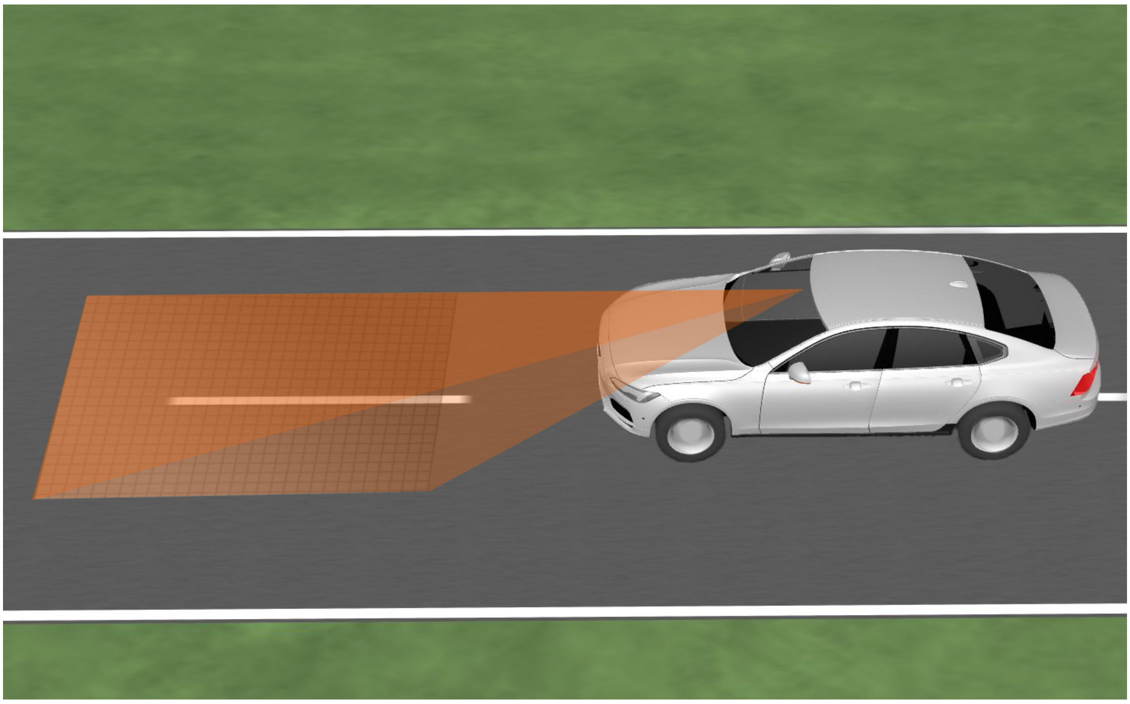
CHALMERS
UNIVERSITY OF TECHNOLOGY



Real-Time Nonlinear Model Predictive Control for Semi-Active Suspension with Road Preview

Master's thesis in Systems, Control and Mechatronics

FREDRIK KJELLBERG, SIMON SUNDELL

Department of Electrical Engineering
CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden 2018

Master’s thesis 2018:EX086

Real-Time Nonlinear Model Predictive Control for Semi-Active Suspension with Road Preview

FREDRIK KJELLBERG
SIMON SUNDELL

Department of Electrical Engineering
Division of Systems and Control
Mechatronics research group
Chalmers University of Technology
Gothenburg, Sweden 2018

Real-Time Nonlinear Model Predictive Control for Semi-Active Suspension with Road Preview
FREDRIK KJELLBERG
SIMON SUNDELL

Supervisor: Pontus Carlsson, Volvo Car Corporation
Examiner: Paolo Falcone, Department of Electrical Engineering

Master's Thesis 2018:EX086
Department of Electrical Engineering
Division of Systems and Control
Mechatronics research group
Chalmers University of Technology
SE-412 96 Gothenburg
Telephone +46 31 772 1000

Cover: Visualization of a car equipped with a vision system that is gathering preview information of the road.

