section 4.3, i. e. the basic cost function, the zone objective function and the cost function to guarantee theoretical closed-loop stability.

5.3.1 Tuning of the Zone Objective Function

The tuning of the zone objective function according to (4.21) (repeated here) can be done based on the following considerations.

$$\min_{\hat{F}_{trac}(k+i|k),\hat{u}_{cons}(k+i|k),\hat{l}_s(k+i|k)} \quad l_{e,z} \cdot \hat{E}_{batt}(k+N_p|k) + \sum_{i=0}^{N_p-1} q_{s,z} \cdot \hat{s}_l(k+i|k)\Big)^2$$

Inside the zone between the minimum and maximum kinetic energy level, the cost function is completely insensitive to the slack variables $\hat{s}_l$ and the only objective is to minimise the electrical energy consumption $\hat{E}_{batt}$. Thus, inside the zone, there is no trade-off among different objectives and the weighting on the terminal energy can simply be set to $l_{e,z} = 1$.

On the other hand, the weighting $q_{s,z}$ on the accumulated slack variables $\hat{s}_l$ must be chosen

- high enough, such that the system state is driven inside the zone whenever it is possible and
- low enough, such that the optimisation problem does not become ill-conditioned and numerically hard to solve.

The value of $q_{s,z}$ is determined in simulations by increasing its value until an immediate enforcing of the soft constraints is given. The value of $q_{s,z} = 0.1$ has been chosen for the following simulations.

The width of the zone has been chosen such that the upper limit is the previously generated kinetic energy reference, generated according to Section 4.6 and the lower limit is 80 % of this kinetic energy reference. This is a design parameter that can be chosen freely according to the acceptable reduction of the driving speed for the sake of energy savings.

5.3.2 Tuning of the Cost Function for Guaranteed Stability

The cost function for guaranteed closed-loop stability is repeated here for the sake of comprehensiveness:

$$\begin{aligned}\min_{\hat{F}_{trac}(k+i|k),\hat{u}_{cons}(k+i|k)} \quad & l_{e,s}\cdot\hat{E}_{batt}(k+N_u)+\bar{q}_{e,s}\cdot\left(\hat{e}_{kin}(k+N_u|k)-\hat{e}_{kin,ref}(k+N_u|k)\right)^2 \\ & +\sum_{i=0}^{N_u}\Big(q_{s,s}\cdot\left(\hat{e}_{kin}(k+i|k)-\hat{e}_{kin,ref}(k+i|k)\right)^2 \\ & +r_{s,s}\cdot\left(\hat{F}_{trac}(k+i-1|k)-\hat{F}_{trac,ref}(k+i-1|k)\right)^2\Big)\end{aligned}$$

The weightings $l_{e,s} = 5e6$ and $q_{s,s} = 1$ are adopted from the tuning for the basic cost function which has been obtained in the previous simulations. As already mentioned in Section 4.3.4, only the system state kinetic energy $\hat{e}_{kin}$ is to be stabilised. Hence, the limit of the series $\sum_{i=0}^{\infty}\left(A_d^T\right)^i Q A_d^i = \bar{Q}$ for $i \to \infty$ is computed. This results in the weighting $\bar{q}_{e,s} = 25.25$.

Furthermore, the weighting on the control input deviation from the reference value must be positive to ensure closed-loop stability. It is determined by simulations and chosen such that the control input is smoothened but the tracking and energy-saving performance is not influenced negatively. A value of $r_{s,s} = 5e4$ is chosen based on these requirements. The reference traction force is the control input which is necessary to drive constantly at the reference speed and can thus be computed as the sum of the driving resistance forces at the given reference speed reference and the given road slopes.

5.3.3 Reactions of the Closed-Loop Control to Speed Reference Changes

In this section, the controllers with the three different cost function types are simulated in scenarios with speed reference changes . First, step reference changes are investigated followed by ramp type references.

5.3.3.1 Reactions of the Closed-Loop Control to Step Reference Changes

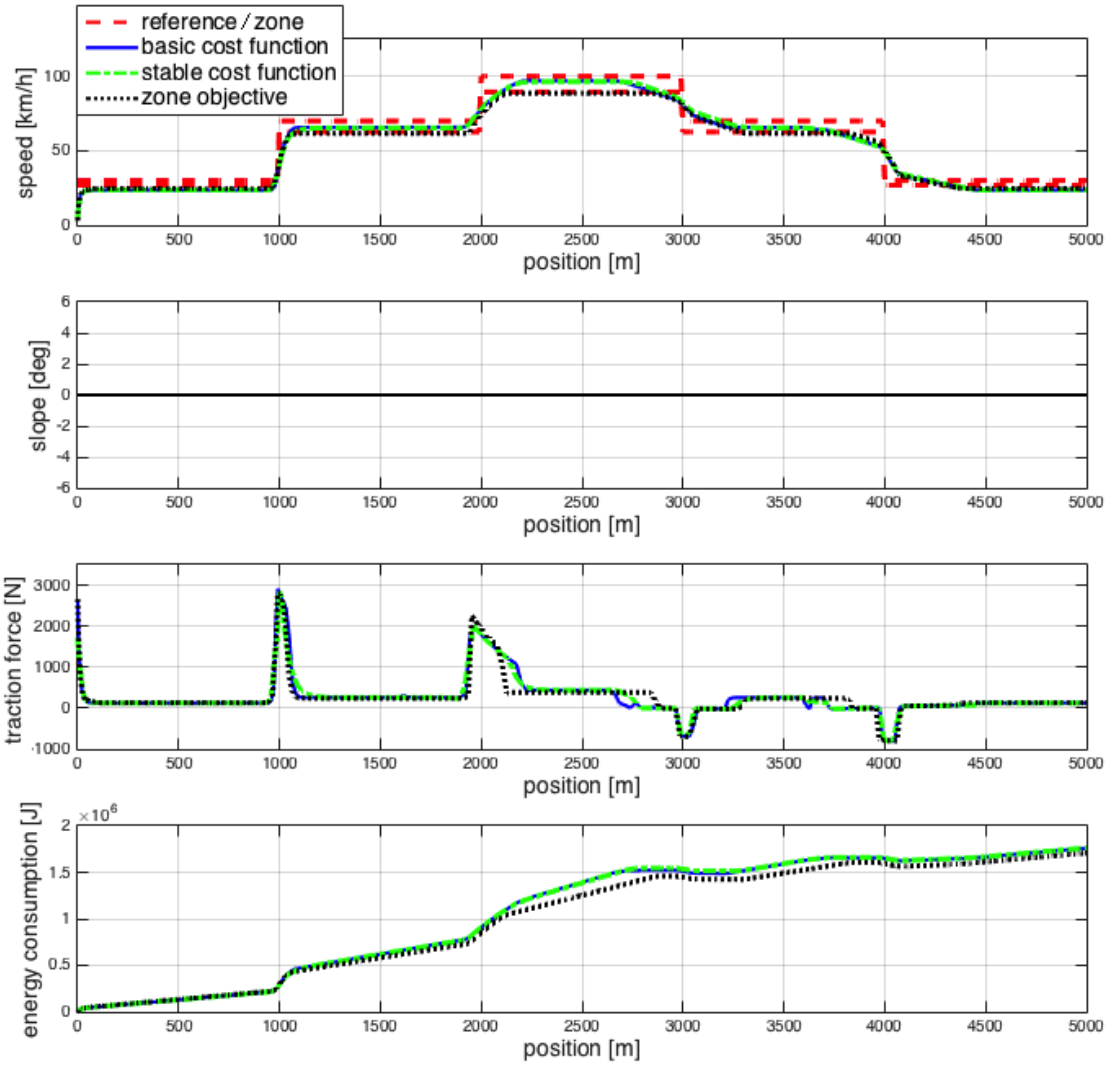
The first scenario to be simulated with the controllers based on different cost functions consists of a series of step reference changes according to the overview in Fig. 5.12. The reference is plotted in dashed red lines together with the lower speed limit for the zone objective.

The first thing to be noted is that the controllers with the basic cost function and that with the cost with theoretically guaranteed closed-loop stability act almost identically. The controller with guaranteed stability applies slight smoother control inputs (traction force) which is mainly due to the weighting on the squared deviation of the traction force from the reference.

Fig. 5.13 provides a closer look on the acceleration manoeuvres in the simulation scenario. Here, it becomes obvious that the controller with the zone objective keeps the speed at the lower limit of the zone in case there are no road slopes in order to save electrical energy. At the same time this controller accelerates as fast as possible at the maximum traction force at every step reference change in order to keep the driving speed within the admissible zone whenever it is possible.

Fig. 5.14 presents the deceleration manoeuvres in detail. It can be noticed that all controllers decelerate in a predictive way. Since no mechanical braking is possible, only comparatively low brake force are possible which necessitate the early decelerations. In summary, the controller with the zone objective achieves the lowest energy consumption but at the same time a lower average speed.

FIGURE 5.12: Results of the closed-loop simulation with step reference changes and different cost function types.

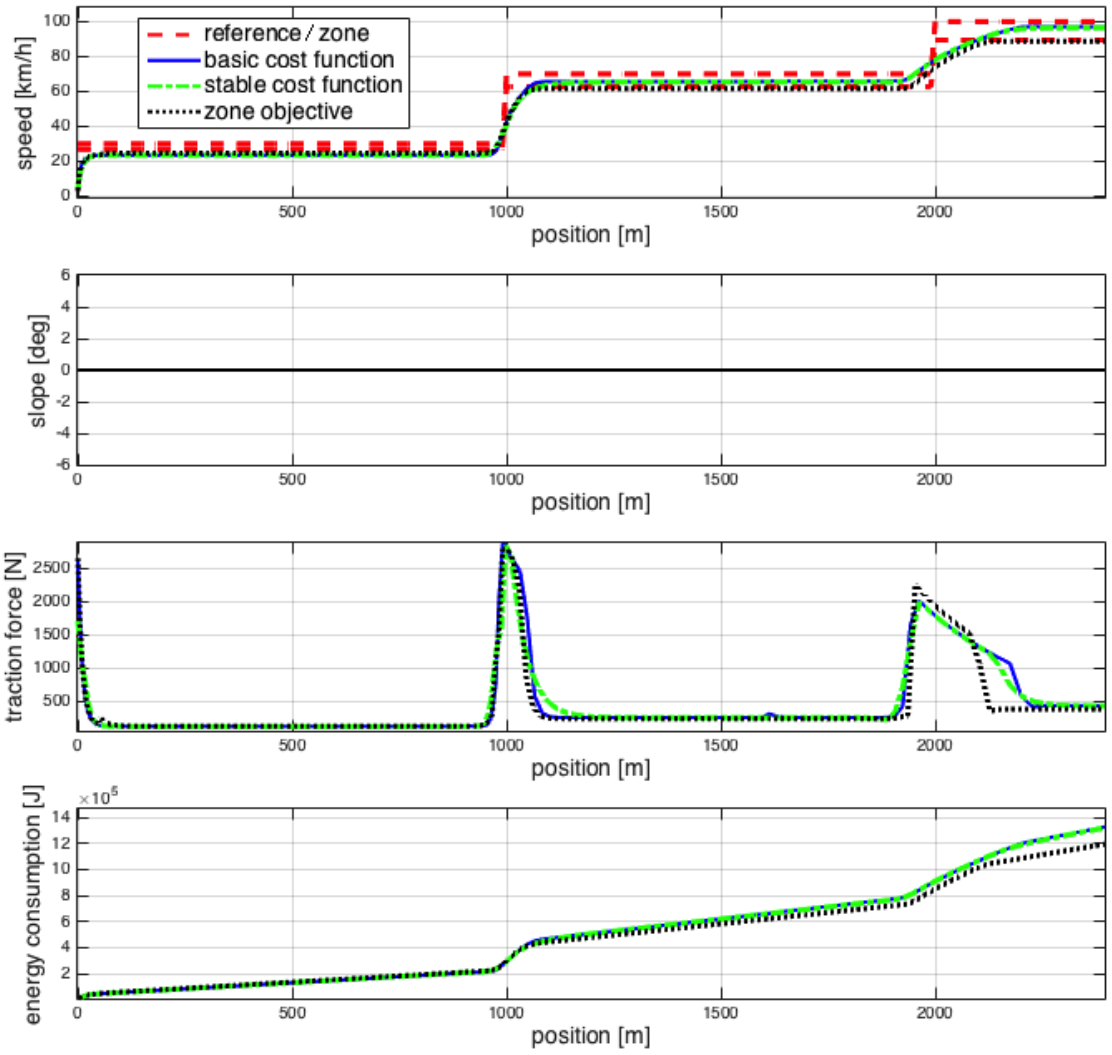


5.3.3.2 Reactions of the Closed-Loop Control to Speed Ramp Reference Changes

This section presents the behaviour of the controller in the case of consecutive linear ramp type speed reference changes. The overview is given in Fig. 5.15. In this scenario, the controller with the basic cost function and the one with guaranteed stability again behave nearly identical.

The acceleration manoeuvres can be seen more in detail in Fig. 5.13. Throughout the first acceleration manoeuvre, all of the three controllers exceed the speed reference and accelerate faster. This is caused by the high energy consumption of the electric machine at low driving speeds, as it can be seen in the energy consumption map Fig. 4.5 on

FIGURE 5.13: Zoom of the simulation results for acceleration manoeuvres with step references and different cost functions.

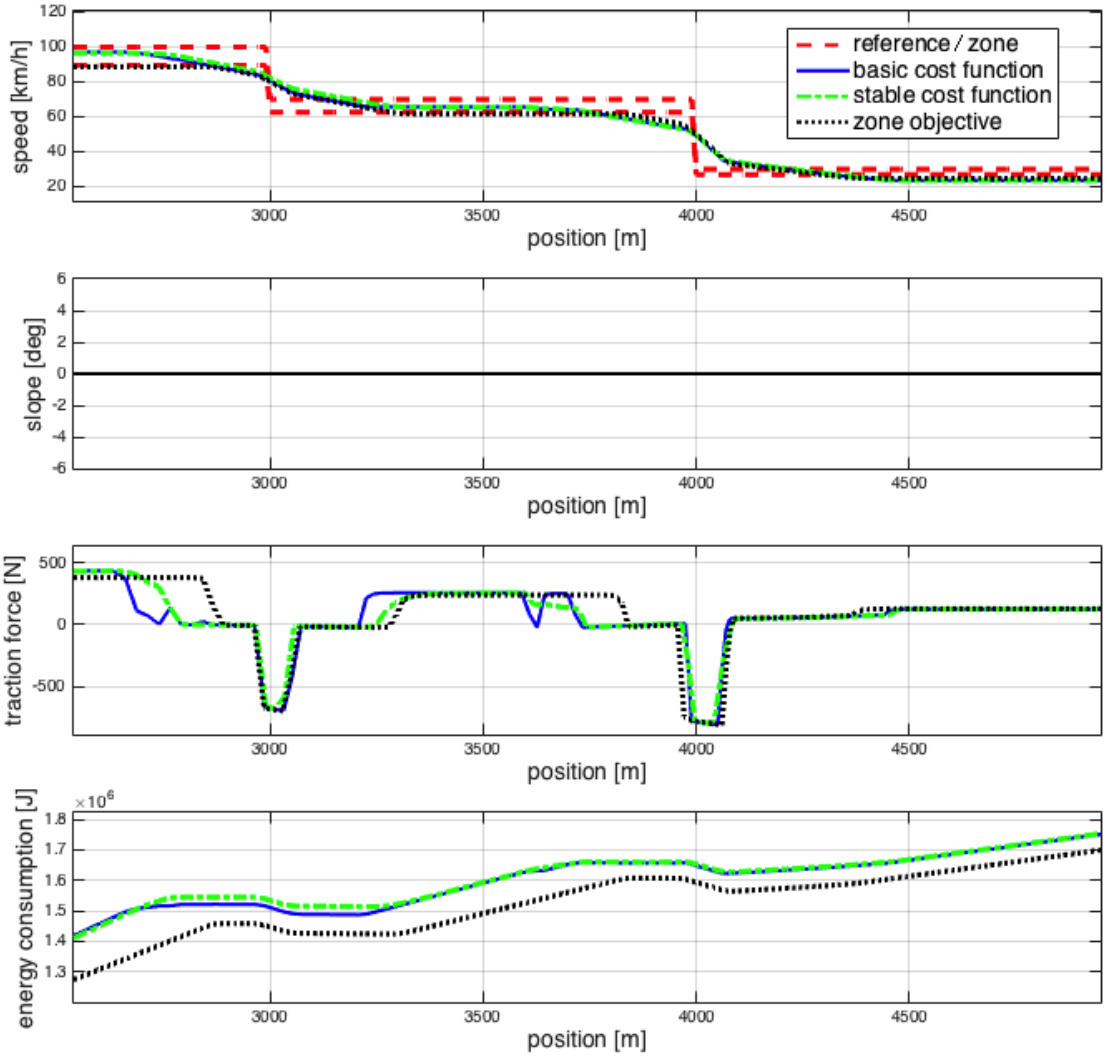


which the prediction model is based. To avoid these operating points, the energy-saving controllers speed-up faster. If this is not desired, it can be avoided by further increasing the weight on the slack variables $\hat{s}_l$ in the zone objective. Apart from this exception, the controller using the zone objective again keeps the driving speed at the lower limit of the zone.

A closer look at the deceleration manoeuvres is provided in Fig. 5.17. It shows that the cost function providing closed-loop stability indeed leads to a smoother control input. In summary, the total energy consumption after 5000 m of driving in this scenarios with ramp reference changes is lower than the consumption in the case of step reference changes but the average driving speed has also been reduced.

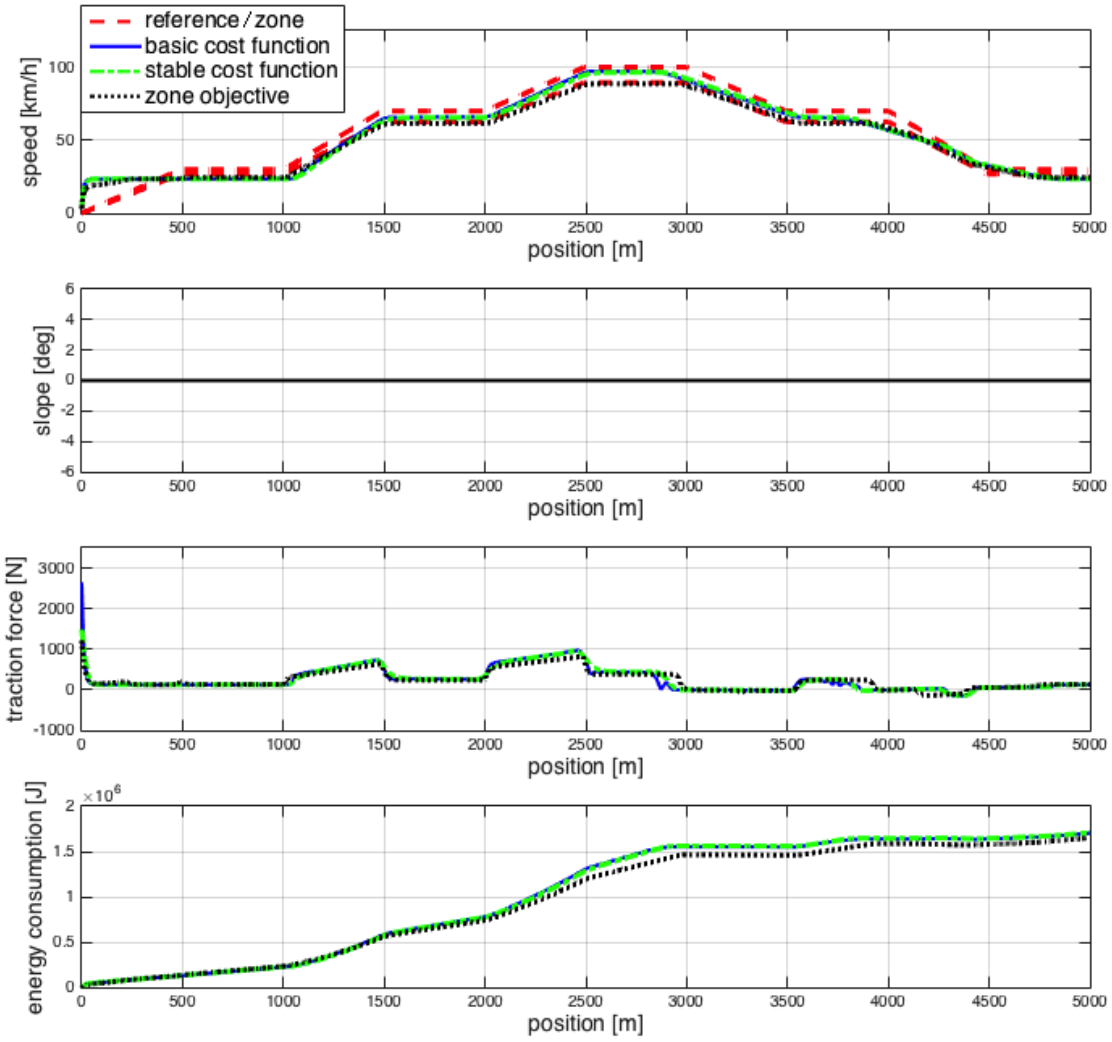
In general, all the controllers do not track the speed reference exactly in order to save

FIGURE 5.14: Zoom of the simulation results for deceleration manoeuvres with step references and different cost functions.



energy. In case of the controller with the zone objective, the deviation depends on the limit of the zone. In case of the reference tracking controllers, the tracking error can be influenced by adjusting the weightings. Generally, the tracking error at higher driving speeds is less than at lower speeds since the controller actually tracks a kinetic energy reference which depends on the square of the driving speed. The same deviation of the speed leads thus to a higher deviation (and a higher penalty) of the kinetic energy at higher driving speeds. The higher penalty enforces a closer tracking.

FIGURE 5.15: Results of the closed-loop simulation with ramp reference changes and different cost function types.



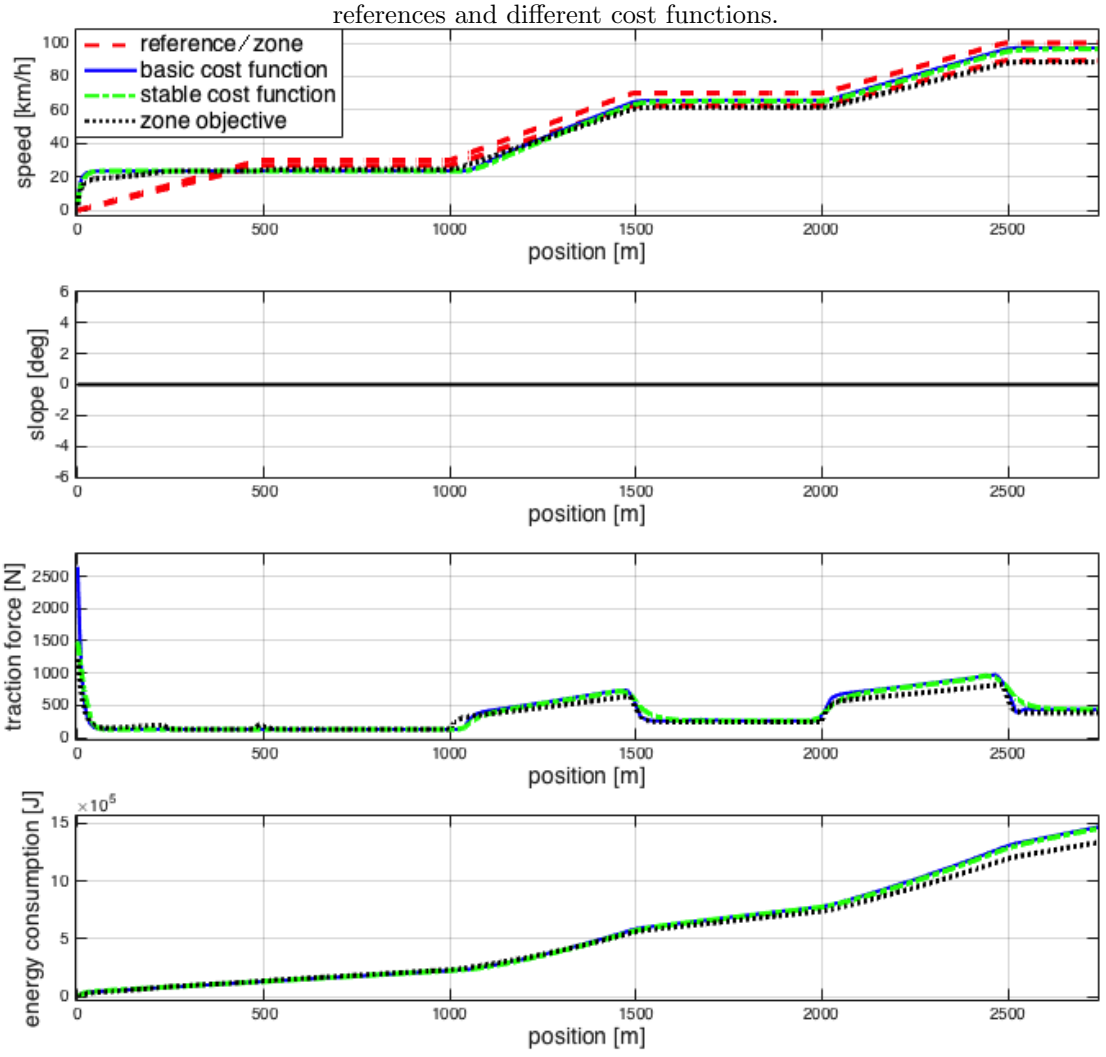
5.3.4 Reactions of the Closed-Loop Control to Road Slopes

This section investigates the reactions of the closed-loop control to different road slopes at different driving speeds of 30 km/h, 70 km/h and 100 km/h.

5.3.5 Reactions of the Closed-Loop Control to Down-Slopes

The first simulation scenario considers a constant reference speed of 30 km/h and starts with driving on an even road (no slope). Then, step-wise down-slopes with a length of 500 m are introduced and followed again by 1000 m of even road before the next slope. The slopes have the angles -1 °, -4 ° and -7 °. Fig. 5.18 presents the simulation results with the three different controller types. The limits of the zone objective have been

FIGURE 5.16: Zoom of the simulation results with acceleration manoeuvres and ramp references and different cost functions.



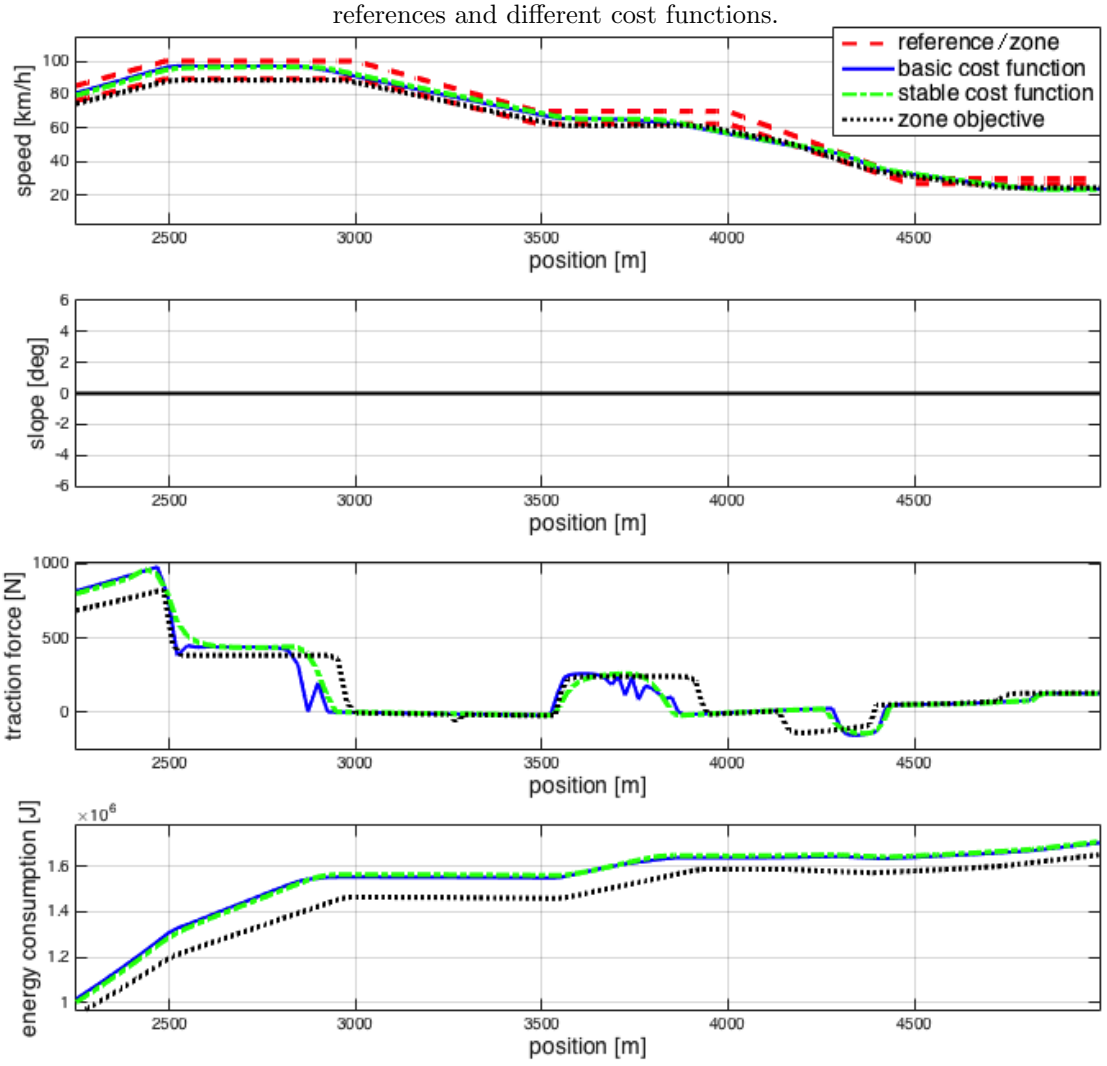
adapted to +5 % of the reference speed (upper limit) and -10 % of the speed reference (lower limit).

The controllers with the basic objective and the one with guaranteed stability decelerate predictively in front of the down-slopes. This deceleration is done in order to accelerate again in the course of the slopes with no traction force effort. In this way, energy can be saved before reaching the slope by reducing the traction force.

Near to the end of the slopes these two controllers accelerate the simulated car further, partially above the speed reference in order to gain momentum and again save energy after the slope by using this momentum.

Throughout the down-hill driving at -7 ° slope angle, the available brake force is not high enough to keep the speed constant. All three controllers apply the maximum brake

FIGURE 5.17: Zoom of the simulation results with deceleration manoeuvres and ramp references and different cost functions.

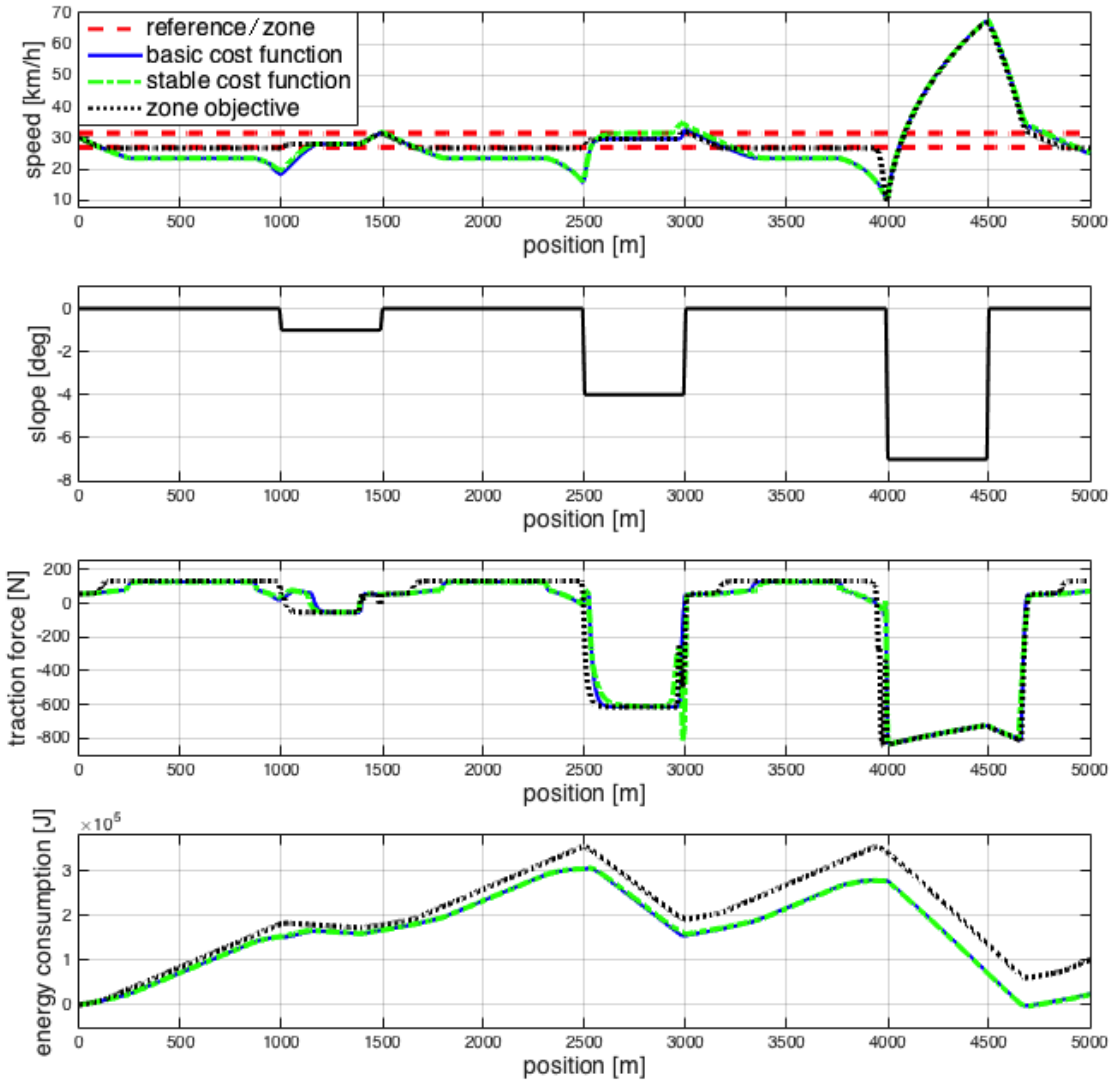


force but exceed the speed reference clearly in spite of this.

For the controllers with the basic objective and the one with guaranteed stability, it can be summarised that the steeper the road gradient angle of the down-slopes, the earlier the deceleration and the lower the driving speed at the beginning of the down-slope.

The controller with the zone objective uses the same principle of gaining momentum during the down-slope, however in a slightly different way. Before reaching the down-slope, the driving speed is already at the lower limit of the admissible zone. This prevents a further slow-down in front of the slope. Throughout the down-hill driving, the driving speed is increased up to the upper limit of the zone in order to use that momentum after the slope. In case of the -7 ° road grade, zone limits cannot be maintained since there is not enough brake force available without using the mechanical brakes.

FIGURE 5.18: Results of the closed-loop simulation with down-hill driving situations at 30 km/h and different cost function types.

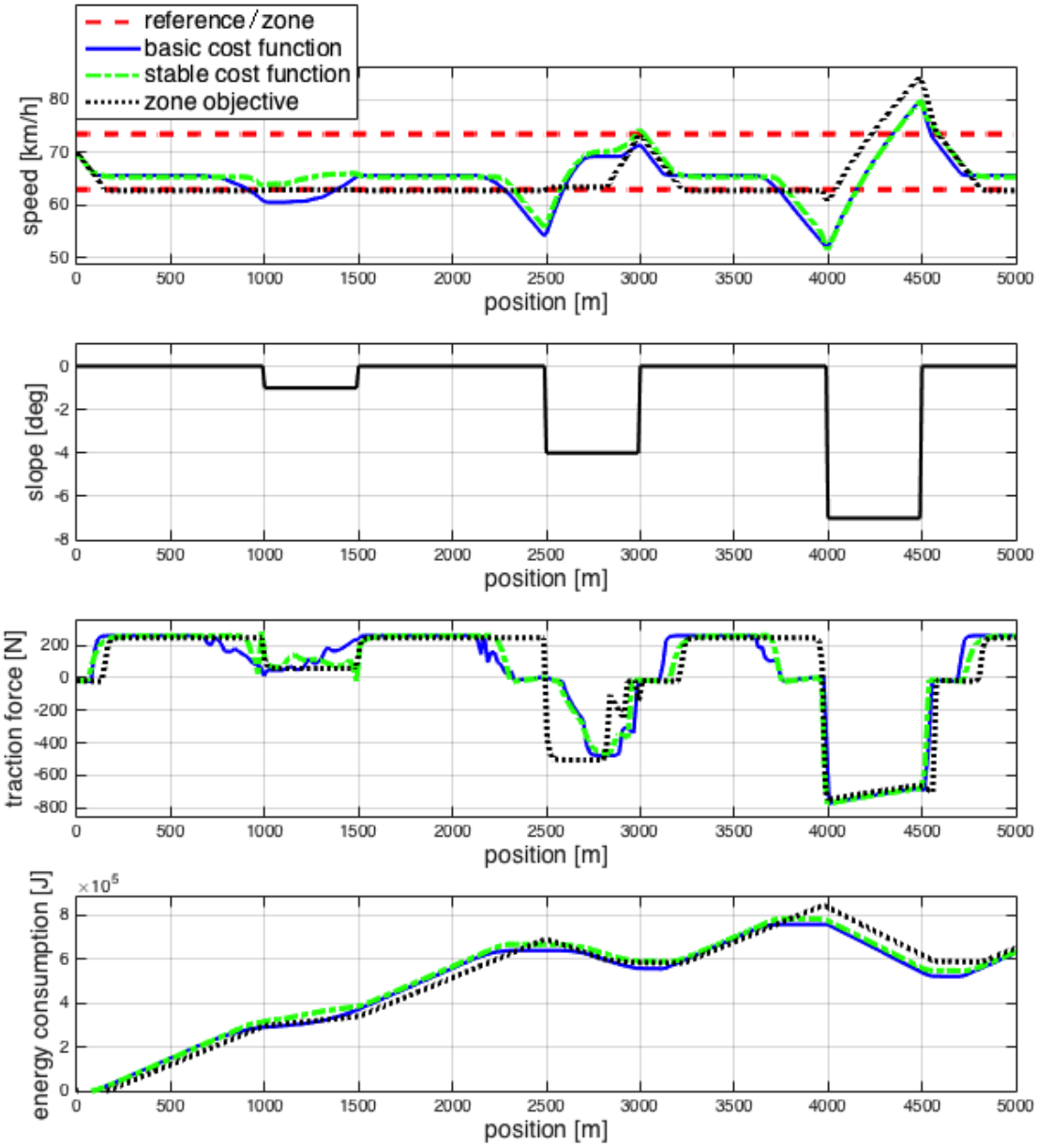


The energy consumption with the zone objective in this scenario is higher than that of the controllers with speed reference since the zone enforces the car to go faster than the others and also prevents a further deceleration before reaching the down-slope. However, this result is only valid for the zone limits chosen here.

The second simulation scenario in Fig. 5.19 consists of the same road slopes as the previous scenario, however the reference speed has changed to 70 km/h.

Again, the two controllers with speed reference tracking decelerate before the slope and accelerate towards the end of the slopes. However, for the slope of -1 °, the simulated car performs only a slight deceleration / acceleration manoeuvre.

FIGURE 5.19: Results of the closed-loop simulation with down-hill driving situations at 70 km/h and different cost function types.



Concerning the other aspects, the control reactions follow the same pattern as in the previous case with a speed reference of 30 km/h. It should be noted that the reference tracking controller with the basic cost function partially applies a slightly oscillating control input which is not the case for the controller with guaranteed stability. This shows that the adjusted terminal weight on the kinetic energy tracking and the weight on the squared deviation from the input reference lead to a smoother control input, indeed.

Fig. 5.20 finally shows the simulation results for the same down-slopes at a speed reference of 100 km/h.

In this, case, the controllers do not slow down considerably for the down-slope of -1 °. since the driving resistance forces at 100 km/h are higher than the down-grade force caused by this slope. Further, the -1 ° road grade section causes an oscillatory control input in case of the reference tracking controller with the basic cost function. This again shows the advantages of the cost function providing stability.

In the case of the -4 ° road grade, the controller with the zone objective does not fully exploit the zone limits to accelerate during the down-slope. It starts accelerating at 2600 m, 400 m before the end of the down-slope (when the end of the down-slope sections appears within the prediction horizon) and it becomes obvious that the momentum can be used after the slope. In the case of the -7 ° grade, the limited brake force causes a speed up when reaching the down-slope.

In the case of a 100 km/h speed reference, the controller with the zone objective yields the lowest energy consumption but also the highest trip time.

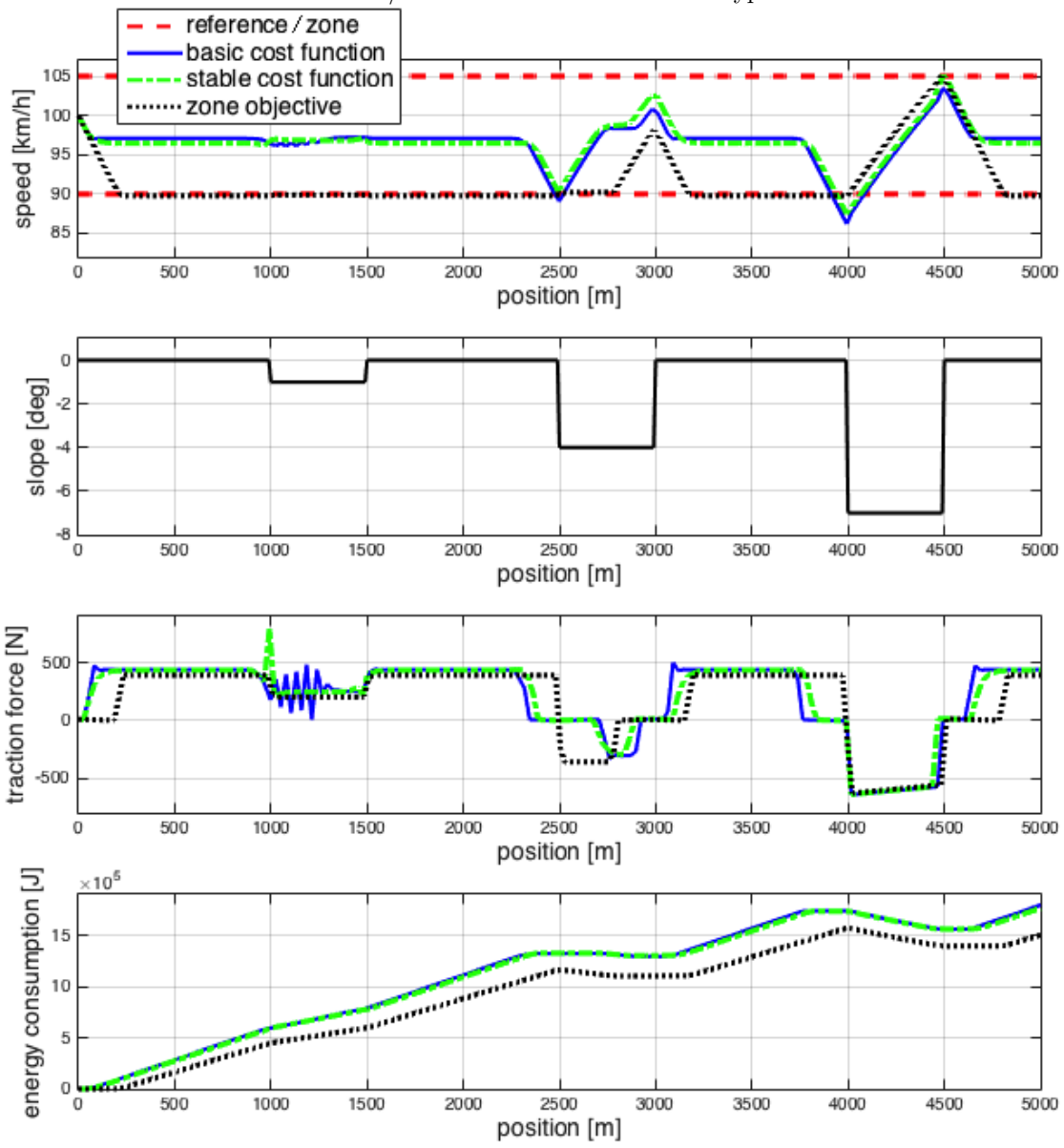
5.3.6 Reactions of the Closed-Loop Control to Up-Slopes

In this section, the control reactions to up-hill driving are investigated. The setting is the same as in the previous section, however the down-slopes are replaced by equivalent up-slopes.

Fig. 5.21 shows the simulation results with a constant speed reference of 30 km/h. The reference tracking controllers (blue and green line) increase the traction force and the driving speed in the road sections with up-hill driving at all three road grades. The controller with a zone objective keeps the driving speed at the lower limit of the zone, decreases the velocity during the +1 ° slope and increases it in the sections with the steeper up-slopes.

The behaviour of the controllers can be explained by looking at the operating points projected on the top view of the approximated energy consumption map as given in Fig. 5.22. The figure shows that driving at the given speeds between 25 and 30 km/h without up-slope (and consequent low traction force) leads to operating points on the

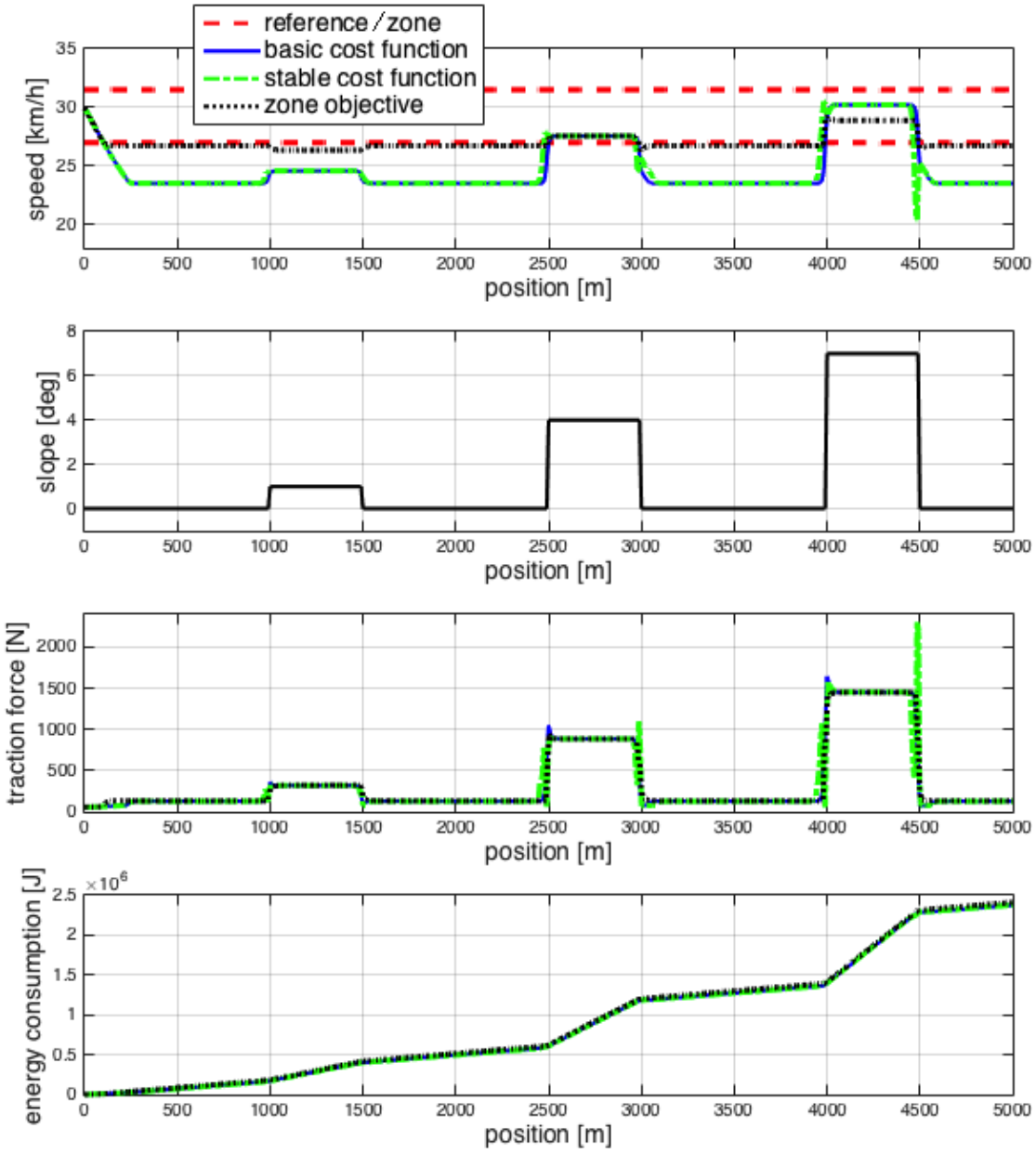
FIGURE 5.20: Results of the closed-loop simulation with down-hill driving situations at 100 km/h and different cost function types.



yellow plane which are rather efficient. However an increase of the traction force which became necessary to keep that speed in case of up-hill driving would move the operating point to the orange plane which would increase the energy consumption strongly. For this reason, the energy saving controllers increase the driving speed to reach operating points on the more efficient yellow plane, again.

Fig. 5.23 presents the simulation of the same up-hill driving at a reference speed of 70 km/h. Here, the controllers with reference tracking objectives decrease the driving

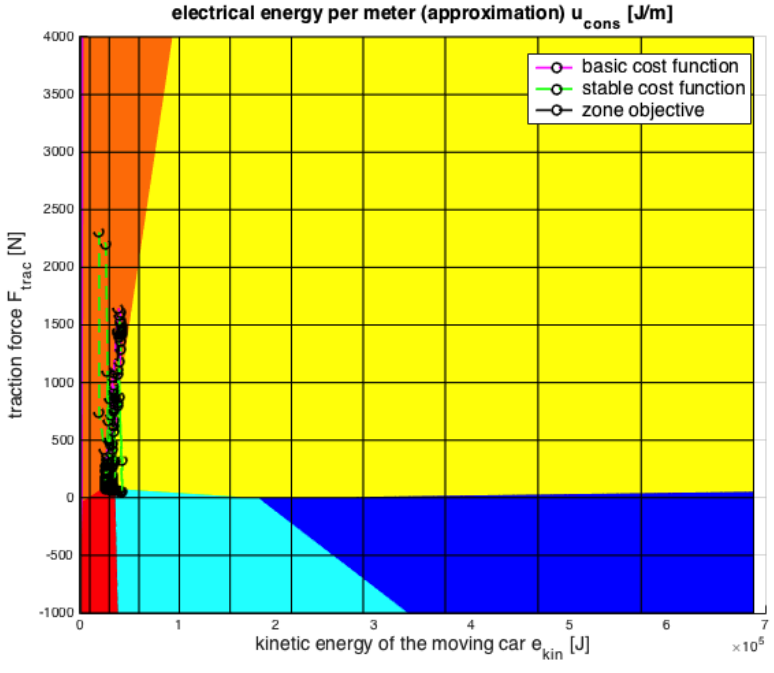
FIGURE 5.21: Results of the closed-loop simulation with up-hill driving situations at 30 km/h and different cost function types.



speed in the sections with up-grades to lower the traction force in these section. Especially in case of the controller with basic cost function, this is preceded by a slight increase of the speed before reaching the up-hill section to gain momentum. The controller with the zone objective lowers the speed throughout the up-hill sections as well and even violates the zone limits.

The operating point trajectory (Fig. 5.24) reveals that the operating points at 70 km/h are always on the (efficient) yellow plane for positive traction force. Thus, an increase of the driving speed during up-slopes like in the case of very low driving speeds does

FIGURE 5.22: Projected top view on the approximated energy consumption map with the operating point trajectories during up-hill driving at 30 km/h with different cost function types.



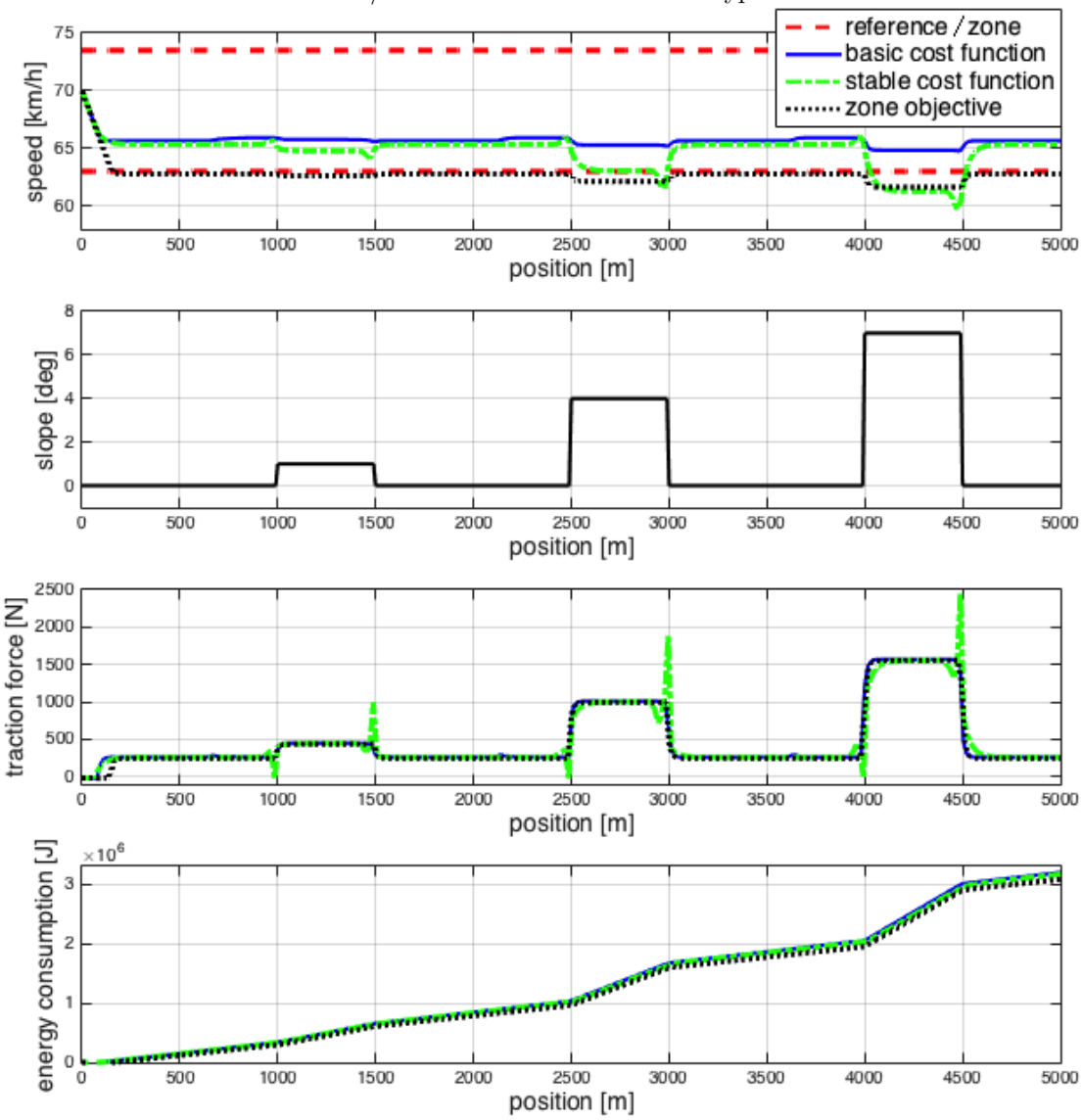
not make sense here. Consequently, the energy-saving mechanism for up-hill driving depends on the driving speed.

Finally, Fig. 5.25 presents the results for a reference of 100 km/h. As already mentioned in Section 5.3.3, deviations from higher speed reference lead to a higher penalty due to the controller model formulated in terms of kinetic energy depending on the square of the driving speeds. Thus, tracking the speed reference becomes more important at these higher speeds than it was in the previous scenarios.

As a consequence, the controllers do not vary the driving speed significantly in case of the +1 ° up-grade. During the +4 ° up-hill section, the two reference tracking controllers reach the maximum traction force limit and cannot keep the driving speed. Thus, the speed drops throughout the slope. This becomes even more evident in the +7 ° slope where the controllers speed-up the simulated vehicle predictively before the slope to avoid the driving speed from dropping too low.

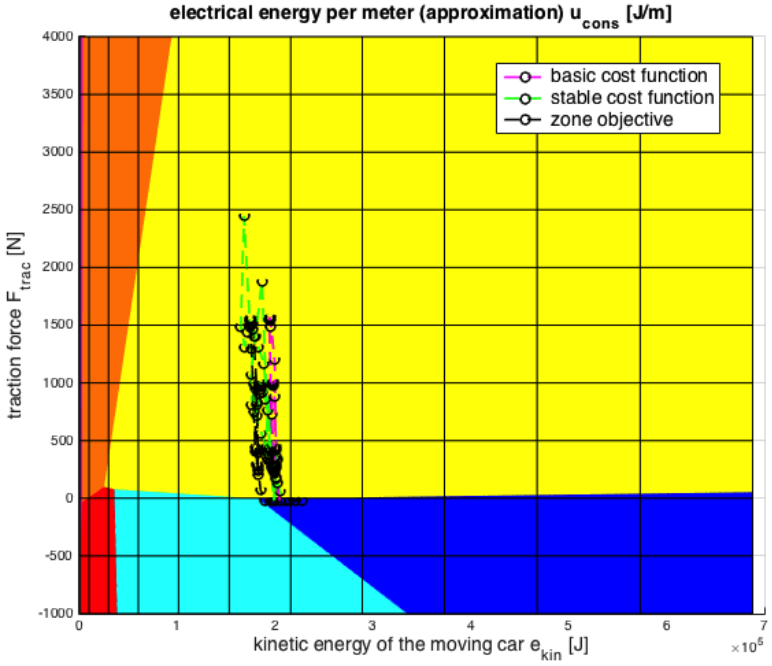
The operating points for this scenario in Fig. 5.26 show that the linear maximum traction force limit is reached with all three controllers in this scenario.

FIGURE 5.23: Results of the closed-loop simulation with up-hill driving situations at 70 km/h and different cost function types.



In summary, the behaviour of the energy-saving controllers in scenarios with up-hill sections depends on the reference speed. At low driving speeds, the up-slopes can lead to an increase of the speed during the slopes to reach more efficient operating points. At medium speed reference, the slope can lead to a decrease of the driving speed in order to reduce the traction force and the consumption. At high driving speeds, the slope can either lead to no changes in the speed (slight up-slopes), a decrease of the driving speed or even an increase before reaching the slope to avoid a too significant reduction of the speed during the slope.

FIGURE 5.24: Projected top view on the approximated energy consumption map with the operating point trajectories during up-hill driving at 70 km/h with different cost function types.



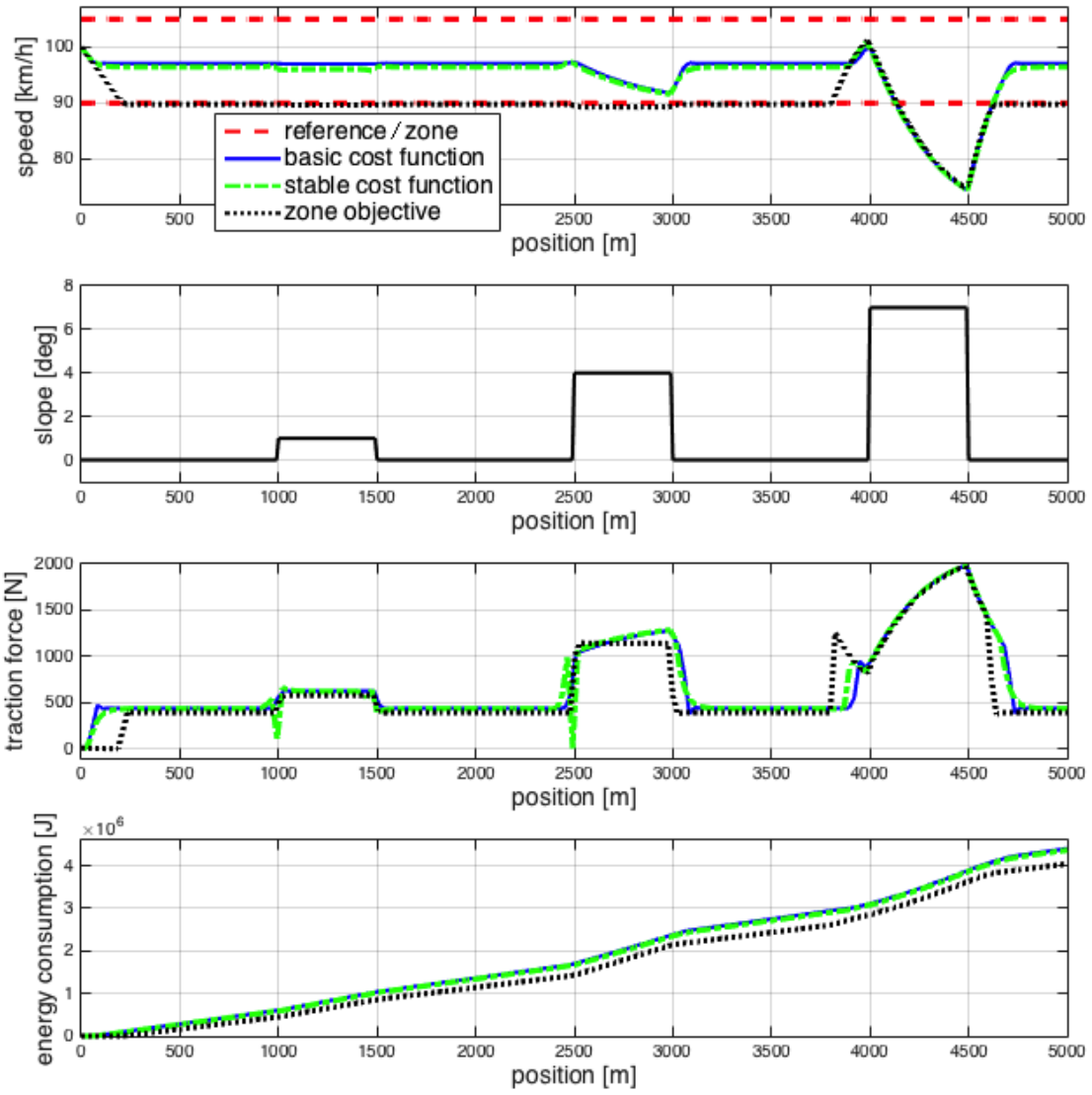
5.3.7 Car-Following Behaviour

The subsequent simulations investigate the behaviour of the vehicle in closed-loop control when a preceding car is present. The distance to a preceding car and its driving speed influence the closed-loop control through the speed reference. It should be noted that the changes of the preceding car speed are not known in advance and thus cannot be included in the prediction of the MPC controller with their true future values. Only the current preceding car speed can be measured and is assumed to be constant throughout the whole prediction horizon. The speed reference generation is then performed depending on the relative speed and relative distance to that car as described in Section 4.6.

The first simulation scenario examines a case where the speed of the preceding car is constant at 50 km/h and the initial speed of the host vehicle is 100 km/h. Furthermore, an up-slope section of a steep 4 ° slope angle followed by a -4 ° down-slope is introduced. The results are depicted in Fig. 5.27.

At the beginning of the simulation, the speed of the host vehicle is higher than the speed of the preceding car and the distance is above the safety distance ($d_{h,safe} = 25\ m$ for the speed $v_h = 50\ km/h$). Consequently, an exponentially decreasing speed reference

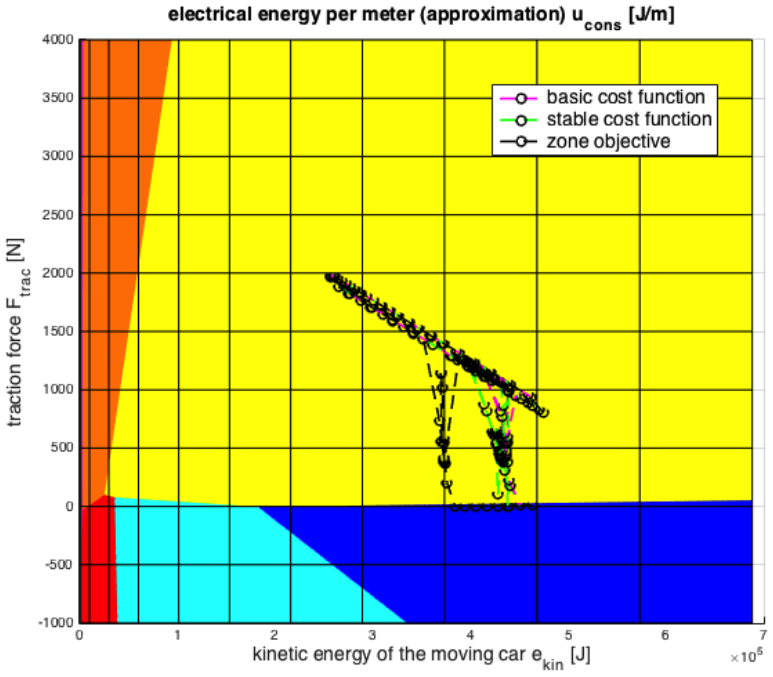
FIGURE 5.25: Results of the closed-loop simulation with up-hill driving situations at 100 km/h and different cost function types.



is fed to the MPC according to Section 4.6, the simulated driving speed drops and the distance approaches the safety distance $d_{h,safe}$.

When the speed of the preceding car is met by the host vehicle, the speed reference is constantly set to the current speed of the preceding car. In this case, the energy-saving cruise controllers fall below this speed reference to safe energy which causes the distance between the cars to increase again. As long as the speed of the host vehicle is below the speed of the preceding car and the headway distance is between the safety distance $d_{h,safe}$ and 200 m, the speed reference is kept constantly at v_p. In this case, the distance between the cars increases up to 200 m (the maximum range of the radar sensor). When this distance is exceeded, the preceding car is not considered thereafter.

FIGURE 5.26: Projected top view on the approximated energy consumption map with the operating point trajectories during up-hill driving at 100 km/h with different cost function types.

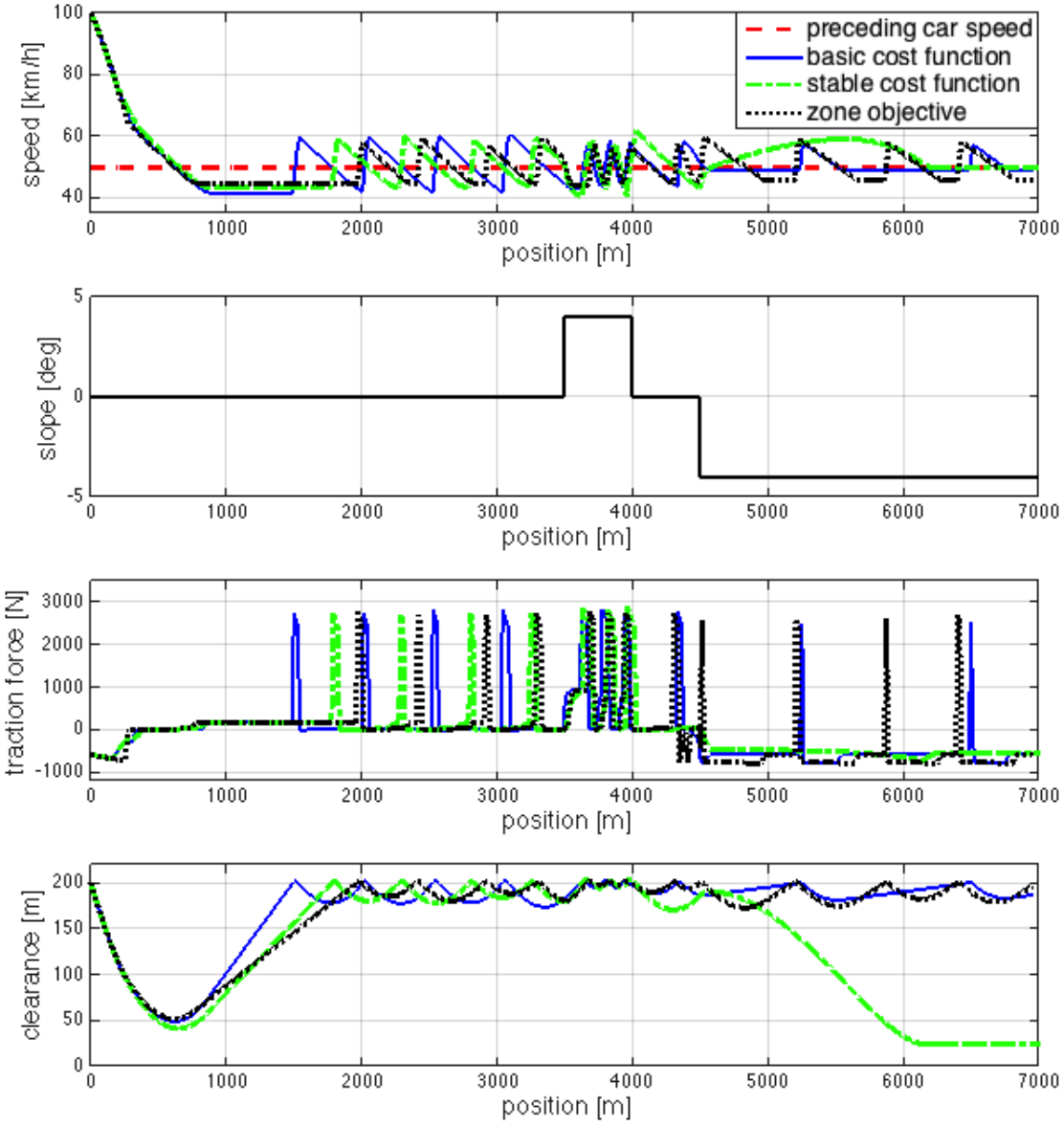


Consequently, the speed reference ist not influenced by that car anymore and set to 100 km/h and the host vehicle accelerates. When the distance drops again below 200 m, the speed reference is reduced again, the driving speed of the host vehicle decreases and the headway distance increases. This process is then repeated and causes a slowly oscillating speed. The subsequent up-slope section increases the frequency of the speed oscillations. However, the period of these oscillations is relatively long (approximately between 250 and 500 m of travelled distance) and thus it does not lead to low driving comfort.

While the behaviour of the three energy-saving controllers is qualitatively similar for the even road and the up-slope segment, their behaviour differs for the down-slope. The controller with the zone objective and the one using the basic cost functions cause oscillations of the speed. In contrast, the controller with the cost functions for guaranteed stability tracks the speed of the preceding car and the safety distance exactly throughout the down-slope.

In contrast to the previous simulation scenario where the speed of the preceding car is constant and the road slope varies, the following simulation results show a case where the preceding car changes its speed in the form of ramps or steps on an even rod segment

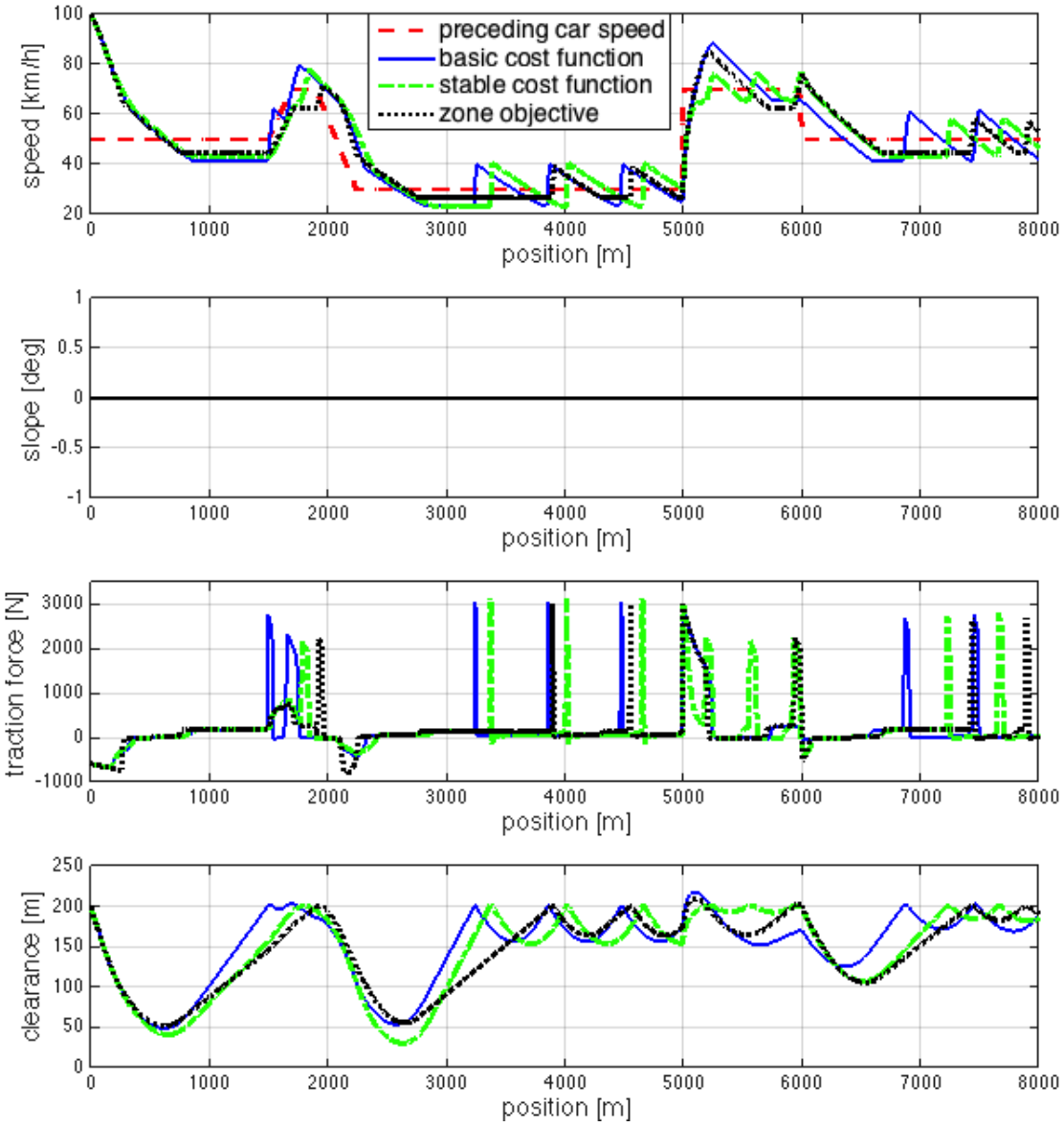
FIGURE 5.27: Results of the closed-loop simulation for following a preceding car driving constantly at 50 km/h with up- and down-slopes and different cost function types.



as presented in Fig. 5.28. The preceding car drives at first at 50 km/h when it is approached by the host vehicle. At a position of 1500 m, it begins to change its speed to 70 and consecutively to 30 km/h as a ramp function of the position and keeps the speed of 30 kmh up to a position of 5000 m. Then, the speed of the preceding car raises step-wise to 70 km/h over 1000 m and finally steps down to 50 km/h again.

The simulation with the three energy- saving cruise controllers shows that the preceding car speed is tracked by the host vehicle and the headway distance does not go below the safety distance. In the segments with constant reference speed, the host vehicle speed

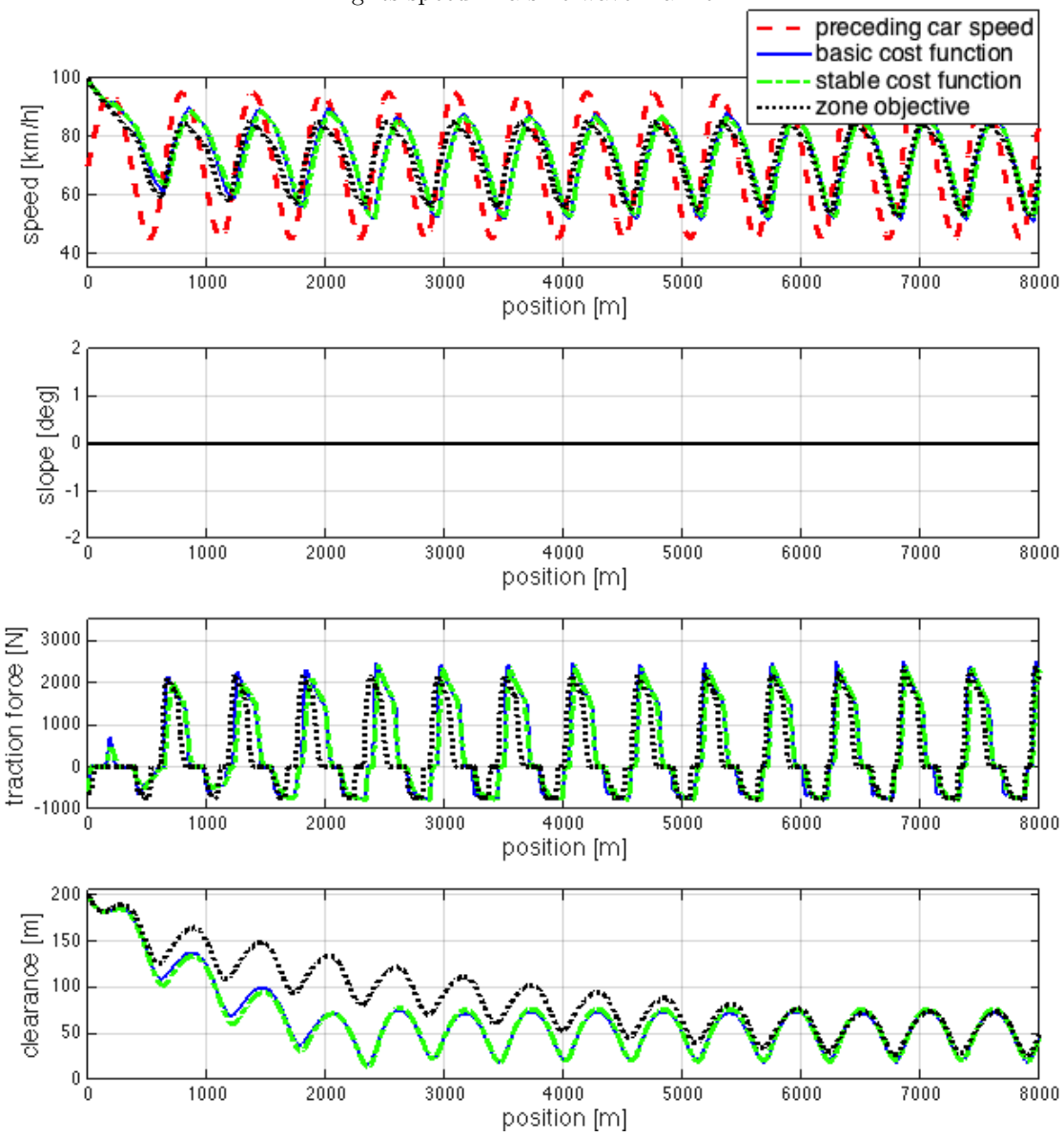
FIGURE 5.28: Results of the closed-loop simulation for following a preceding car changing its driving speed.



oscillates around the speed of the preceding car but again with long periods of more than 500 m.

Finally, a scenario were the speed of the preceding car varies in a sine wave manner with an average of 70 km/h and an amplitude of 25 km/h is simulated. This car is approached by the host vehicle with an initial speed of 100 km/h. With all three controllers, the speed of the host vehicle takes a sinusoidal form with the same period as that of the preceding car, however with a lower amplitude and a phase shift / delay of approximately 100 m. This causes the headway distance to vary in a sinusoidal form as well. While the two controllers with the basic and stable cost function show practically the same

FIGURE 5.29: Results of the closed-loop simulation for following a preceding car changing its speed in a sine wave manner.



behaviour, the speed using the controller with a zone objective approaches that of the preceding car slower.

In all simulated cases, a safe distance to the preceding car is maintained. Regarding the oscillatory behaviour in segments with a constant speed of the preceding car, it must be kept in mind that the speed of a preceding car as well as the road slope angle and the speed limit will unlikely be exactly constant which will cause variations in the speed in any case. Furthermore, it must be considered that the eco-cruise control system is not meant to maintain a constant distance to a preceding car. After approaching it

safely, a subsequent increase of the distance provides additional freedom for energy-saving driving maneuvres like gaining momentum during down-slopes. In unforeseen or emergency situations like an abrupt stop of the preceding car, the eco-cruise control system requires the driver's intervention in unforeseen or emergency situations since the brake pedal is not actuated.

5.3.8 Comparison of the Presented Approach to a Non-Linear Formulation

To show the effectiveness of the reformulations applied in this work to achieve a linear MPC formulation with quadratic cost function, a comparison is made to a non-linear energy-saving cruise control formulation presented in the literature. For this comparison, the work [12] is chosen that presents closed-loop simulation results of a non-linear energy-saving cruise controller in up- and downhill driving scenarios.

The NMPC controller in [12] is designed for fuel-powered cars, however without considering gear shifting which makes the approach comparable to the one developed in the at-hand thesis to some extent.

The road profiles and simulation data have been extracted from the plots in [12] for an up-slope, a down-slope, a down-up-slope and an up-down-slope scenario and have been used to simulate the same scenario with the presented linear MPC approach and re-plot the results in one figure. To achieve a better comparability, the prediction horizon and weightings have been adapted. In [12], the optimal control problem is formulated as a function of the time t with a prediction horizon of 10 s. At a driving speed of 50 km/h as given in the simulation examples, this is equivalent to 140 m of travelled distance which is used as prediction horizon in the simulation, here (14 sample steps of 10 m). The comparison is carried out with the cost function for guaranteed stability where the weightings are adjusted to $l_{e,s} = 5e5$, $\bar{q}_{e,s} = 25.25$, $q_{s,s} = 1$ and $r_{s,s} = 523$ due to the shortened horizon.

Fig. 5.30 shows the simulation results for a down-slope (left) and an up-slope scenario (right) using the controller developed in this work (continuous lines) as well as the simulation results in [12] (dotted lines). Additionally, pure cruise control with a zero

weighting $l_{e,s} = 0$ on the energy term in the cost function (black lines) is compared to energy-saving driving (green lines).

In the case of the down-slope, the pure reference tracking cruise controllers compensate the effect of the down-slope by using a traction force input proportional to the road slope to track the constant reference speed. In contrast, the energy saving cruise controllers try to use the momentum that can be gained throughout the down-slope. The speed profile of both the linear and the non-linear energy-saving MPC cruise control approach show qualitatively the same shape. Before reaching the down-slope, the driving speed is predictively reduced. During the down-slope, the momentum is regained and after the down-slope, the constant steady driving speed is approached again. Looking at the control input traction force, the non-linear eco-cruise control shows a smoother input trajectory while the linear approach tries to keep the traction force constant throughout the down-slope. This can be explained by using a piecewise linear energy consumption map leading to optimum operating points at the intersections of the linear functions (cf. the previous simulation examples). However, the driving speed is equally smooth and comfortable in both cases.

In the up-slope scenario (Fig. 5.30, right), the pure cruise controllers again track the constant reference speed by using a traction force trajectory proportional to the road slope. The NMPC eco-cruise control increases the speed before the hill and reduces it during the up-hill segment. The linear MPC approach presented in this work slightly decreases the speed during the up-hill segment without an increase before. This can again be explained with the help of the piecewise linear consumption map as given in Fig. 4.5. For driving at a constant speed of approximately 50 km/h, the necessary increase of the traction force to overcome the up-hill at 50 km/h does not cause operating points on another linear function and hence does not change the drive train efficiency. Thus, it is not beneficial to gain momentum before the hill to overcome the hill at lower traction force levels.

Fig. 5.31 presents the simulation results of a down-up and an up-down hill scenario. Here, the controller reactions are qualitatively the same as in the preceding example and can be explained with the same reasoning.

FIGURE 5.30: Comparison of the given linear control approach to a non-linear MPC formulation for fuel-powered cars presented in [12] for a down-slope (left) and an up-slope (right) scenario.

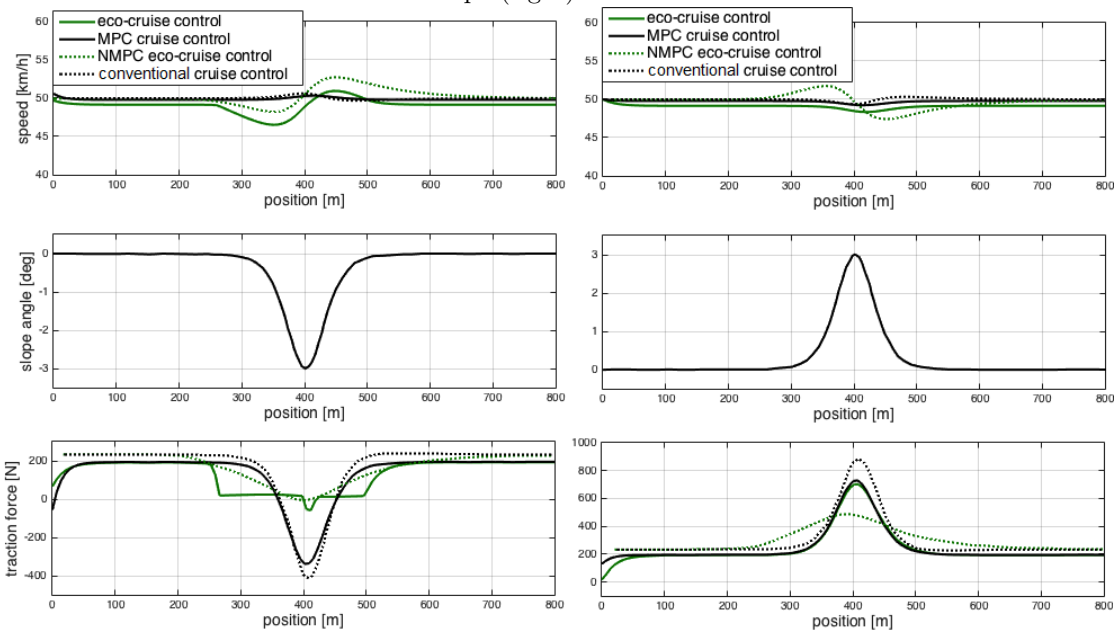
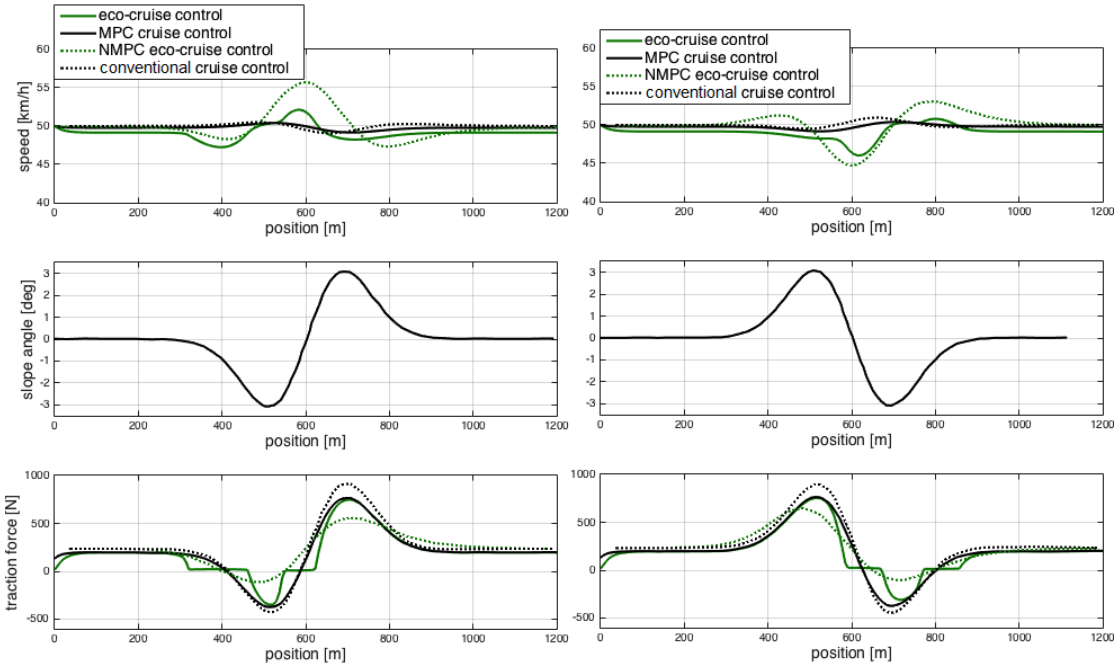


FIGURE 5.31: Comparison of the given linear control approach to a non-linear MPC formulation for fuel-powered cars presented in [12] for a down-up-slope (left) and an up-down-slope (right) scenario.



5.4 Computation Time

The maximum computation time requirements for the implementation can be estimated in different ways. On the one hand, the control system should be able to update the control action at least as fast as a human driver which corresponds to the human reaction time of approximately 0.3 s. On the other hand, the prediction step size in the underlying controller model has been chosen to 10 m and the control decision must thus be updated before travelling the distance of 10 m at the maximum driving speed of the *Smart ED*. At the maximum driving speed of 125 km/h, the travelling time for a distance of 10 m is 0.29 s. Both requirements indicate more or less the same demand for the maximum computation time. Since the application of a cruise controller is safety critical, the control action shall be updated significantly faster than this maximum requirement and a sample time of 0.1 s is demanded in this case.

The dimension of the optimisation problem is defined by the number of optimisation variables (in this case 80) and the number of inequality constraints (in this case 360). The discretised problem has the form of a convex quadratic program. Convex quadratic programs can be solved in polynomial time [108] which is a good basis for a real-time algorithm.

[144] provides a study on the real-time capabilities of embedded quadratic programming using the C-code generator *CVXGEN*. For a model-predictive control problem with large dimensions (336 optimisation variables, 252 inequality constraints), a computation time of 13 ms is achieved on a comparatively weak *Intel Atom* PC with a peak power consumption of 2 W. The comparison of the presented model-predictive control approach in the work at hand to this reference allows the conclusion that the proposed problem formulation is real-time capable. However, if necessary, the number of prediction steps can be reduced or techniques like input move-blocking can be applied to obtain a lower-dimensional problem.

The computational time to solve the given optimisation problem within *MATLAB* on a desktop PC (Intel Core i7, 2.2 Ghz) running under Windows 7 is approximately 0.1 s (see Fig. 5.5) during the related simulation. Thus, a real-time computation is achieved on this hardware platform. Solving the same problem with a C-code based quadratic programming solver, a significantly faster computation can even be expected.

The computation time to generate the speed reference according to the procedure in Section 4.6 has shown to be at least one order of magnitude lower than the time to solve the MPC problem and is thus negligible in comparison.

5.5 Conclusion of the Chapter

The simulation examples in this chapter have presented the control performance in standard situations such as the reaction to reference changes, road slopes and car following with different cost function types and different predictions horizons and weightings. Furthermore, a comparison to a non-linear MPC formulation is made.

The chapter starts by investigating a simple cost function being the sum of kinetic energy tracking error and electrical energy consumption. The simulations with different prediction horizons indicate that the control performance does not increase significantly for prediction horizons above 800 m. As a compromise between computational complexity and control performance, a prediction horizon of 400 m is the most reasonable choice, here.

As a matter of fact, the simulation runs with different weightings show that an increased weight on the energy consumption leads to larger deviations from the speed / kinetic energy reference. The choice of the weighting is a design question and depends on the requirements of the driver.

The investigation of standard situations has revealed several energy saving mechanisms of the eco-cruise control. The basic saving mechanism is the reduction of the driving speed (when there is no down-slope). Lower driving speeds reduce the air drag resistance force, and then, driving at lower traction forces and consequently lower energy consumption is possible.

When a down-slope is encountered, the energy-saving cruise control aims at gaining momentum during the down-slope by slowing down before reaching it and speeding up again throughout the down-slope without traction force effort.

Concerning up-slopes, the reaction of the energy-saving cruise controller depends on the (reference) driving speed. At medium and higher driving speeds, the speed is reduced during up-slopes to lower the traction force effort and the energy consumption. At low

driving speed, the energy-saving cruise control may accelerate during up-slopes to avoid inefficient operating points of the electric drive system.

This corresponds to the energy-saving mechanism of choosing operating points at the intersections of the linear functions approximating the energy consumption map. The intersection lines represent those traction forces, where an increase or alternatively a decrease of the traction force is not possible without stepping to another drive train efficiency. Thus these borderlines of machine efficiency are likely to be part of the optimum solutions. Furthermore, very low traction force values close to zero (i. e. freewheeling without accelerating or decelerating) avoid electrical losses in the drive train.

When comparing the aforementioned energy saving techniques to those of fuel-powered cars, some of them are identical like the use of the momentum gained during down-slopes and the preferred operation in efficient operating points. However, there are also decisive differences to the efficient operation of fuel-powered cars. Firstly, the electrical drive train of the *Smart ED* remains in efficient operation for a wide range of operating points whereas the efficiency of fuel-powered vehicles strongly depends on the operating point. Furthermore, the investigated car like many electric vehicles has a fixed transmission ratio and thus does not make use of gear-shifting. Because of these two properties, the use of a smart gear shifting and acceleration strategy to keep an efficient engine operation is not applicable for electric cars in many cases and the main energy saving mechanism is the reduction of the driving speed (going along with a reduction of the air drag resistance and the necessary traction force). Additionally of course, electrical vehicles have the possibility of recovering energy by regenerative braking. During the simulation, this feature is only used in the case when braking cannot be avoided.

The study of the control design with different cost functions types (basic cost function, cost function extended for closed-loop stability and zone objective function) has revealed that using the basic cost function can indeed lead to oscillations in the control input. This issue can be resolved by using a cost function providing theoretical closed-loop stability with an appropriate weight on the terminal state and on the control input. On the one hand, the zone objective provides a better control on the acceptable speed margins. On the other hand, the flexibility of the controller to plan energy-saving manoeuvres is lower with this type of cost function and sometimes leading to a higher energy consumption.

For these reasons, the "stable" cost function is chosen for further investigations in the sequel.

While this chapter has mainly investigated standard hypothetical driving situations, the next chapter will investigate driving on a real road in combination with experiments to show the control performance in realistic scenarios.

Chapter 6

Practical Implementation of the Controller and Experimental Results

To validate the results of this work, the proposed model-predictive control system is experimentally implemented in a *Smart ED* series-production vehicle. Since driving on public roads with modified vehicles is not allowed, all experiments using the automatic cruise controller are performed on the closed test track of the *Centre de Formation pour Conducteurs S.A.*, Colmar-Berg, Luxembourg, depicted in Fig. 6.1.

The track has a total length of 1.255 km and includes turns and two circular paths as well as a 9 % ascent / decent (depending on the direction of driving).

To realise the predictive energy-efficient cruise control on this track, the road gradient angles, the road curvature and speed limits are required. These data are extracted from different sources. The basis is a bird view image of the track taken from *Google Maps* [16]. With the help of the webpage *Geocontext.org* [17], the latitude and longitude coordinates of the test track are extracted. Since the predictive energy-saving cruise control is based on a one-dimensional prediction model, the travelled distance on the track is defined as a position coordinate s beginning at the starting point at the entrance of the track and the vehicle is assumed to move forward in terms of this coordinate. Furthermore, the road curvature is extracted from the road topology of the bird view image and approximated

FIGURE 6.1: Test track of the *Centre de Formation pour Conducteurs S.A.*, Colmar-Berg, Luxembourg [15].

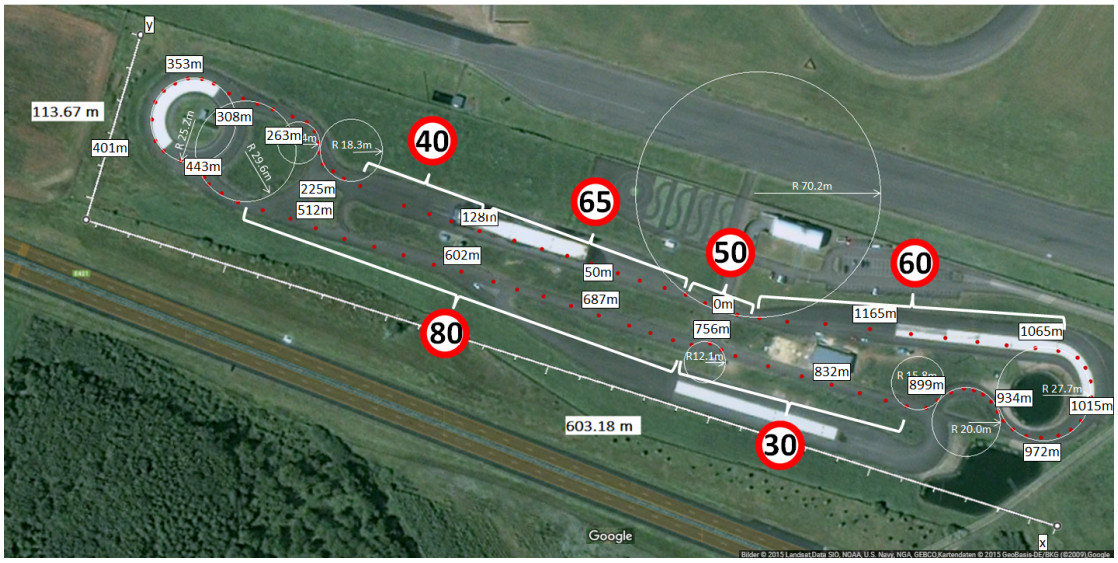


FIGURE 6.2: Test track of the *Centre de Formation pour Conducteurs S.A.*, Colmar-Berg, Luxembourg with position coordinate and approximate road curvature represented by circles. Image modified based on map data by [16, 17]

.

by circles. The bird view map of the track together with the curve radiuses r_c and the position coordinate s are given in Fig. 6.2.

The road slope angles are finally derived by recording the altitude measurements from a GPS sensor on-board the car during a calibration drive along the track. The gradient angles a_{sl} are computed from two subsequent altitude measurements h_1 and h_2 at two

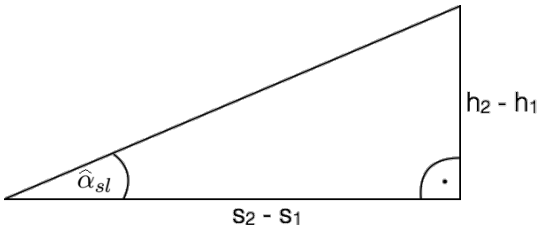


FIGURE 6.3: Approximate calculation of the road slope angle in driving direction from two subsequent altitude and position measurements.

subsequent positions on the track s_1 and s_2 by a simple triangle approximation according to Fig. 6.3.

$$\hat{\alpha}_{sl} \approx \arctan\left(\frac{h_2 - h_1}{s_2 - s_1}\right) \tag{6.1}$$

Finally, speed limits are introduced in front of the circular curves and the 9 % ascent. Additionally, a maximum speed of 80 km/h on the long straight area has been introduced according to Fig. 6.2 to keep the driving experiments safe.

In summary of the results above, the relevant road information (kinetic energy reference $\hat{e}_{kin,ref}$, road slope angle $\hat{\alpha}_{sl}$ and road curve radius $\hat{r}_c$) for the predictive energy-saving cruise control system can be stored in a lookup table as a function of the position coordinate s. When the vehicle position on the track is known, this table can be used to extract the relevant reference and disturbance data in an anticipatory way for the whole prediction horizon (i. e. for the current and the upcoming positions).

6.1 Conception of the Implemented Control System

After making the road data available, the concept of the control system implementation in the *Smart ED* is described in this section. To achieve a high flexibility and a high adaptability in further projects, a powerful car PC platform with a Linux operating system is chosen. The computer can be operated with a voltage of 12 V and be mounted in the trunk of the car. On top of this, the freely available open-source *Robot Operating System (ROS)* [145] is run on the system to facilitate the (soft) real-time execution of

TABLE 6.1: Layout principle of the lookup table linking latitude and longitude data with the position coordinate s on the test track.

latitude	**longitude**	**position s**
49.1	6.1	0
49.1001	6.105	10
...	...	...

the different scripts to realise the cruise control system, the data exchange among them and to simplify the recording and analysis of the measurements.

Within the *ROS* framework, C-Code or Python Code can be used to define executables (so-called *nodes*) that are executed at a certain sample rate and exchange data in the form of messages (so-called *topics*) among them. These topics can be recorded in data files (so-called *ROSbags*) that can be replayed after the experiment to analyse the measured data and eventual errors. Available functionalities such as sensor hardware drivers (so-called *packages*) can be downloaded from an open repository and included in the system. With these features, *ROS* is a useful tool to run scripts in (soft) real-time and record and analyse the measured data.

6.2 Localisation of the Vehicle on the Track

The next step to enable the predictive cruise control in experiments is the localisation of the vehicle on the track. For this purpose, the vehicle is equipped with a GPS sensor providing a position measurement with a tolerance of around +/- 10 m. This positioning accuracy is acceptable here since the road parameters vary only slightly within distances of 10 m. To find the position on the track in terms of the coordinate s (cf. Fig. 6.2), the track positions are linked to their related GPS positions (latitude and longitude) beforehand in a lookup table with the layout given in Tab. 6.1.

After receiving a GPS measurement during the control operation, the Euclidean deviation of this measurement from all latitude and longitude pairs in the table is computed and the pair with the smallest deviations is taken as the current position based on the GPS measurement s_{GPS}.

Since GPS measurements are only received with a sample rate of 1 s, a simple correction via odometry is made according to the following equation to determine vehicle position between two GPS measurements.

$$s \approx s_{GPS} + t_{GPS} \cdot v(t_{GPS}) \tag{6.2}$$

According to (6.2), the elapsed time t_{GPS} since the last GPS measurement (< 1 s) multiplied with the driving speed $v(t_{GPS})$ at that time approximates the travelled distance since the last measurement and is added to the position on the track s_{GPS} linked with the last GPS measurement.

During the practical tests, very robust and reliable positioning results have been achieved with this very simple localisation approach.

6.3 Measurement of the Vehicle State

Besides the vehicle position on the track, the driving state of the vehicle must be measured to apply and analyse the energy-saving cruise control system, i. e. the driving speed v, the battery current I_{batt} and voltage U_{batt}, the state of charge SOC as well as the accelerator and brake pedal positions.

All these data can be read from the vehicle without installing additional sensors by reading the *on-board diagnose (OBD)* interface of the car. It is linked to the *Controller Area Network (CAN)* bus interconnecting different sensors, actuators and electronic control units (ECUs) in the car. This data is accessed by using a CAN card attached to the car PC and read in real-time throughout the control operation.

The measurement of the driving speed v is necessary to compute the initial value of the kinetic energy $\hat{e}_{kin}(k|k)$ in the optimal control problem while the other data serves to analyse the control performance.

6.4 Online Solution of the Optimal Control Problem

With the results of the previous sections, the initial state according to the current driving speed, the reference and disturbance trajectory related to the actual position on the track, the optimal control problem can be set up according to the methodology given in Section 4.5. For this purpose, the matrices H_{QP}, w_{QP}, $E_{ineq,QP}$, $F_{ineq,QP}$ and $g_{ineq,QP}$ of the quadratic program need to be set up. For an easier working with the matrices, the freely available C-Code library *Eigen* [146] is implemented.

The solver *QPOASES* [139] which is freely available as a C-Code library is used to solve the quadratic program. Finally, optimisation problem (4.31) with the selected cost function for guaranteed stability can be solved online throughout the driving tests.

6.5 Realisation of the control commands

The optimal control input variable computed by solving (4.31) is the traction force. However, the traction force can neither directly be measured nor set in the real car. Instead, the control commands are realised by actuating the accelerator pedal. That way, the inner control system of the car remains untouched such that the legislation of the car for public roads and the safety measures of the manufacturer are kept.

The accelerator pedal position however can be regarded as a "traction force demand" and thus the pedal position can serve to realise the traction force commands. The actual traction force related the actual driving speed and pedal position has been measured in the course of the dynamometer measurements presented in Chapter 3. This relation can be inverted to read the pedal position that goes along with the demanded traction force and the actual driving speed (as measured from the OBD interface). The characteristic map is plotted in Fig. 6.4. Using this relation as a lookup table, the required pedal position can be read based on the traction force as calculated by the MPC controller.

One way to realise the desired pedal position is to actuate the pedal electro-mechanically with the help of an electric servo-motor. This however goes along with delays in the actuation and with further necessary modifications of the car. A more elegant way is to disconnect the accelerator pedal and to imitate the electric signals generated by the

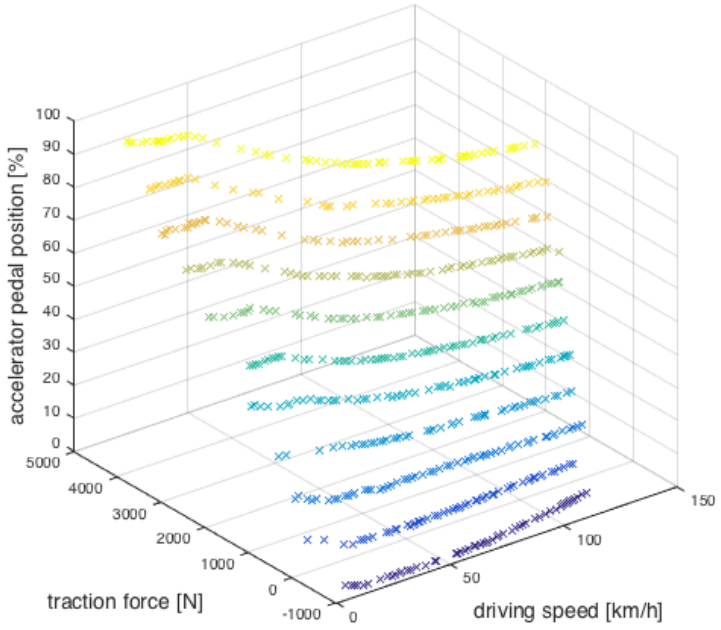


FIGURE 6.4: Measured relation between the accelerator pedal position, the driving speed and the traction force at the wheels.

position sensors in the accelerator pedal to "pretend" an artificial pedal position to the vehicle ECU.

An investigation of the signal type and behaviour of the accelerator pedal shows that the pedal provides two analog voltage signals at the pedal connector that vary proportionally with the pedal position. The two voltages always have a ratio of 2:1 and serve the purpose to detect faults in the pedal position encoder. If the 2:1 ratio is not given, a technical fault is detected by the ECU of the vehicle. The accelerator pedal and its connector layout are depicted in Fig. 6.5.

In conclusion, it is possible to mimic an artificial accelerator pedal position by generating these voltage signals and feeding them to the pedal connector. However, it is important to maintain the 2:1 ratio in the voltage level exactly synchronous over time.

To achieve this, a freely programmable *Teensy 3.1* microcontroller [18] based on the Arduino platform is used to receive the desired voltage level (representing a pedal position) from the car PC through its USB interface and to output a proportional voltage at its analog output port. This analog voltage is amplified by an operational amplifier circuit to achieve the two necessary voltage levels in the proper ratio. By generating one

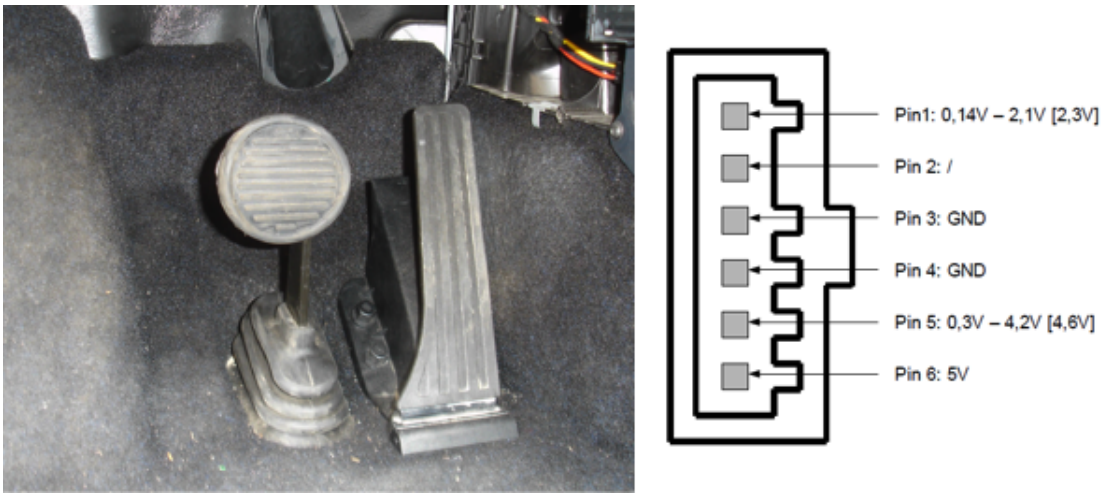


FIGURE 6.5: Accelerator and brake pedal of the *Smart ED* together with the accelerator pedal connector pin assignment.

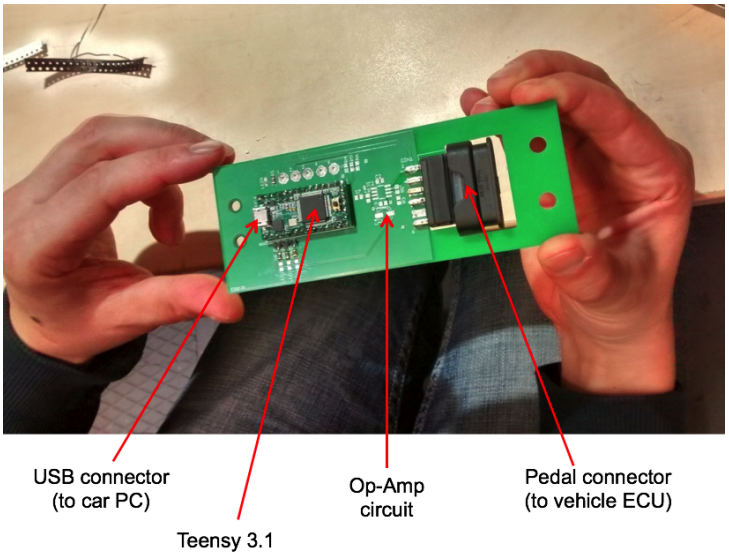


FIGURE 6.6: The actuator board to mimic an artificial accelerator pedal position throughout the driving tests including a *Teensy 3.1* microcontroller [18] together with an operational amplifier circuit and an interface for the pedal connector.

voltage at first which is then amplified, it is ensured that the two resulting voltages are exactly synchronised over time and no faults are generated.

The *Teensy 3.1* microcontroller is included on a printed circuit board together with the operational amplifier circuit and the accelerator pedal connector of the vehicle is connected to this board (instead of the real accelerator pedal) throughout the driving experiments on the test track. The board is displayed in Fig. 6.6. While using this device as an automatic accelerator pedal, the brake pedal can still be used by the driver in emergency situations.

With the complete experimental setup, the position of the vehicle on the track can be determined, the initial driving state of the car can be read, the reference and disturbance trajectory can be extracted and the optimised accelerator position can be realised such that the MPC controller can finally be tested and validated in practical experiments.

6.6 Results of the Practical Measurements

The first plots in this section present experimentally measured driving data of the *Smart ED* running on the CFC test track at Colmar-Berg using the energy-saving cruise control with the cost function for guaranteed stability (4.32). These measurements are compared to simulation results with the vehicle on the track using the same slope, curve and speed reference profile. The purpose of this comparison is to investigate the accordance of simulation and practical measurement. In both cases, the car starts at standstill and then continuously runs two laps on the track. The speed reference, road slope and curvature profiles are plotted in Fig. 6.7. The position coordinate starts at the same point as indicated in Fig. 6.2. The simulations are carried out in *MATLAB / Simulink* as described in Chapter 5 while the measurements are done using the *ROS* setup described in Chapter 6. In contrast to the previous sections, the weighting factor on the energy consumption is reduced to $l_{e,s} = 2.5e5$. All other parameters are kept the same.

The comparison of the driving speed shows a very high accordance of measurement and simulation. Only at the beginning of the measurement, the acceleration of the real car is slower than the simulated one.

Since there is no direct way to measure the traction force in the experimental setup, the presented values are the control inputs computed by the MPC. Comparing the control inputs in the simulation and the experimental setup, the matching is again very high except at the start of the experiment where the traction force is lower than the simulated one.

The high accordance between measurement and simulation concerning the speed as well as the control input traction force shows that the dynamic prediction model is sufficiently accurate since the real car accelerates and decelerates very closely to the predicted behaviour. Furthermore, it shows that the computed traction force is properly realised

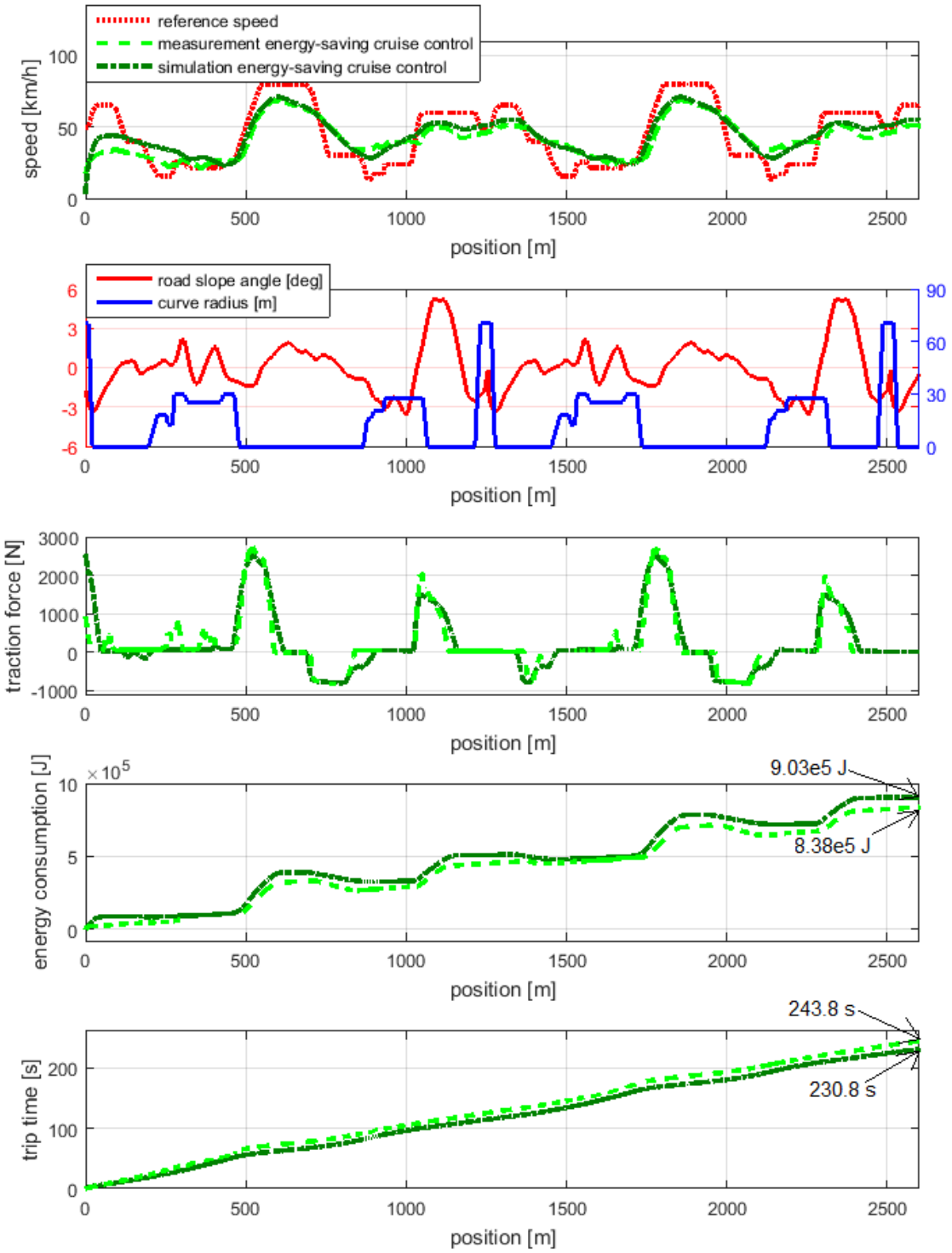


FIGURE 6.7: Experimental measurements of driving data on the CFC test track (2 laps in sequence) at Colmar-Berg of the *Smart ED* under eco-cruise control compared to a simulation of the same scenario.

using the conversion to an accelerator pedal position according to the lookup table given in Fig. 6.4.

The small differences in the traction force trajectories between measurement and simulation however can result from two sources. Firstly, due to small inaccuracies in the prediction model and the road slope profile, the vehicle state is not exactly the same as in the simulation which then causes a different initial state measurement in the subsequent sample step and consequently a different computed traction force. Secondly, the solvers used to compute the solution of the quadratic programs are different in simulation and experiment. The simulation uses the *MATLAB* algorithm *quadprog()* while the experiment uses the toolbox *QPOASES* [139]. This may explain slightly different results and different acceleration behaviour at the beginning of the trajectories.

The power consumption in the simulation is computed with the help of the characteristic map Fig. 3.9 while the measured data is the product of the measured battery current and voltage. The accordance of the two trajectories is high - however the measured energy consumption values are slightly lower than the simulated values which can be due to inaccuracies in the model Fig. 3.9 or also because the car in the experiments was driving slightly slower than in the simulations (cf. the trip times).

An energy saving mechanism which is evident in both simulation and experiment, is the tendency to keep the traction force close to zero (at the intersection line of the yellow and the (light) blue plane in Fig. 4.4). In these operating points, the electric motor is neither consuming nor recovering electrical energy and the car is freewheeling. This avoids electric losses in the electric motor and the power electronics and can thus be considered to be efficient. In the second lap on the track, the vehicle using the energy-saving cruise control accelerates only two times throughout the lap with a maximum traction force at positions 1777 m and 2312 m and recovering energy only twice around the positions 1383 m and 2000 m. The rest of the track is done at freewheeling.

The effect of the significant slopes of the track on the control behaviour can hardly be seen since the steepest slope on the track (encountered after 1083 m and a second time after 2339 m) is preceded by a tight curve with a radius of 28 m, which does not give any freedom to adapt the speed. The other slopes on the track are less steep and their effect is thus minor.

In conclusion, it can be stated that the accordance of practical experiment and simulation is very high which confirms the dynamic prediction model and the experimental setup. In addition, the energy saving mechanism of freewheeling can be shown. The next step is to compare the energy-saving cruise control to the driving style of human drivers.

For this purpose, the following measurements present three human driving styles of two different drivers ("driver A" and "driver B") and the vehicle using the energy-saving cruise control in two different settings on the CFC test track in Colmar-Berg. Again, the car starts at standstill and then drives two laps on the track.

The human drivers are familiar with the car and told to follow the same speed references as the energy-saving cruise control given in Fig. 6.2. The energy-saving speed control is tested with two different weightings on the energy consumption: "full eco-driving" at $l_{e,s} = 5e5$ and "soft eco-driving" at $l_{e,s} = 2.5e5$.

Fig. 6.8 presents the speed profiles of the different drivers and controllers. The fastest laps are done by "driver A" and "driver B", however by exceeding the speed reference at some places. In these two runs, the drivers try to drive normally without a special focus on the energy consumption. After this experiment, "driver A" repeats the driving with a stronger focus on observing the speed limits and driving energy-efficiently. The results of this test run are plotted in black ("driver A, eco mode"). These three different driving styles are compared to the energy-saving cruise control, plotted in light green with the lower weight on the energy consumption and in dark green with the higher weighting factor.

The tractions forces applied by the human drivers have been computed by using the (inverted) characteristic map given in Fig. 6.4. The comparison of the traction force trajectories show high variations in the traction force applied by the human drivers while the energy-saving cruise controllers apply constant (and very low) traction forces over long periods of the track. Apart from reductions in the energy consumption, this leads to a comfortable and smooth driving.

Comparing the energy consumption, "driver A" has spent the highest amount of energy by far (1.83 MJ) which is almost three times more (!) than the energy-saving cruise control with a high weighting on the consumption (0.66 MJ). At the same time, the trip time increases by a third using the energy-efficient cruise control (210 s for "driver

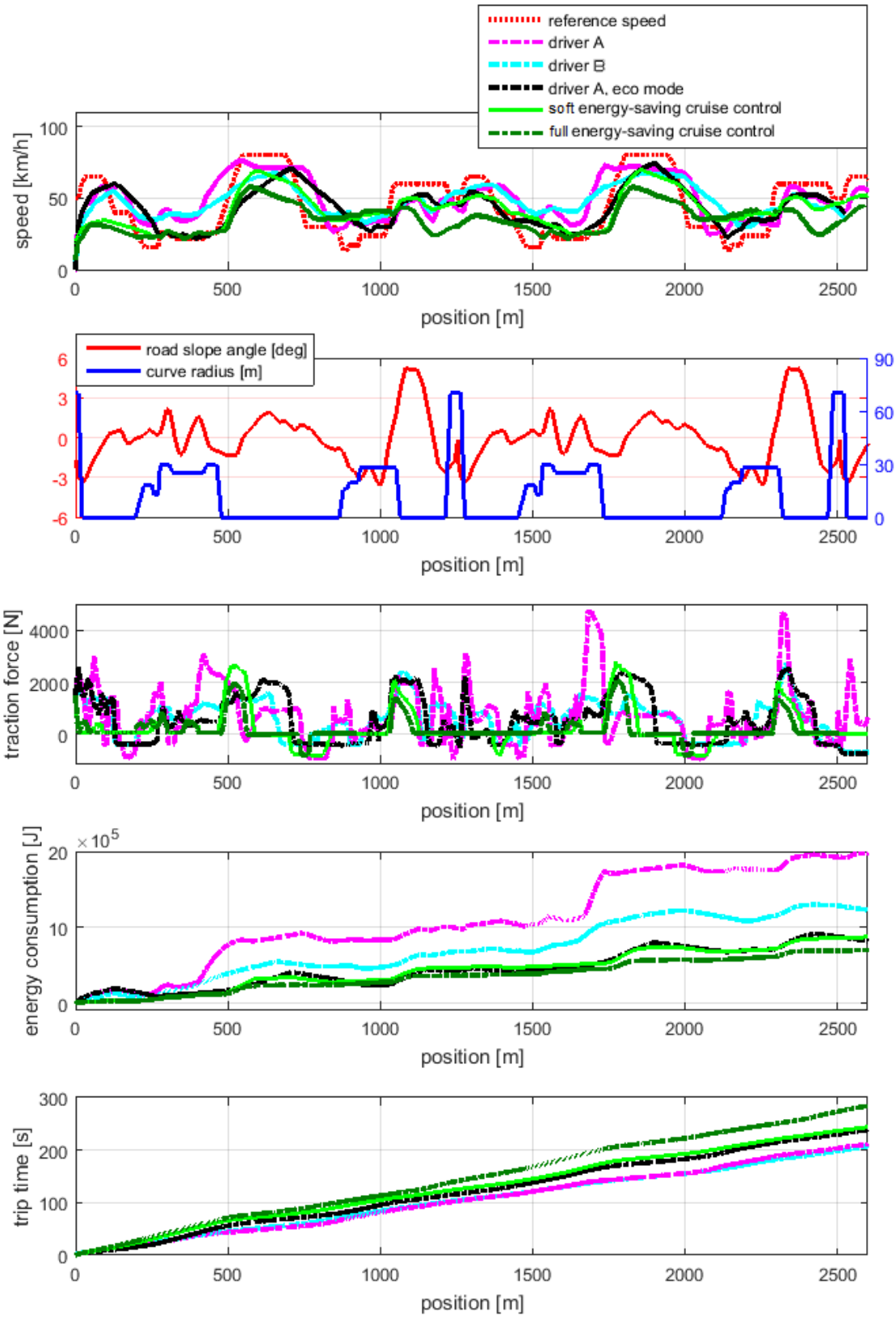


FIGURE 6.8: Experimental measurements of driving data on the CFC test track (2 laps in sequence) at Colmar-Berg of the *Smart ED* using eco-cruise control and human driving.

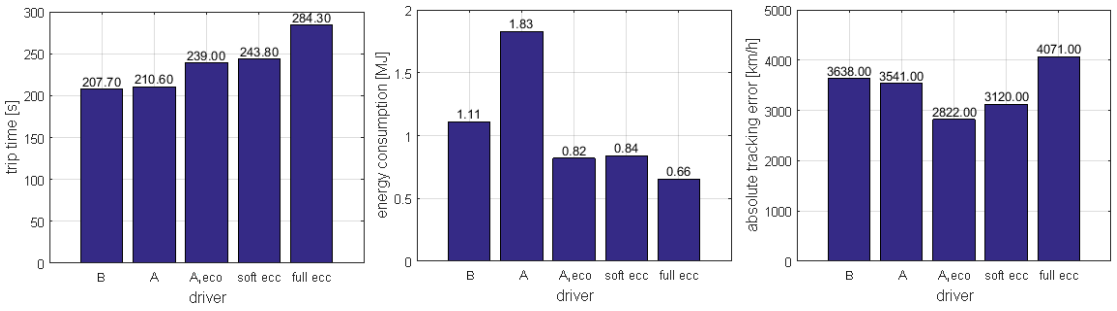


FIGURE 6.9: Comparison of trip time, energy consumption and average tracking error for different human drivers and the energy-saving cruise control (ecc) running two laps on the CFC test track at Colmar-Berg, Luxembourg.

A", 284 s for the energy-saving cruise control with full weight on the consumption in comparison). Compared to the savings of energy, this is a rather small increase. The largest waste of energy by "driver A" is caused by an earlier and faster acceleration before reaching the reference speed of 80 km/h at the positions of 550 m and 1800 m which can be seen from the energy consumption trajectories as an almost stepwise increase.

It should be pointed out that "driver A", when trying to save energy, achieves a slightly lower consumption than the "soft energy-saving cruise control" and at the same time a slightly lower trip time (239 s instead of 244 s for two laps), see also the statistics in Fig. 6.9. This is somehow surprising since the human driver cannot keep the traction force constantly at the level where freewheeling is possible which requires slightly pushing the accelerator pedal to a certain extent. Instead, the human driver either accelerates (positive traction force) or decelerates with a released pedal (negative traction force, energy recovery). Both are connected with electrical losses and thus can be expected to be less efficient than freewheeling. However, the synchronous electric machine and the power electronics work with a very high efficiency such that the associated efficiency losses do not lead to a higher energy consumption in the end. However, an unaware driver who is not focusing on driving efficiently would not achieve this result as it can be seen from the other test runs. A constant focus on the energy efficiency cannot be expected from a "normal" driver and would furthermore be very tedious in everyday traffic. Finally, the lowest energy consumption is achieved by the energy-saving cruise control with the full weight on the energy consumption.

The differences between unaware driving by the human driver and the energy-saving cruise control are astonishing. An immense decrease of 64 % can be measured between the driving style of "driver A" and the energy-saving cruise control with full weight on

the consumption. This of course goes along with an increase of the trip time by 35 % but it demonstrates the impressive fuel saving potential in the driving style. The "soft energy-saving cruise control" with a lower weight on the consumption represents a good compromise between energy savings and trip time. It still provides savings of 54 % compared to "driver A" while increasing the trip time by only 16 %.

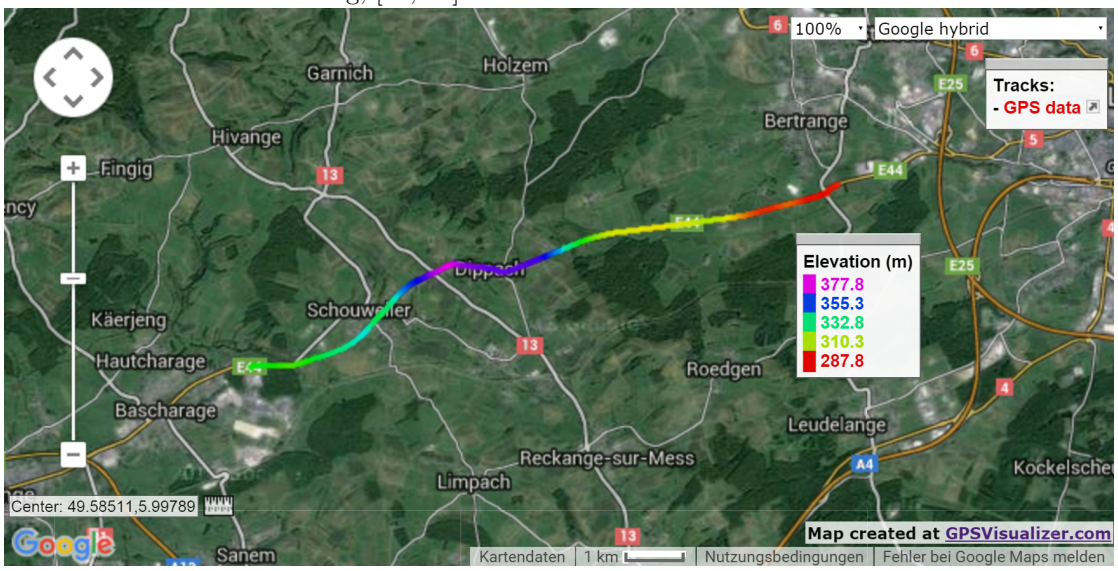
6.7 Simulation and Measurements in Realistic Road and Traffic Scenarios

The simulations presented in Chapter 5 have presented hypothetical example cases to investigate the control performance and energy-saving mechanisms in different standard situations. The preceding measurements in Section 6.6 could only be performed on a closed track for legal reasons. The simulations and measurements have provided deep insights into the behaviour and the potential of an energy-saving cruise control but do not reflect real traffic situations and the presence of other cars.

Since experimental testing of the control system on public roads is not possible, a realistic simulation scenario shall investigate the behaviour of the control loop in real driving situations in terms of driving safety, comfort and energy savings. Furthermore, a comparison to a human driving style shall be made. For this purpose, the car is driven by a human driver on public roads in Luxembourg containing urban and rural sections. The human driver tries to respect the speed limits strictly. While driving, the GPS coordinates and the altitude, as well as the driving speed, the accelerator and brake pedal positions and the battery power consumption are recorded.

For the simulation of the very same scenario in a realistic way, the road slope profile, a reference speed and the distance to eventually appearing preceding cars are required. The road slope profile can be extracted from the altitude measurements of the GPS sensor. The recorded speed profile can be used as a lifelike preceding car speed in the simulation since it is known that this speed profile is drivable in the real traffic under realistic conditions and it can be assumed that a second car using the automatic energy-saving speed control is following behind this car (cf. Fig. 6.11). Then, a suitable speed reference can be generated from the measured driving speed by rounding to the common speed limits in Luxembourg with the following logic:

FIGURE 6.10: The route from Bascharage, Luxembourg, towards Luxembourg city on urban and rural roads is used to compare simulations in a realistic environment to human driving, [16, 19]. The colours indicate the altitude level.



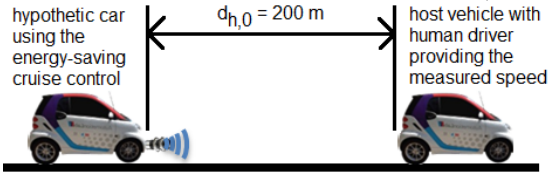
$$v_{ref}(s) = \begin{cases} 2 & \text{if } v_h(s) < 15 \\ 30 & \text{if } 15 \leq v_h(s) < 30 \\ 50 & \text{if } 35 \leq v_h(s) < 55 \\ 70 & \text{if } 55 \leq v_h(s) < 75 \\ 90 & \text{if } 75 \leq v_h(s) < 95 \\ 110 & \text{if } 95 \leq v_h(s). \end{cases} \tag{6.3}$$

This leads to a speed reference which fits to the measured host vehicle speed and considers all relevant influence factors during driving (speed limits, curves, traffic situation) since it is based on the measured trip data.

To calculate the distance between the measured GPS waypoints as well as the road slope profile and to visualise the data, the online tool *GPS Visualizer* [19] is used, here. The trip starts at the village Bascharage, Luxembourg, going towards to city of Luxembourg, see Fig. 6.10.

The simulation results with the eco-cruise controller using the cost function for guaranteed closed-loop stability are presented in Fig. 6.12. The soft eco-driving with the weighting factor $l_{e,s} = 2.5e5$ has been applied. The simulation starts with a headway

FIGURE 6.11: Initial situation for the comparison of measurement and simulation. The measured driving speed of the host vehicle with a human driver is considered as the speed of a preceding car. The vehicle using the energy-saving cruise control is assumed to follow this car with an initial distance of $d_{h,0} = 200\ m$.

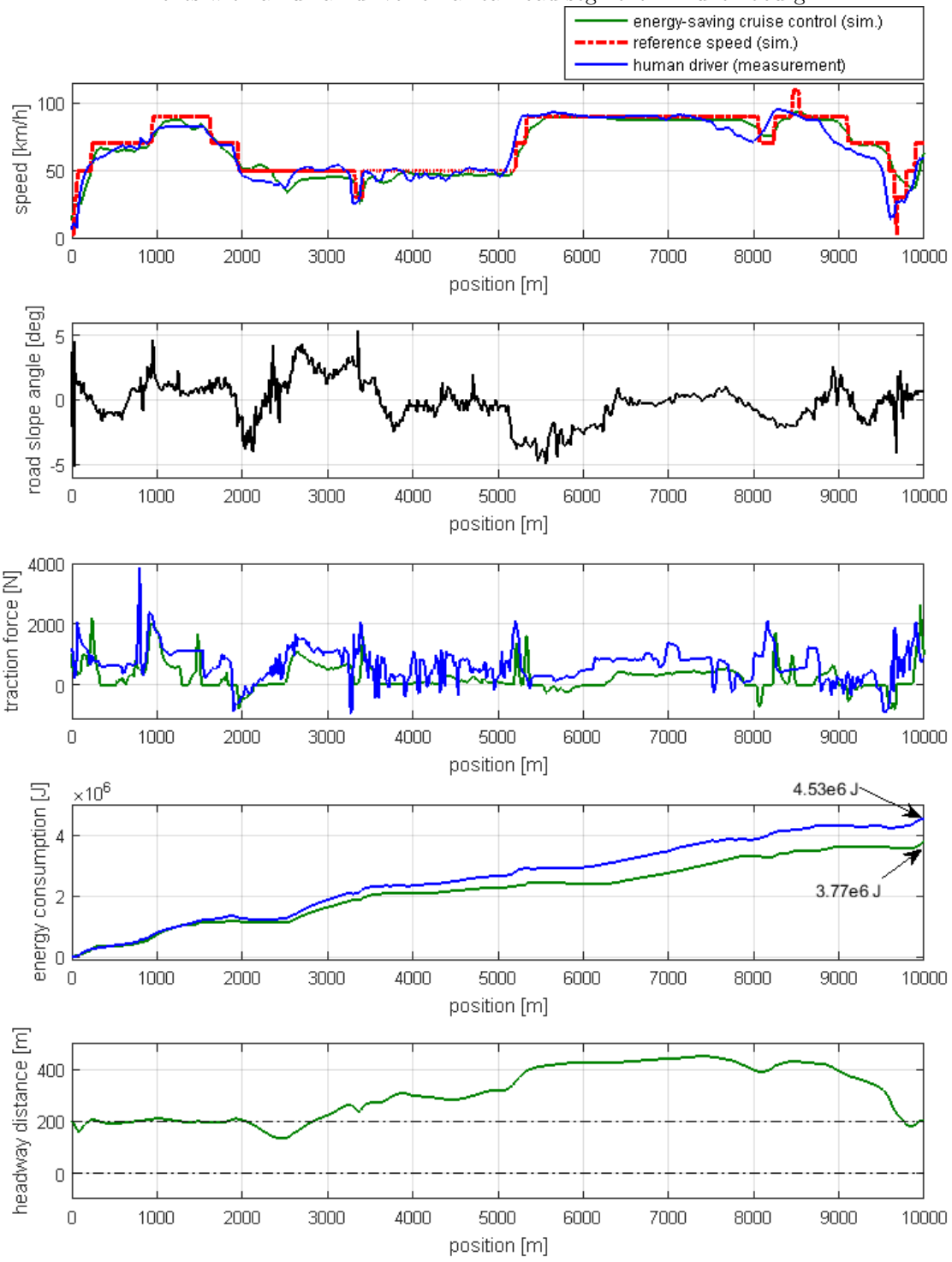


distance of $d_{h,0} = 200\ m$ between the host vehicle (using the energy-saving cruise control) and the preceding car (driving at the measured speed). Whenever the headway distance increases to more than 200 m, the energy-saving cruise control is not influenced by the preceding car. Whenever the cars are closer to each other, the speed reference generation of the eco-cruise controller follows the logic presented in Section 4.6.1.

From the speed plot at the top of the figure, it can be seen that the energy-saving cruise control follows approximately the speed of the preceding car, however running more smoothly trying to stay longer at constant speed decreasing the energy consumption and increasing the driving comfort. This is also confirmed by the traction force trajectory realised by the energy-saving cruise control which is much smoother than the measurement and displays a lot of freewheeling sections close to zero traction force. In the major part of the trip, the driving speed of the eco-cruise control stays below the reference which guarantees safe driving. Throughout the whole trip, the headway distance stays above the safety distance and the two cars arrive with a distance of 200 m and almost the same driving speed (and thus the same kinetic energy reserve) at the end of the track. Since the cars have also started with a distance of 200 m, this means that the vehicle using the energy-saving cruise controller accomplishes the trip with the same trip time at the end of the simulation and the energy consumption can directly be compared in this example. The final value of the consumption decreases from 4.53e6 J by 16.8 % to 3.77e6 J when using the energy-saving cruise control which is an immense energy saving.

In summary, this 10 km simulation scenario demonstrates the energy economy and driving comfort of the developed speed control in a realistic road and traffic situation.

FIGURE 6.12: Comparison of a simulation of the car in closed-loop control to measurements with a human driver on a real road segment in Luxembourg.



6.8 Conclusion of the Chapter

The aim of this Chapter is to validate the simulation results by measurements and to validate the control performance by comparing the results to the driving style of human drivers.

To make this possible, an experimental setup using a *Linux*-based PC platform running the *Robot Operating System (ROS)* has been created to measure the vehicle state and its position, determine the control action and sending an accelerator position commands to the car.

For legal reasons, the energy-saving cruise control is tested on a closed test track and not on public roads. The comparison of simulations and experiments with the energy-saving cruise control shows a very good accordance and validates the dynamic motion and energy consumption model.

Furthermore, the comparison of the energy-saving cruise control to human driving on the test track shows that the cruise controller applies a smoother (hence more comfortable) traction force trajectory than the human driver and a very low energy consumption. Even though it is possible for human drivers to achieve the same energy economy on the test track, this requires a constant awareness on the consumption which can make the driving tedious and unpleasant. Consequently, an energy-efficient automatic cruise control is a very useful tool.

Finally, measurements of a human driving style on a 10 km segment on public roads are carried out and the same driving scenario is repeated in a realistic simulation with the eco-cruise control. The measured driving speed of the human driver is used as a realistic speed of a preceding vehicle for the simulation (i. e. the simulated car is supposed to follow a car driving at the measured driving speed) to reproduce the traffic situation exactly. This realistic simulation demonstrates an energy-saving potential of approximately 17 % in relation to the human driver.

Chapter 7

A Flexible Move Blocking Strategy to Speed Up MPC

With the controller formulation presented in Chapter 4, the real-time implementation of the eco-cruise control loop is already possible. However, it is desirable to reduce the computational burden further to reduce the cost of implementation and to save a part of the electrical energy needed for the computation.

In Section 2.2, different strategies to speed up the calculation of the MPC control input have been presented. In this work, the MPC controller has been formulated with a linear prediction model with a straightforward zero-order hold discretisation. This classical MPC formulation directly suggests the classical input parameterisations of choosing a control horizon shorter than the prediction horizon (cf. Section 2.2.1) or input move blocking (cf. Section 2.2.2).

Shortening the control horizon however, implies the idea of taking a sequence of control actions at the beginning of the prediction horizon to regulate the system states to the set-point and keep the controls constant after this is achieved. For the eco-cruise control with a reference and a known disturbance varying over the prediction horizon, this is not a preferable option. In this case, the control cannot react to these reference and disturbance changes after the control horizon which leads to a deterioration of the control performance.

Consequently, the classical input move blocking seems more promising since it allows reactions of the controller throughout the whole prediction horizon. However, the reduced-order controls lead to a lower reference tracking performance as well. To overcome these shortcomings, a strategy to adjust the input blocks flexibly according to the reference and disturbance trajectory is investigated in this chapter.

Offset move blocking as described in Section 2.2.3 is not applicable here, since the energy-saving cruise controller cannot be formulated as a linear quadratic control law due to the dependency on the inequality constraints.

7.1 The Influence of Input Move Blocking on the Reference Tracking Performance

As a first step, the influence of the classical input move blocking approach according to Section 2.2.2 is investigated. For this purpose, the MPC controller presented in Chapter 4 is simplified to a purely-reference tracking MPC with simple box constraint limits on the control input traction force and without consideration of the vehicle energy consumption as specified by the following formulation:

$$\min_{\tilde{F}_{trac}(k+i|k)} \quad J(\hat{F}_{trac}) = \sum_{i=0}^{N_p-1} q_{s,m} \cdot \left(\hat{e}_{kin}(k+i|k) - \hat{e}_{kin,ref}(k+i|k)\right)^2 \tag{7.1a}$$

subject to the model of the system dynamics:

$$\begin{aligned}\hat{e}_{kin}(k+i+1|k) =& a_{11} \cdot \hat{e}_{kin}(k+i|k) + b_{11} \cdot \hat{F}_{trac}(k+i|k) \\ &+ e_{11} \cdot \sin(\hat{\alpha}_{sl}(k+i|k)) + e_{12} \cdot \cos(\hat{\alpha}_{sl}(k+i|k)) \quad \forall i \in \{0,1,...,N_p-1\}\end{aligned} \tag{7.1b}$$

subject to the initial conditions:

$$\hat{e}_{kin}(k|k) = e_{kin}(0) \tag{7.1c}$$

subject to the limitations of the traction force:

$$\begin{aligned} 0 \leq \hat{e}_{kin}(k+i|k) \quad & \forall i \in \{0,1,...,N_p-1\} \\ -F_{trac,lim} \leq \hat{F}_{trac}(k+i|k) \leq F_{trac,lim} \quad & \forall i \in \{0,1,...,N_p-1\} \end{aligned} \tag{7.1d}$$

subject to the input move blocking:

$$\hat{F}_{trac} = T \cdot \tilde{F}_{trac} \tag{7.1e}$$

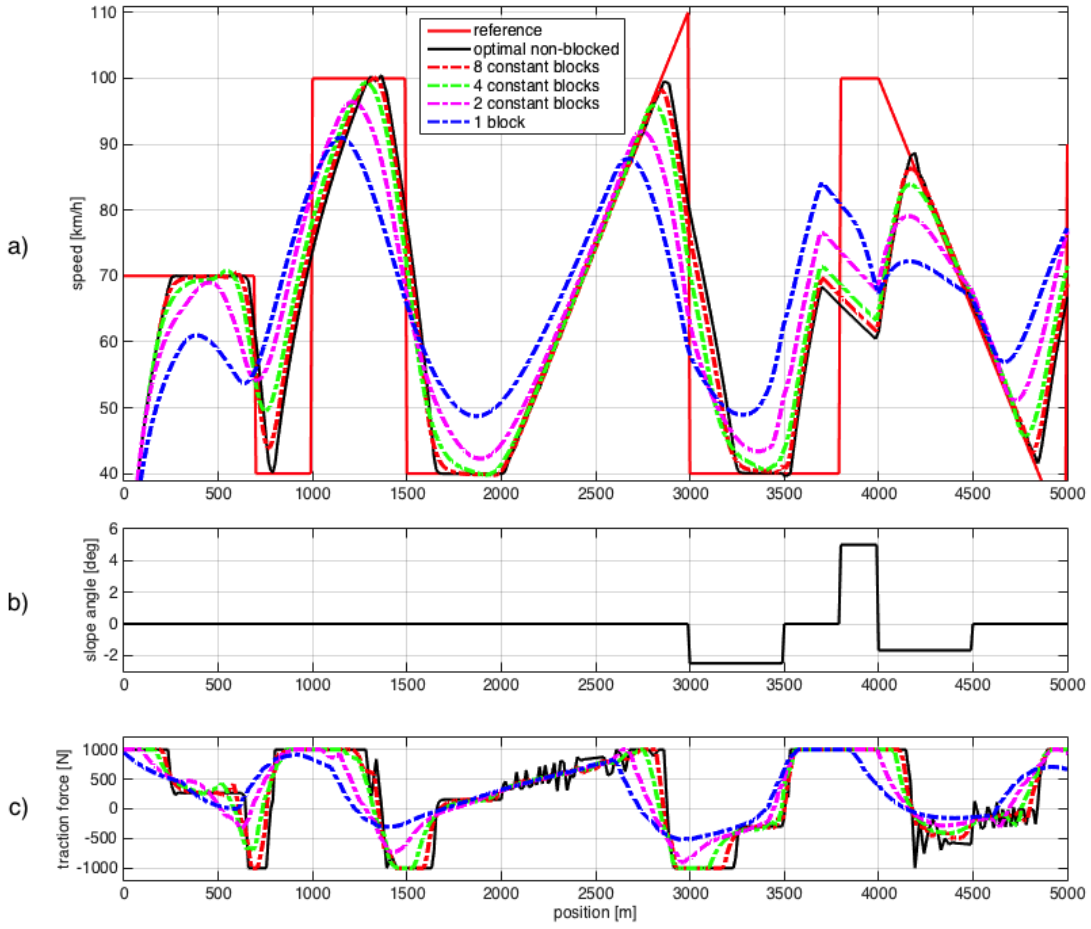
This simplified problem serves the purpose of a better visualisation of the control performance which now can simply be measured by the average absolute tracking error. The parameters remain the same as given in Tab. 4.1.

Using an MPC controller based on the optimisation problem (7.1) with a prediction horizon of $N_p = 40$ steps and a sample step of 10 m, a reference tracking scenario with standard input blocking schemes with different degrees of freedom is simulated. The blocked sections of the control input are evenly distributed over the prediction horizon (constant block length). The results are summarised in Fig. 7.1 for one, two, four , eight and forty (non-blocked) degrees of freedom. The state kinetic energy $\hat{e}_{kin}$ is converted to the driving speed v in km/h and plotted in 7.1a together with the reference trajectory. In the final part of the simulation starting at 3000 m, a known disturbance (the road slope angle $\hat{\alpha}_{sl}$) is introduced as given in Fig 7.1b. The control input traction force $\hat{F}_{trac}$ is plotted in Fig. 7.1c.

On the one hand, the figure shows that the tracking performance increases with the degrees of freedom. On the other hand, the computation time increases as well with the degree of freedom which creates a trade-off between computational effort and control performance.

However, the control performance does not only depend on the degrees of freedom of the input blocking scheme but also on the block distribution, i.e. which parts of the prediction horizon are assigned to a constant control input and where the control is allowed to vary. If the control inputs are merged to a block in regions where the original non-blocked input would be constant anyway, this does not lead to suboptimality and does not decrease the tracking performance. However, it does decrease the computational time to find the optimal solution.

FIGURE 7.1: Simulation results of reference tracking with an MPC control with different degrees of freedom and classical input move blocking.

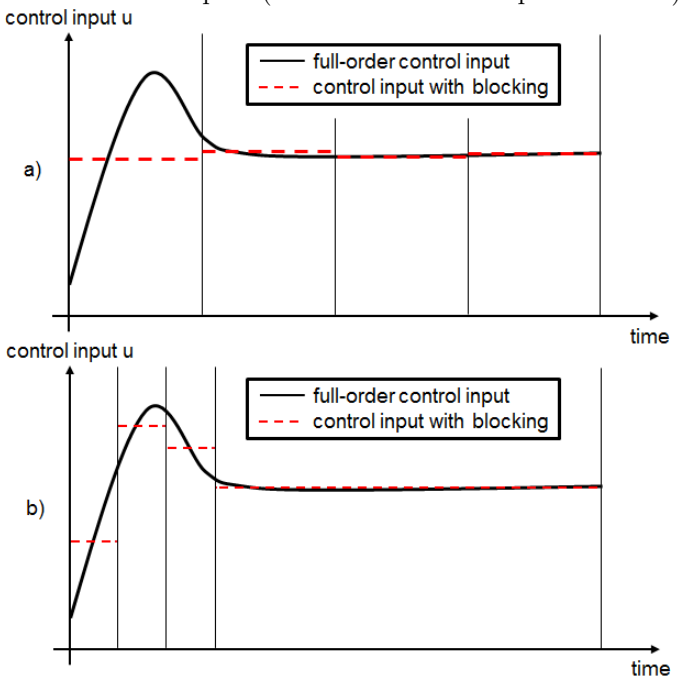


The trade-off between tracking performance and computational effort could thus be improved by adjusting the input blocking scheme and allowing variations in the control where it is necessary and enforcing a constant control input where it does not influence the performance to a large extent. Fig. 7.2 shows exemplary how the blocking scheme determines the values of the control input samples that finally determine the tracking performance of the MPC.

While the computational effort in both exemplary cases presented in Fig. 7.2 can be expected to be similar, the values of the first control input at the beginning of the horizon differ considerable which leads to a different behaviour of the control.

This shows that it is worthwhile to find a suitable blocking scheme for the given control task. This however can be complicated since the full-order control input is not known, expensive to compute and depends on the initial state of the system, the reference and the known disturbances.

FIGURE 7.2: Exemplary control input trajectory of a full degree of freedom MPC and a four-degree-of-freedom MPC. a) and b) show different block distributions leading to different control inputs (and different control performances).



7.2 A New Flexible Input Move Blocking Scheme Retaining a High Tracking Performance

In the existing literature, an approach to determine a suitable input parameterisation for reference tracking is proposed in [120] by including the weighted control input trajectory from the previous MPC step into the optimisation problem of the current step. Furthermore, the issue of maintaining feasibility guarantees in the case of input move blocking has been addressed by a relaxation of the inequality constraints on the problem in [119]. Feasibility as well as stability has been addressed by the so-called *moving-window blocking* proposed in [115].

A flexible move blocking scheme allowing the use of parallel computing has been proposed in [147]. A recent flexible blocking algorithm for the case of a constant reference point is presented in [148]. [149] shows how a blocking scheme retaining the region of attraction of an MPC controller and exploiting the available hardware capabilities can be found.

None of these strategies however allows to deal with future information about upcoming variations of the reference and the known disturbance trajectory. To resolve this issue, a new flexible input move blocking strategy is developed in the at-hand work to retain

a high reference tracking performance in the case of input move blocking by reacting online during the control operation to the current initial state as well as the reference and disturbance trajectory.

To achieve this, an estimate of the unblocked full-degree of freedom control input is taken first and then, the block distribution is adapted to this estimate. Consequently, these two main challenges need to be addressed:

- Estimating the full-order MPC control input.
- Deriving a block distribution according to this estimate.

These are investigated in the sequel for the example of problem (7.1).

7.2.1 Estimating the Full-Degree-of-Freedom MPC Control Input Trajectory

As a matter of fact, the full-order control input can be determined by solving the associated full-order optimisation problem. However, this is not an option here due to the associated computational complexity.

To obtain an estimate of the full-order solution, the following approach is taken. First, the (already) blocked MPC input of the previous MPC step is taken as an estimate of the current (blocked) MPC input. Since the moving horizon only propagates by one sample step, the optimisation problem of the previous MPC step is assumed to be similar to the current problem. This is also a common assumption in warm-starting the optimisation algorithm within an MPC framework.

By substituting the state propagation based on the prediction model (7.1b) into the cost function (7.1a), the cost can be expressed as a function of the control input sequence $\hat{F}_{trac}(k+i|k), i = 0...N_p - 1$. The blocked input sequence is not optimal with respect to the non-blocked full-order cost function of problem (7.1) in the case when the input blocking constraints (7.1e) are disregarded.

By estimating how the unblocked optimal control sequence looks like, the next block distribution can be adapted to reach the unblocked result more closely. An estimate of

the optimal unblocked input is taken by computing how sensitive the cost function is to changes in the samples of the blocked input sequence of the previous step $\hat{F}^*_{trac,k-1} = T_{k-1} \cdot \tilde{F}^*_{trac,k-1}$. If there are no inequality constraints active and the control input is optimum, the sensitivities will be zero. However, since the blocked control input will not be optimum when the blocking constraints are disregarded, the cost function will be sensitive to changes in the blocked control samples.

To identify to what extent a move of one control input sample $\hat{F}_{trac}(k+i|k)$ leads to a decrease in the full-order cost function value, the sensitivities s_J of the updated cost function (7.1a) with respect to each sample of the control input of the previous step are computed. For this purpose, the optimum blocked control input sequence of the previous step $\hat{F}^*_{trac,k-1} = T_{k-1} \cdot \tilde{F}^*_{trac,k-1}$ is inserted into the cost function as well as the updated initial state ($\hat{e}_{kin}(k|k)$), the updated reference ($\hat{e}_{kin,ref}(k+i|k|k), i = 0...N_p-1$) and disturbance ($\hat{\alpha}_{sl}(k+i|k|k), i = 0...N_p-1$) sequence. In this operating point, the change of the cost function value is computed subject to a small variation in each single control input sample $\hat{F}^*_{trac}(k-1+i|k-1)$ (i.e. the partial derivative). For simplicity, inequality constraints are disregarded at this first step.

$$s_J(k+i|k) = \Big| \frac{\partial J\big(\hat{F}_{trac}\big)}{\partial \hat{F}_{trac}(k-1+i|k-1)} \Big|_{\hat{F}^*_{trac,k-1}, \hat{e}_{kin}(k|k), \hat{e}_{kin,ref,k}, \hat{\alpha}_{sl,k}} \tag{7.2}$$

The cost function comprises N_p sample steps and so does the blocked control input sequence $\hat{F}^*_{trac,k-1}$. Even though certain samples of the blocked control input sequence have the same value according to the blocking constraints, the sensitivities of the cost function vary at each sample because the predicted state trajectory is not blocked. Since the cost function $J(\hat{F}_{trac})$ results in scalar quadratic and linear terms, the partial derivatives can be derived analytically to compute s_J. Alternatively, the sensitivities can be approximated numerically by finite differences.

Having the N_p sensitivities of the full-order cost function (7.1a) related to changes in each single component of the input trajectory, they are multiplied by a gain factor a_s and then added to the blocked input. The result can be seen as a step towards the optimum non-blocked full-order control input. (The sensitivities indicate how much the cost function value changes with a change of the value of the related control sample.

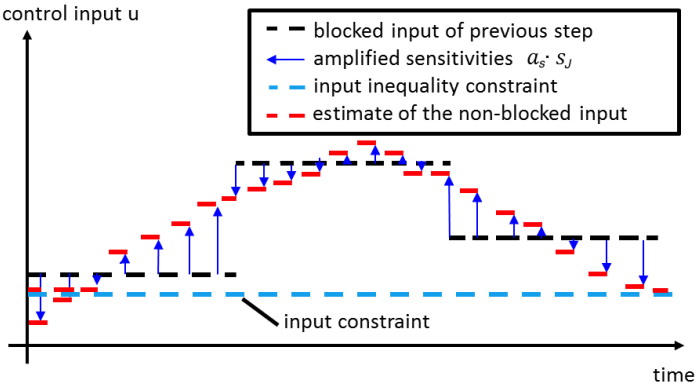


FIGURE 7.3: Exemplary blocked control input $U = T\tilde{U}$ of the previous sample step (black dashed line) with the amplified sensitivities $a_s \cdot s_J(k+i|k)$ of the full-order cost function related to changes of single input samples $u(k+i|k)$ illustrated as blue arrows. The resulting estimate of the non-blocked input is plotted as red dashed line.

This suggests that a larger step must be taken towards the optimum where the influence on the cost function is high, analog to gradient-descent methods.)

The amplified sensitivities are illustrated in Fig. 7.3 as blue arrows.

$$\hat{F}'_{trac,est}(k+i|k) = \tilde{F}^*_{trac}(k+i|k-1) + a_s \cdot s_J(k+i|k) \quad \forall i \in \{0,1,...,N_p-1\} \tag{7.3}$$

In a second step, the trajectory $\hat{F}'_{trac,est}$ is limited / saturated by the inequality constraints on the control $-F_{trac,lim} \leq \hat{F}_{trac}(k+i|k) \leq F_{trac,lim}$, i.e.

$$\hat{F}_{trac,est}(k+i|k) = \begin{cases} -F_{trac,lim} & \text{if } \hat{F}'_{trac,est}(k+i|k) < -F_{trac,lim} \\ \hat{F}'_{trac,est}(k+i|k) & \text{if } -F_{trac,lim} \leq \hat{F}'_{trac,est}(k+i|k) \leq F_{trac,lim} \\ F_{trac,lim} & \text{if } \hat{F}'_{trac,est}(k+i|k) > F_{trac,lim} \end{cases} \tag{7.4}$$

This is necessary because even if the full-order cost function is sensitive to changes at certain samples, there is no use in facilitating these changes by unblocking the input if the inequality constraints do not allow this change. The result of this saturation is then finally taken as an estimate for the non-blocked full-order control input.

7.2.2 Adapting the Block Distribution to the Estimated Input

With the estimate of the full-order control input trajectory $\hat{F}_{trac,est}$ obtained in the previous section, the input block distribution must be found such that the input blocks are short where the variations in $\hat{F}_{trac,est}$ are high in order to give the freedom to vary the control input especially in these samples. The fitting error of a piecewise constant approximation $\hat{F}_{trac,est,approx}$ is taken as a measure for the variation.

Finding the best piecewise constant approximation $\hat{F}_{trac,est,approx}$ of the estimate of the non-blocked control input $\hat{F}_{trac,est}$ with a limited number M of blocks is in general a hard combinatorial optimisation problem [150]. Since a solution with low computational effort is required here, a heuristic algorithm is applied to find a suboptimal piecewise-constant approximation, instead.

However, it should be guaranteed that the obtained solution gives a better or at least equally good approximation of the estimated non-blocked input than the evenly distributed blocking scheme with equal block lengths. To achieve this, an algorithm is used that varies the block distribution with constant block lengths to gain a lower overall fitting error as follows.

Evenly distributed input blocks are taken as the starting point. Then, all possible piecewise constant fits that can be achieved by shifting only *one* block margin left or right by only *one* sample step are computed (i.e. one input block is enlargened and another one is shortened by one sample per iteration). The specific one-sample move that leads to the piecewise constant least-squares fit with the lowest fitting error is then stored and the procedure is repeated starting with this new distribution. By this, the algorithm will consecutively shift that specific block margin where the highest gain in the overall fitting quality can be achieved until no further improvement is possible by shifting only one single margin. This will lead to a locally optimum block distribution. The described procedure is listed in the following and illustrated in Fig. 7.4.

1. Start with an initial block distribution with **M** degrees of freedom/blocks as the "current working distribution" and compute the related squared piecewise constant least-squares fitting error of the estimated full degree of freedom input: $||\hat{F}_{trac,est} - \hat{F}_{trac,est,approx}||_2^2$

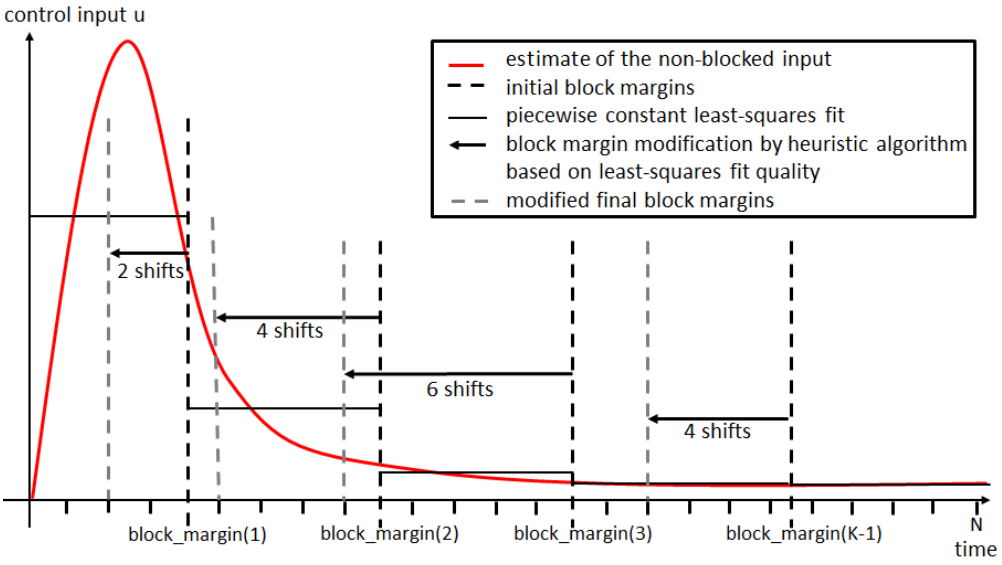


FIGURE 7.4: Working principle of the block size adjustment based on shifting the block margins according to the fitting error of a piecewise constant approximation.

2. **For i=1, i++, i=M-1**: If possible without violating the margins of neighbouring blocks, shift the margin of block **i** to the left by **one** sample and store the resulting block distribution.

3. **For i=1, i++, i=M-1**: If possible without violating the margins of neighbouring blocks, shift the margin of block **i** to the right by **one** sample and store the resulting block distribution.

4. For each of these new block distributions created in steps (2) and (3), compute the related piecewise constant squared least-squares fitting error of the estimated full degree of freedom input: $||\hat{F}_{trac,est} - \hat{F}_{trac,est,approx}||_2^2$

5. Choose the new block distribution with the lowest resulting fitting error.

6. **If** this new fitting error is lower than the lowest of the previous iteration, assign the block distribution with the lowest fitting error as the "current working distribution" and go to **2)**.

7. **Else:** Return and output the "current working distribution" to be used as blocking scheme in the reduced-order MPC.

Due to the computational simplicity of this procedure, it can be run online several times at each MPC step with different initial block distributions at each MPC step to achieve a solution closer to the global optimum.

7.2.3 Numerical Results with the New Flexible Input-Move Blocking Scheme

Both, the estimate of the full-order control input obtained by the sensitity anaysis (Section 7.2.1) as well as the adaptation of the block length according to this estimate (Section 7.2.2) have been added to the example problem (7.1) to show the effectiveness of the approach. The classical approach of a constant blocking scheme has been compared to the new flexible approach which adapts the block sizes online during control operation in terms of control performance and computation time. This comparison has been performed for different degrees of freedom of the input trajectory.

The results of the simulation are displayed in Fig. 7.5. The prediction horizon consists of 40 sample steps of 10 m. For the sake of comparison, the first simulation is performed without move blocking ($T = I_{N\times N}$), i.e. the control may vary at each of the 40 sample steps (full degree of freedom). Then, the freedom of the control is reduced from forty to only four or two input blocks of equal lengths. These results are then compared to the proposed flexible blocking scheme with four and two degrees of freedom and presented in Fig. 7.5.

The fact that none of the controllers can follow the reference more closely is due to the input limitations. As expected, Fig. 7.5 shows that the controller with the full-order freedom in the control tracks the reference best with a mean absolute tracking error of 9.34 km/h. In comparison, the controller with four input blocks of equal length shows a considerable performance degradation and requires longer distances to finally reach the reference trajectory. The average absolute tracking error here amounts to 10 km/h (+ 7 %). However, the accumulated computational time to compute the control responses throughout the simulation with the *quadprog* function in *MATLAB* can be reduced from 60 s to 30 s (-50 %). The flexible move blocking with four degrees of freedom achieves the same average absolute tracking error as the full degree of freedom solution at an accumulated computational time of 33 s (-45 % compared to the full-order solution). This demonstrates the potential of flexible blocking schemes.

The controller with two flexible degrees of freedom even achieves a saving 43 % of computational time at a minor performance degradation of +4.8 % mean absolute tracking error compared to the full-degree of freedom controller.

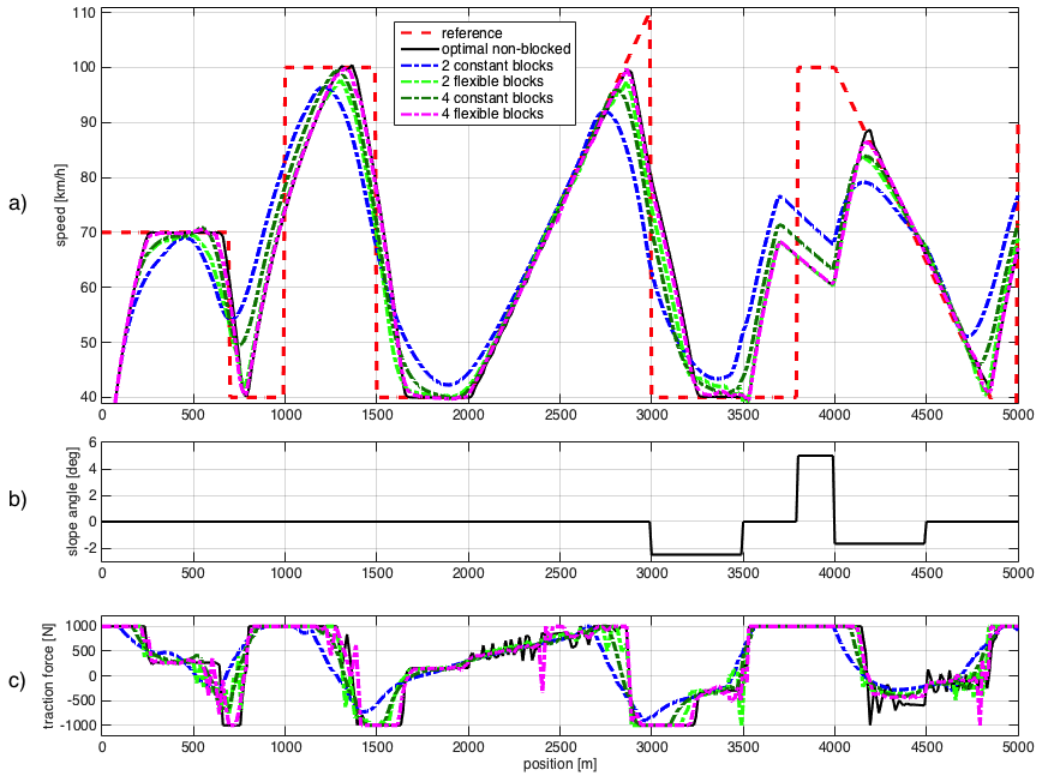


FIGURE 7.5: Simulation results of a reference tracking scenario comparing the classical input-move blocking with the new flexible strategy.

The simulation of the same scenario has then been repeated with different degrees of freedom, both with the classical constant as well as the proposed flexible blocking scheme. The accumulated computation time and the mean absolute tracking error of all simulation runs is compared in Fig. 7.6.

Fig. 7.6 shows that the flexible move blocking decreases the mean absolute tracking error in all cases. However, the computation time to determine a suitable input block distribution increases with the degree of freedom. While the computation of the sensitivities of the full-order cost function is independent of the number of blocks (cf. Section 7.2.1), the determination of the block lengths requires more time and iterations when the number of blocks increases (cf. Section 7.2.2). Having this in mind, the proposed algorithm is reasonable when using a low degree of freedom in the control with a flexible adjustment to the conditions to achieve a solution comparable to the full degree of freedom at a lower computation time. In the above presented numerical example, it is reasonable to replace the full degree of freedom solution with two, four or five flexible input block sections in order to achieve a better control performance and at the same time a lower computation time than the solutions with constant input-move blocking

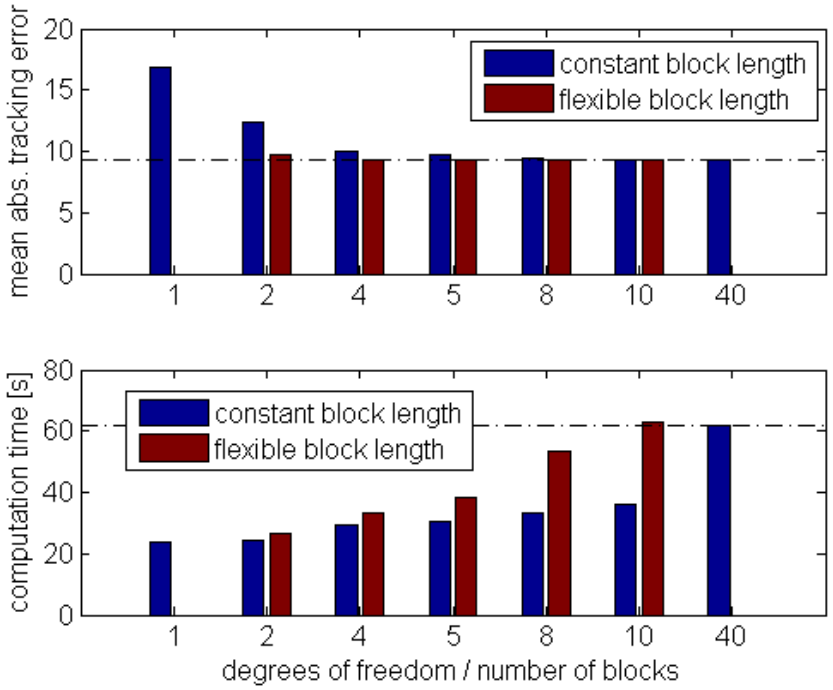


FIGURE 7.6: Mean absolute tracking error and accumulated computation time of the MPC simulation with the constant (standard) and flexible (proposed) blocking scheme and different degrees of freedom.

schemes. Higher degrees of freedom do not lead to a lower tracking error but increase the computational time required for the flexible parameterisation considerably.

7.3 Limitations of the Flexible Input-Move Blocking Scheme and Conclusion of the Chapter

To apply the proposed method, the gain a_s in (7.3) must be tuned properly to achieve good results. By now, this has been done based on simulations. Here, further research should be done to achieve a more sophisticated approach. Further, oscillations in the control input may occur since the input blocking scheme can vary from MPC step to MPC step which leads to differing optimisation results and control inputs. This also prohibits stability guarantees for the control. For example, if the blocking scheme oscillates between two different block distributions, the computed control input will also vary and this might cause an oscillating system state. The block distribution algorithm must consequently be included in a stability proof of the control which will be very hard.

However, the gains in computational time and reference tracking quality are very promising and outperform existing methods. As the numerical example in this paper shows, a four degree of freedom approach can achieve almost the same performance as the full-order 40 degree of freedom control at only a half of the computation time. This motivates further research to improve the proposed algorithms to make them faster, more accurate and to establish stability guarantees.

Finally, an extension of the results to non-linear systems would be very promising. Here, the computation time is an even more critical issue while the computation of the flexible input move blocking would not require more additional effort than in the case of linear systems.

An application of the flexible blocking scheme to the energy-efficient cruise control can increase the sample rate and reduce the cost of implementation at the same time. However, further development is necessary to enable a practical implementation of the approach.

Chapter 8

Conclusion and Outlook

The thesis report presents the design and development of an advanced driver-assistant system for the energy-saving cruise control especially of electric vehicles. The main advantage of the presented approach over existing energy-saving cruise control approaches is the practical applicability and the ability to consider road slopes, curves, and the traffic situation in a realistic way. A major challenge is to design the optimal controller such that the control decisions match the behaviour of a real series-production vehicle on the one hand and can be computed in real-time on the other hand.

This is achieved by conducting measurements with the vehicle to identify a dynamic prediction model and by converting the originally non-linear optimal control problem to a quadratic program by making use of a change of variables and the technique of separable programming. The resulting problem formulation is very suitable for a real-time implementation since the convexity and thus the convergence of the algorithm in polynomial time can be guaranteed. Moreover, theoretical closed-loop stability guarantees can be derived.

Besides the design of the control system, the thesis report investigates the energy-saving potential of electrical vehicles in different driving situations with the help of optimal control.

The closed-loop control is evaluated in simulations as well as experimentally implemented in a *Smart ED* series-production vehicle for drive tests on a closed track. Experiments and simulations have shown a high accordance which validates the underlying prediction

model of the controller as well as the simulation model. Furthermore, a significant energy saving compared to human drivers is demonstrated both in simulations and experiments.

The precise amount of energy savings compared to a human driver is hard to assess since it highly depends on which human driving style forms the basis for the comparison. When the comparison is carried out at the basis of an equal trip time, the saving potential of the energy-saving cruise control could be demonstrated between 3 and 17 % which is a high saving compared to the possible improvements in the efficiency of the vehicle technology.

In summary, it can be said that the presented advanced driver assistance system can significantly reduce the energy consumption of an electric vehicle and is suitable for a practical implementation thanks to the special problem formulation. Consequently, such a system can contribute partly to solving the problem of the limited range of current electric vehicles.

To further decrease the computational load of the MPC implementation, a complexity reduction scheme based on input parameterisation referred to as *flexible move blocking* has been studied. In contrast to existing move blocking approaches the input parameterisation adapts to the initial state of the controlled system, the reference trajectory and a predictively known disturbance to retain the tracking performance.

For future development, different extensions of the work at hand are possible. The control formulation itself can by extended by including the headway distance to the preceding car as a controlled state to improve the headway tracking performance. This can be supported by stochastic methods to consider unpredictable events like unforeseen changes in the speed of the preceding vehicle or cut-in manoeuvres of preceding cars. Another interesting field of research in this context is the vehicle-to-vehicle (V2V) and vehicle-to-infrastructure (V2I) communication which can make more information about the traffic situation available.

Besides that, the professional implementation on a system with hard real-time guarantees can be aimed at to take the step towards the industrial implementation.

Furthermore, the flexible move blocking is a promising approach to reduce the computational load while retaining the tracking performance. To make the method ready for a

practical implementation, further development regarding the stability of the closed-loop control is required.

Bibliography

[1] TOYOTA Motor Corporation. Tips on eco driving - eco driving flowchart, 2014. URL http://www.toyota.co.za/environment/tips-on-eco-driving. [Online; December, 22nd, 2014].

[2] Jérôme Barbé, GA Boy, and Mariano Sans. Gerico: A human centered eco-driving system. In *Proc. Int. Conf. 10th IFAC/IFIP/IFORS/IEA Symposium on Analysis, Design, and Evaluation of Human-Machine Systems, Seoul, Korea*, 2007.

[3] GARMIN Ltd. Garmin ecoroute hd, 2014. URL http://www.garmin.com/de/ecoroutehd/. [Online; December, 22nd, 2014].

[4] S Sakaguchi, M Shiomi, and M Oomori. Development of eco pedal system assisting eco-driving. *JSAE proceedings (Society of Automotive Engineers of Japan)*, 48(10): 1–6, 2010.

[5] Scania AB. Scania active prediction, 2014. URL http://www.scania.com/products-services/trucks/safety-driver-support/driver-support-systems/active-prediction/. [Online; December, 22nd, 2014].

[6] Dongbin Zhao, Zhaohui Hu, Zhongpu Xia, Cesare Alippi, Yuanheng Zhu, and Ding Wang. Full-range adaptive cruise control based on supervised adaptive dynamic programming. *Neurocomputing*, 125(0):57 – 67, 2014.

[7] Dai Li, Xia Yuanqing, Fu Mengyin, and Mahmoud Magdi. Advances in discrete time systems. 2012. URL http://www.intechopen.com/books/export/citation/BibTex/advances-in-discrete-time-systems/discrete-time-model-predictive-control.

[8] The MathWorks Inc. About model-predictive control, 2014. URL http://de.mathworks.com/help/mpc/ug/optimization-problem.html. [Online; December, 22nd, 2014].

[9] Smart. Specifications, 2014. URL http://www.smart.de. [Online; December, 22nd, 2014].

[10] A Zomotor. Fahrwerktechnik: Fahrverhalten, 1991.

[11] Idaho National Laboratory. Bev battery testing results, 2014 smart electric drive, vin 7525, 2014. URL http://avt.inl.gov/pdf/fsev/batterySmart7525.pdf. [Online; December, 22nd, 2014].

[12] M.A.S. Kamal, M. Mukai, J. Murata, and T. Kawabe. Ecological vehicle control on roads with up-down slopes. *Intelligent Transportation Systems, IEEE Transactions on*, 12(3):783–794, Sept 2011.

[13] J. M. Maciejowski. *Predictive Control with Constraints*. Pearson Education, Prentice Hall, 2001.

[14] U. Brannolte and S. Holz. Simulation des verkehrsablaufs auf landstraßen - modellerweiterung. *Forschung Straßenbau und Straßenverkehrstechnik*, 402, 1983.

[15] Centre de Formation pour Conducteurs S.A. Test track of the centre de formation pour conducteurs s.a., 2014. URL http://www.cfc.lu. [Online; December, 22nd, 2014].

[16] Google Maps. Map data luxembourg, 2015. URL https://www.google.de/maps. [Online; August, 18th 2015].

[17] Geocontext.org. Geocontext profiler, 2014. URL www.geocontext.org/publ/2010/04/profiler/de/. [Online; December, 22nd, 2014].

[18] Electronic Projects Components Available Wordwide. Teensy 3.1, 2014. URL https://www.pjrc.com/teensy/. [Online; December, 22nd, 2014].

[19] Adam Schneider. Gps visualizer, 2015. URL http://www.gpsvisualizer.com/. [Online; 12. August 2015].

[20] John P. Crank and Linda S. Jacoby. 2 - what is global warming? In John P. CrankLinda S. Jacoby, editor, *Crime, Violence, and Global Warming*, pages 23 – 50. Anderson Publishing, Ltd., 2015.

[21] Mark Z. Jacobson. Review of solutions to global warming, air pollution, and energy security. *Energy Environ. Sci.*, 2:148–173, 2009.

[22] Patrícia Baptista, José Tavares, and Gonçalo Gonçalves. Energy and environmental impacts of potential application of fully or partially electric propulsion vehicles: Application to lisbon and são miguel, portugal. *Transportation Research Procedia*, 3(0):750 – 759, 2014. 17th Meeting of the {EURO} Working Group on Transportation, EWGT2014, 2-4 July 2014, Sevilla, Spain.

[23] Jurgen Buekers, Mirja Van Holderbeke, Johan Bierkens, and Luc Int Panis. Health and environmental benefits related to electric vehicle introduction in {EU} countries. *Transportation Research Part D: Transport and Environment*, 33(0):26 – 38, 2014.

[24] Melecita M. Archuleta. Toxicity of materials used in the manufacture of lithium batteries. *Journal of Power Sources*, 54(1):138 – 142, 1995. Proceedings of the Seventh International Meeting on Lithium Batteries.

[25] L. Hollmotz and M. Hackmann. Sicherheit von lithium-ionen-batterien im kraftfahrzeug. stand der technik, risiken, zielkonflikte, testverfahren und bewertungsmethoden. *VDI Berichte*, 2144:205 – 216, 2011. Entwicklungen im Karosseriebau: Tagung.

[26] C.E. Thomas. Fuel cell and battery electric vehicles compared. *International Journal of Hydrogen Energy*, 34(15):6005 – 6020, 2009.

[27] M.A. Hannan, F.A. Azidin, and A. Mohamed. Hybrid electric vehicles and their challenges: A review. *Renewable and Sustainable Energy Reviews*, 29(0):135 – 150, 2014.

[28] Catarina C. Rolim, Patricia C. Baptista, Gonçalo O. Duarte, and Tiago L. Farias. Impacts of on-board devices and training on light duty vehicle driving behavior. *Procedia - Social and Behavioral Sciences*, 111(0):711 – 720, 2014. Transportation: Can we do more with less resources? – 16th Meeting of the Euro Working Group on Transportation – Porto 2013.

[29] Jack N. Barkenbus. Eco-driving: An overlooked climate change initiative. *Energy Policy*, 38(2):762 – 769, 2010.

[30] Matthew Barth and Kanok Boriboonsomsin. Energy and emissions impacts of a freeway-based dynamic eco-driving system. *Transportation Research Part D: Transport and Environment*, 14(6):400 – 410, 2009. The interaction of environmental and traffic safety policies.

[31] Maria Staubach, Norbert Schebitz, Frank Koester, and Detlef Kuck. Evaluation of an eco-driving support system. *Transportation Research Part F: Traffic Psychology and Behaviour*, 27, Part A(0):11 – 21, 2014.

[32] Md. Saniul Alam and Aonghus McNabola. A critical review and assessment of eco-driving policy & technology: Benefits & limitations. *Transport Policy*, 35(0): 42 – 49, 2014.

[33] Austrian Energy Agency. Ecodriven campaign catalogue for european ecodriving & traffic safety campaigns, 2014. URL http://ec.europa.eu/energy/intelligent/projects/sites/iee-projects/files/projects/documents/ecodriven_catalogue_campaign_en.pdf. [Online; December, 22nd, 2014].

[34] TOGETHER Consortium. Eco-driving tips – how to drive energy efficient?, 2014. URL http://www.together-eu.org/docs/102/TOGETHER_Eco-driving_5_Handout_14.pdf. [Online; December, 22nd, 2014].

[35] LeasePlan Corporation. The 5 golden rules of eco-driving, 2014. URL https://www.leaseplan.sk/welcome-to-leaseplan-slovakia/my-car/the-5-golden-rules-of-eco-driving?page_id=6332104. [Online; December, 22nd, 2014].

[36] Irish School of Motoring. 7 golden rules of eco-driving - saving you money & the environment, 2014. URL http://www.ism.ie/blog/7-golden-rules-eco-driving. [Online; December, 22nd, 2014].

[37] Planet Forward. 10 eco-driving rules that can lower costs and carbon footprint, 2014. URL http://planetforward.ca/blog/10-eco-driving-rules-that-can-lower-costs-and-carbon-footprint/. [Online; December, 22nd, 2014].

[38] MAN Truck International. Man profidrive® - training courses from professionals for professionals, 2014. URL http://www.truck.man.eu/global/en/services-and-parts/efficient-operation/man-profidrive/services/Services.html. [Online; December, 22nd, 2014].

[39] Volvo Trucks. Driver training, 2014. URL http://www.volvotrucks.com/trucks/na/en-us/products/productivity/drivertraining/Pages/training.aspx. [Online; December, 22nd, 2014].

[40] Charnwood Trucks. Fuel economy, 2014. URL http://www.hardstaffgroup.co.uk/site/charnwood-truck-services/fuel-economy-training. [Online; December, 22nd, 2014].

[41] K. Boriboonsomsin, M.J. Barth, Weihua Zhu, and A. Vu. Eco-routing navigation system based on multisource historical and real-time traffic information. *Intelligent Transportation Systems, IEEE Transactions on*, 13(4):1694–1704, Dec 2012.

[42] Michalis Masikos, Konstantinos Demestichas, Evgenia Adamopoulou, Filippo Cappadona, and Stephane Dreher. Ecogem - cooperative advanced driver assistance system for green cars. In Gereon Meyer and Juergen Valldorf, editors, *Advanced Microsystems for Automotive Applications 2011*, VDI-Buch, pages 213–223. Springer Berlin Heidelberg, 2011.

[43] Kyoungho Ahn and Hesham A. Rakha. Network-wide impacts of eco-routing strategies: A large-scale case study. *Transportation Research Part D: Transport and Environment*, 25(0):119 – 130, 2013.

[44] University of California Sean Nealon. Cutting electric vehicle energy use 51 percent, 2014. URL http://www.cert.ucr.edu/news/2014/2014-09-08.html. [Online; December, 22nd, 2014].

[45] FIAT S.p.A. Fiat eco:drive, 2014. URL http://www.fiat.com/ecodrive/. [Online; December, 22nd, 2014].

[46] Driving Curve Inc. Driving curve, 2014. URL http://www.drivingcurve.com/. [Online; December, 22nd, 2014].

[47] IFP Energies Nouvelles. Geco drive, 2014. URL http://www.geco-drive.fr/. [Online; December, 22nd, 2014].

[48] M.C. van der Voort. Fest. a new driver support tool that reduces fuel consumption and emissions. In *Advanced Driver Assistance Systems, 2001. ADAS. International Conference on (IEE Conf. Publ. No. 483)*, pages 90–93, 2001.

[49] T. Wada, K. Yoshimura, S.-I. Doi, H. Youhata, and K. Tomiyama. Proposal of an eco-driving assist system adaptive to driver's skill. In *Intelligent Transportation Systems (ITSC), 2011 14th International IEEE Conference on*, pages 1880–1885, Oct 2011.

[50] T. Bar, R. Kohlhaas, J.M. Zollner, and K. Scholl. Anticipatory driving assistance for energy efficient driving. In *Integrated and Sustainable Transportation System (FISTS), 2011 IEEE Forum on*, pages 1–6, June 2011.

[51] E. Yay and N.M. Madrid. A new driving system towards energy-efficient and safe driving behaviour. In *Intelligent Solutions in Embedded Systems (WISES), 2012 Proceedings of the Tenth Workshop on*, pages 3–8, July 2012.

[52] V. Corcoba Magaña and M. Munoz-Organero. Artemisa: An eco-driving assistant for android os. In *Consumer Electronics - Berlin (ICCE-Berlin), 2011 IEEE International Conference on*, pages 211–215, Sept 2011.

[53] R. Araujo, A. Igreja, R. de Castro, and R.E. Araujo. Driving coach: A smartphone application to evaluate driving efficient patterns. In *Intelligent Vehicles Symposium (IV), 2012 IEEE*, pages 1005–1010, June 2012.

[54] R. Frank, G. Castignani, R. Schmitz, and T. Engel. A novel eco-driving application to reduce energy consumption of electric vehicles. In *Connected Vehicles and Expo (ICCVE), 2013 International Conference on*, pages 283–288, Dec 2013.

[55] Doo Seop Yun, Jeong-Woo Lee, Shin-Kyung Lee, and Oh-Cheon Kwon. Development of the eco-driving and safe-driving components using vehicle information. In *ICT Convergence (ICTC), 2012 International Conference on*, pages 561–562, Oct 2012.

[56] K. Satou, R. Shitamatsu, M. Sugimoto, and E. Kamata. Development of the on-board eco-driving support system. *International Scientific Journal for Alternative Energy and Ecology*, 9(89):35–40, 2010.

[57] A.B. Schwarzkopf and R.B. Leipnik. Control of highway vehicles for minimum fuel consumption over varying terrain. *Transportation Research*, 11(4):279 – 286, 1977. ISSN 0041-1647.

[58] B. Saerens, H.A. Rakha, M. Diehl, and E. Van den Bulck. A methodology for assessing eco-cruise control for passenger vehicles. *Transportation Research Part D: Transport and Environment*, 19(0):20 – 27, 2013. ISSN 1361-9209.

[59] M. A. Athans and P. L. Falb. *Optimal Control*. McGraw Hill, New York, 1966.

[60] Bart Saerens. *Optimal Control Based Eco-Driving Theoretical Approach and Practical Applications*. PhD thesis, Katholieke Universiteit Leuven, 2012.

[61] Kenneth Sebesta. *Optimal observers and optimal control : improving car efficiency with Kalman et Pontryagin*. PhD thesis, Université de Bourgogne; Université du Luxembourg, 2010.

[62] Anders Fröberg and Lars Nielsen. Optimal control utilizing analytical solutions for heavy truck cruise control, 2008.

[63] Tao Wang, Christos G Cassandras, and Sepideh Pourazarm. Optimal motion control for energy-aware electric vehicles. *Control Engineering Practice*, 38:37–45, 2015.

[64] Nicolas Petit and Antonio Sciarretta. Optimal drive of electric vehicles using an inversion-based trajectory generation approach. *IFAC World Congress*, 18(1): 14519–14526, 2011.

[65] Wissam Dib, Alexandre Chasse, Philippe Moulin, Antonio Sciarretta, and Gilles Corde. Optimal energy management for an electric vehicle in eco-driving applications. *Control Engineering Practice*, 29(0):299 – 307, 2014.

[66] Richard Bellmann. *Dynamic Programming*. Princeton University Press, Princeton, 1957.

[67] T. W. T Ivens. Predictive cruise control for heavy duty vehicles, 2012.

[68] W. Dib, A. Chasse, D. Di Domenico, P. Moulin, and A. Sciarretta. Evaluation of the energy efficiency of a fleet of electric vehicle for eco-driving application. *Oil and Gas Science and Technology–Revue IFP Energies nouvelles*, 67(4):589–599, 2012.

[69] Felipe Jimenez and Wilmar Cabrera-Montiel. System for road vehicle energy optimization using real time road and traffic information. *Energies*, 7(6):3576, 2014.

[70] Meng Wang, W. Daamen, S. Hoogendoorn, and B. van Arem. Potential impacts of ecological adaptive cruise control systems on traffic and environment. *Intelligent Transport Systems, IET*, 8(2):77–86, March 2014.

[71] Felicitas Mensing, Eric Bideaux, Rochdi Trigui, and Helene Tattegrain. Trajectory optimization for eco-driving taking into account traffic constraints. *Transportation Research Part D: Transport and Environment*, 18(0):55 – 61, 2013. ISSN 1361-9209.

[72] Felipe Jiménez and Wilmar Cabrera-Montiel. System for road vehicle energy optimization using real time road and traffic information. *Energies*, 7(6):3576–3598, 2014.

[73] Yan Chen, Xiaodong Li, C. Wiet, and Junmin Wang. Energy management and driving strategy for in-wheel motor electric ground vehicles with terrain profile preview. *Industrial Informatics, IEEE Transactions on*, 10(3):1938–1947, Aug 2014. ISSN 1551-3203.

[74] Xiaohai Lin, Daniel Görges, and Steven Liu. Eco-driving assistance system for electric vehicles based on speed profile optimization. In *Control Applications (CCA), 2014 IEEE Conference on*, pages 629–634, Oct 2014.

[75] M. Kuriyama, S. Yamamoto, and M. Miyatake. Theoretical study on eco-driving technique for an electric vehicle with dynamic programming. In *Electrical Machines and Systems (ICEMS), 2010 International Conference on*, pages 2026–2030, Oct 2010.

[76] Erik Hellstroem. *Look-Ahead Control of Heavy Vehicles*. PhD thesis, Linkoeping University, 2010.

[77] Q. Cheng and L. Nouveliere. A new eco-driving assistance system for a light vehicle: Energy management and speed optimization. In *Intelligent Vehicles Symposium (IV), 2013 IEEE*, pages 1434–1439, June 2013.

[78] G. De Nunzio, C. Canudas de Wit, P. Moulin, and D. Di Domenico. Eco-driving in urban traffic networks using traffic signal information. In *Decision and Control (CDC), 2013 IEEE 52nd Annual Conference on*, pages 892–898, Dec 2013.

[79] S. Glaser, O. Orfila, L. Nouveliere, R. Potarusov, S. Akhegaonkar, F. Holzmann, and V. Scheuch. Smart and green acc, adaptation of the acc strategy for electric vehicle with regenerative capacity. In *Intelligent Vehicles Symposium (IV), 2013 IEEE*, pages 970–975, June 2013.

[80] M.A.S. Kamal, M. Mukai, J. Murata, and T. Kawabe. Ecological driving based on preceding vehicle prediction using mpc. In *18th IFAC World Congress, Milano, Italy*, pages 3843–3848, Aug 2011.

[81] D. Yamaguchi, M. A. S. Kamal, M. Mukai, and T. Kawabe. Model predictive control for automobile ecological driving using traffic signal information. *Journal of System Design and Dynamics*, 6(3):297–309, 2012.

[82] M.A.S. Kamal, M. Mukai, J. Murata, and T. Kawabe. Model predictive control of vehicles on urban roads for improved fuel economy. *Control Systems Technology, IEEE Transactions on*, 21(3):831–841, May 2013. ISSN 1063-6536.

[83] MAS Kamal, S Taguchi, and T Yoshimura. Efficient vehicle driving on multi-lane roads using model predictive control under a connected vehicle environment. In *Intelligent Vehicles Symposium (IV), 2015 IEEE*, pages 736–741. IEEE, 2015.

[84] Kaijiang Yu, Junqi Yang, and Daisuke Yamaguchi. Model predictive control for hybrid vehicle ecological driving using traffic signal and road slope information. *Control Theory and Technology*, 13(1):17–28, 2015.

[85] N. Kohut, K. Hedrick, and F. Borrelli. Integrating traffic data and model predictive control to improve fuel economy. In *12th IFAC Symposium on Control in Transportation Systems (2009)*, pages 155–160, 2009.

[86] Marcus Kalabis and Steffen Mueller. A model predictive approach for a fuel efficient cruise control system. In Heike Proff, Joerg Schoenharting, Dieter Schramm, and Juergen Ziegler, editors, *Zukuenftige Entwicklungen in der Mobilitaet*, pages 201–211. Gabler Verlag, 2012.

[87] Dang Ruina, He Chaozhe, Zhang Qiang, Li Keqiang, and Li Yusheng. Acc of electric vehicles with coordination control of fuel economy and tracking safety. In *Intelligent Vehicles Symposium (IV), 2012 IEEE*, pages 240–245, June 2012.

[88] M. Henzler, M. Buchholz, and K. Dietmayer. Online velocity trajectory planning for manual energy efficient driving of heavy duty vehicles using model predictive control. In *Intelligent Transportation Systems (ITSC), 2014 IEEE 17th International Conference on*, pages 1814–1819, Oct 2014.

[89] Christian Kirches. *Fast Numerical Methods for Mixed-Integer Nonlinear Model-Predictive Control*. PhD thesis, University of Heidelberg, 2010.

[90] Frank Lattemann, Konstantin Neiss, Stephan Terwen, and Thomas Connolly. The predictive cruise control–a system to reduce fuel consumption of heavy duty trucks. Technical report, SAE Technical paper, 2004.

[91] Stephan Terwen. *Vorausschauende Längsregelung schwerer Lastkraftwagen*. PhD thesis, Karlsruher Institut fuer Technologie, 2010.

[92] A. Sciarretta, G. De Nunzio, and L.L. Ojeda. Optimal ecodriving control: Energy-efficient driving of road vehicles as an optimal control problem. *Control Systems, IEEE*, 35(5):71–90, Oct 2015.

[93] Daimler AG. Distronic, 2014. URL http://m.mercedes-benz.de/de_DE/distronic/detail.html. [Online; December, 22nd, 2014].

[94] D. Corona and B. De Schutter. Adaptive cruise control for a smart car: A comparison benchmark for mpc-pwa control methods. *Control Systems Technology, IEEE Transactions on*, 16(2):365–372, March 2008.

[95] E. Kural and B.A. Güvenç. Model predictive adaptive cruise control. In *Systems Man and Cybernetics (SMC), 2010 IEEE International Conference on*, pages 1455–1461, Oct 2010.

[96] Tao Chen, Yugong Luo, and Keqiang Li. Multi-objective adaptive cruise control based on nonlinear model predictive algorithm. In *Vehicular Electronics and Safety (ICVES), 2011 IEEE International Conference on*, pages 274–279, July 2011.

[97] Shengbo Li, Keqiang Li, R. Rajamani, and Jianqiang Wang. Model predictive multi-objective vehicular adaptive cruise control. *Control Systems Technology, IEEE Transactions on*, 19(3):556–566, May 2011.

[98] Li-hua Luo, Hong Liu, Ping Li, and Hui Wang. Model predictive control for adaptive cruise control with multi-objectives: comfort, fuel-economy, safety and car-following. *Journal of Zhejiang University SCIENCE A*, 11(3):191–201, 2010.

[99] Daimler AG. Predictive powertrain control - clever cruise control helps save fuel, 2012. URL http://media.daimler.com/dcmedia/0-921-1404221-1-1491206-1-0-0-0-0-0-0-0-0-0-0-0-0-0-0.html. [Online; December, 22nd, 2014].

[100] Martin Roth. Betriebsstrategie. In *Energiemanagement im Kraftfahrzeug*, ATZ/MTZ-Fachbuch, pages 323–365. Springer Fachmedien Wiesbaden, 2014.

[101] Philip Markschläger, Hans-Georg Wahl, Frank Weberbauer, and Matthias Lederer. Assistance system for higher fuel efficiency. *Auto Tech Review*, 1(12):40–45, 2012. ISSN 2250-3390.

[102] Jing Sun. Optimisation-based control for electrified vehicles: challenges and opportunities. *Journal of Control and Decision*, 2(1):46–63, 2015.

[103] James Ferguson. A brief survey of the history of the calculus of variations and its applications, 2004. URL http://arxiv.org/abs/math/0402357v1. [Online; December, 22nd, 2014].

[104] O. Foellinger. *Optimale Regelung und Steuerung*. Oldenbourg Verlag, Muenchen, 1994.

[105] Hans Georg Bock and Karl-Josef Plitt. A multiple shooting algorithm for direct solution of optimal control problems. 1983.

[106] Oskar von Stryk. Numerical solution of optimal control problems by direct collocation. In *in Optimal Control, (International Series in Numerical Mathematics 111*, pages 129–143, 1993.

[107] Stephen Boyd and Lieven Vandenberghe. *Convex optimization*. Cambridge university press, 2009.

[108] Mikhail K Kozlov, Sergei P Tarasov, and Leonid G Khachiyan. The polynomial solvability of convex quadratic programming. *USSR Computational Mathematics and Mathematical Physics*, 20(5):223–228, 1980.

[109] Christopher V. Rao and James B. Rawlings. Linear programming and model predictive control. *Journal of Process Control*, 10(2–3):283 – 289, 2000.

[110] David Q Mayne, James B Rawlings, Christopher V Rao, and Pierre OM Scokaert. Constrained model predictive control: Stability and optimality. *Automatica*, 36 (6):789–814, 2000.

[111] Alberto Bemporad, Manfred Morari, Vivek Dua, and Efstratios N Pistikopoulos. The explicit linear quadratic regulator for constrained systems. *Automatica*, 38 (1):3–20, 2002.

[112] Petter TøNdel, Tor Arne Johansen, and Alberto Bemporad. An algorithm for multi-parametric quadratic programming and explicit mpc solutions. *Automatica*, 39(3):489–497, 2003.

[113] Yang Wang and Stephen Boyd. Fast model predictive control using online optimization. *Control Systems Technology, IEEE Transactions on*, 18(2):267–278, 2010.

[114] M. Kvasnica, P. Grieder, and M. Baotić. Multi-Parametric Toolbox (MPT), 2004. URL http://control.ee.ethz.ch/~mpt/.

[115] Raphael Cagienard, Pascal Grieder, Eric C Kerrigan, and Manfred Morari. Move blocking strategies in receding horizon control. *Journal of Process Control*, 17(6): 563–570, 2007.

[116] N Lawrence Ricker. Use of quadratic programming for constrained internal model control. *Industrial & Engineering Chemistry Process Design and Development*, 24 (4):925–936, 1985.

[117] S Joe Qin and Thomas A Badgwell. A survey of industrial model predictive control technology. *Control engineering practice*, 11(7):733–764, 2003.

[118] Petter Tøndel and Tor A Johansen. Complexity reduction in explicit linear model predictive control. *15th IFAC World Congr. Autom. Control, Barcelona, Spain*, 2002.

[119] Ravi Gondhalekar and Jun-ichi Imura. Least-restrictive move-blocking model predictive control. *Automatica*, 46(7):1234–1240, 2010.

[120] G. Valencia-Palomo, M Pelegrinis, J. A. Rossiter, and R. Gondhalekar. A move-blocking strategy to improve tracking in predictive control. In *American Control Conference (ACC), 2010*, pages 6293–6298. IEEE, 2010.

[121] Dewei Li, Yugeng Xi, and Zongli Lin. An improved design of aggregation-based model predictive control. *Systems & Control Letters*, 62(11):1082–1089, 2013.

[122] Jacques Vlassenbroeck and Rene Van Dooren. A chebyshev technique for solving nonlinear optimal control problems. *Automatic Control, IEEE Transactions on*, 33(4):333–340, 1988.

[123] Gamal Elnagar, Mohammad A Kazemi, and Mohsen Razzaghi. The pseudospectral legendre method for discretizing optimal control problems. *Automatic Control, IEEE Transactions on*, 40(10):1793–1796, 1995.

[124] G. Valencia-Palomo, J.A. Rossiter, Colin Jones, Ravi Gondhalekar, and B. Khan. Alternative parameterisations for predictive control: how and why? In *Proceeding of the American Control Conference*, 2011.

[125] I Michael Ross and Mark Karpenko. A review of pseudospectral optimal control: From theory to flight. *Annual Reviews in Control*, 36(2):182–197, 2012.

[126] Johannes Unger, Martin Kozek, and Stefan Jakubek. Reduced order optimization for model predictive control using principal control moves. *Journal of Process Control*, 22(1):272–279, 2012.

[127] Keck Voon Ling, Jan Maciejowski, Arthur Richards, and Bing Fang Wu. Multiplexed model predictive control. *Automatica*, 48(2):396–401, 2012.

[128] Lino Guzzella and Antonio Sciarretta. *Vehicle propulsion systems*, volume 1. Springer, 2007.

[129] T. Schwickart, H. Voos, J.-R. Hadji-Minaglou, and M. Darouach. A novel model-predictive cruise controller for electric vehicles and energy-efficient driving. In *Advanced Intelligent Mechatronics (AIM), 2014 IEEE/ASME International Conference on*, pages 1067–1072, July 2014.

[130] T. Schwickart, H. Voos, J.-R. Hadji-Minaglou, M. Darouach, and A. Rosich. Design and simulation of a real-time implementable energy-efficient model-predictive cruise controller for electric vehicles. *Journal of the Franklin Institute*, 352(2):603 – 625, 2015. Special Issue on Control and Estimation of Electrified vehicles.

[131] L. Nouveliere, S. Mammar, and H.-T. Luu. Energy saving and safe driving assistance system for light vehicles: Experimentation and analysis. In *Networking, Sensing and Control (ICNSC), 2012 9th IEEE International Conference on*, pages 346–351, April 2012.

[132] S. Stefanov. *Separable Programming - Theory and Methods*. Kluwer Academic Publishers, 2001.

[133] S. Boyd and L. Vandenberghe. *Convex Optimization*. Cambridge University Press New York, 2004.

[134] Alessandro Magnani and StephenP. Boyd. Convex piecewise-linear fitting. *Optimization and Engineering*, 10(1):1–17, 2009. ISSN 1389-4420.

[135] Y.-C. Ho A. E. Bryson, Jr. *Applied Optimal Control, Optimization, Estimation and Control*. Halsted Press, John Wiley and Sons, New York, 1975.

[136] W. Dib, L. Serrao, and A. Sciarretta. Optimal control to minimize trip time and energy consumption in electric vehicles. In *Vehicle Power and Propulsion Conference (VPPC), 2011 IEEE*, pages 1–8, Sept 2011.

[137] Christopher V Rao, Stephen J Wright, and James B Rawlings. Application of interior-point methods to model predictive control. *Journal of optimization theory and applications*, 99(3):723–757, 1998.

[138] Stephen J Wright. Applying new optimization algorithms to model predictive control. In *AIChE Symposium Series*, volume 93, pages 147–155. Citeseer, 1997.

[139] H.J. Ferreau, C. Kirches, A. Potschka, H.G. Bock, and M. Diehl. qpOASES: A parametric active-set algorithm for quadratic programming. *Mathematical Programming Computation*, 6(4):327–363, 2014.

[140] R. Fletcher. *Practical Methods of Optimization*. Wiley, 1987.

[141] Steven J. Wright. *Primal-Dual Interior-Point Methods*. SIAM, 1997.

[142] T. Terlaky. *Interior Point Methods of Mathematical Programming*. Applied Optimization. Springer, 1996.

[143] T. Schwickart, H. Voos, and M. Darouach. A real-time implementable model-predictive cruise controller for electric vehicles and energy-efficient driving. In *Control Applications (CCA), 2014 IEEE Conference on*, pages 617–622, Oct 2014.

[144] Jacob Mattingley and Stephen Boyd. Cvxgen: a code generator for embedded convex optimization. *Optimization and Engineering*, 13(1):1–27, 2012. ISSN 1389-4420.

[145] Morgan Quigley, Ken Conley, Brian Gerkey, Josh Faust, Tully Foote, Jeremy Leibs, Rob Wheeler, and Andrew Y Ng. Ros: an open-source robot operating system. In *ICRA workshop on open source software*, volume 3, page 5, 2009.

[146] Gaël Guennebaud, Benoît Jacob, et al. Eigen v3. http://eigen.tuxfamily.org, 2010.

[147] Stefano Longo, Eric C. Kerrigan, Keck Voon Ling, and George A. Constantinides. A parallel formulation for predictive control with nonuniform hold constraints. *Annual Reviews in Control*, 35(2):207 – 214, 2011.

[148] Dewei Li, Yugeng Xi, and Zongli Lin. An improved design of aggregation-based model predictive control. *Systems & Control Letters*, 62(11):1082–1089, 2013.

[149] Rohan C Shekhar and Chris Manzie. Optimal move blocking strategies for model predictive control. *Automatica*, 61:27–34, 2015.

[150] L. El Ghaoui. Optimization models and applications. http://livebooklabs.com/keeppies/c5a5868ce26b8125, December 2014.