Typeset in LaTeX
Gothenburg, Sweden 2018

Real-Time Nonlinear Model Predictive Control for Semi-Active Suspension with Road Preview

FREDRIK KJELLBERG, SIMON SUNDELL
Department of Electrical Engineering
Chalmers University of Technology

Abstract

Semi-active suspensions can be used to improve ride comfort in cars since the stiffness of the four dampers can be controlled. The ride comfort can be further improved by utilizing preview of the road height profile in the control strategy. Model Predictive Control (MPC) is a suitable control strategy for this purpose because it can explicitly incorporate the road preview as well as hard constraints on the force exerted by the semi-active dampers, which is necessary for formulating the semi-active suspension control problem. A big challenge in control of semi-active suspensions is that a damper can only dissipate energy and this constraint makes a real-time implementation of MPC difficult to achieve. This thesis targets the real-time problem and presents an implementation of Nonlinear Model Predictive Control (NMPC) as well as the simplifications made to achieve a real-time implementation. The controller is based on a vertical full-car model capable of describing the heave, roll and pitch motions of the car and minimizes heave, roll and pitch accelerations in order to improve ride comfort. The performance of the real-time controller is compared to a reference controller that represents the best-case performance. Simulation results show that ride comfort is improved over a passive suspension and that the performance of the real-time controller is close to the performance of the reference controller. A vehicle implementation that incorporates nonlinear damper characteristics and limits on suspension travel is also presented together with experimental results obtained with a car equipped with semi-active dampers and a vision system for gathering the road preview.

Keywords: Semi-active suspension, Model predictive control, Road preview, Real-time, Full-car model

Acknowledgements

We would like to express our biggest gratitude to Pontus Carlsson, our supervisor at Volvo Car Corporation, for his excellent support and advice throughout the thesis. We would also like to thank Albin Dahlin and Stavros Angelis for their help with more technical matters. Finally, we would like to thank Paolo Falcone, our examiner and supervisor at Chalmers, for his expert advice regarding both technical matters and the report.

Fredrik Kjellberg & Simon Sundell, Gothenburg, June 2018

Nomenclature

Abbreviations

COG	Center Of Gravity
DAE	Differential Algebraic Equation
FFT	Fast Fourier Transform
FL	Front Left
FR	Front Right
LQR	Linear Quadratic Regulator
LTI	Linear Time Invariant
MIQP	Mixed Integer Quadratic Programming
MPC	Model Predictive Control
NMPC	Nonlinear Model Predictive Control
ODE	Ordinary Differential Equation
QP	Quadratic Programming
RL	Rear Left
RMS	Root Mean Squared error
RR	Rear Right

Contents

List of Figures

List of Tables

1

Introduction

Ride comfort is an important attribute in cars. It is achieved by isolating the passengers from vibrations induced by the road. The main component that affects ride comfort is the suspension and a comfortable ride is indicated by low vehicle body accelerations. Improving comfort does not only increase passenger satisfaction but it also improves safety as it reduces driver fatigue, especially during long trips, [1]. Handling is also an important attribute in cars and it is achieved by making the wheels and car body follow the road as closely as possible. Comfort and handling are, however, conflicting objectives when tuning passive suspensions and a trade-off has to be made between the two, [2]. Semi-active suspensions can be used to make the trade-off between comfort and handling less significant and thus improve the vehicle behaviour, [2].

With semi-active suspensions it is possible to control the damper stiffness. A limitation that follows is that it is only possible to dissipate energy from the system since a damper can only generate a force that has opposite direction to its deflection velocity. In contrast, an active suspension makes it possible to add energy to the system, for example by lifting the car, which enables better performance. Despite this, semi-active suspensions are more widely used because they do not require as expensive actuators as active suspensions, [3].

The control strategy for semi-active suspensions is usually to react to some measured body motion signal and control the actuators as quickly as possible. However, as technology in active safety and autonomous driving progresses, more advanced sensors are installed in new cars. Various types of vision and radar systems are used to detect the surroundings of the car, such as other vehicles and pedestrians. These sensor systems can be used to gather information about the road surface in front of the vehicle, i.e. a preview of the road height profile which can be incorporated in the design of a semi-active suspension controller.

A control strategy well suited for utilizing the road preview is Model Predictive Control (MPC). MPC uses a system model to compute the optimal sequence of control signals over a prediction horizon, applies the first control signal and then repeats the process at each sampling instant. It is also possible to incorporate hard constraints in the MPC, such as constraints on the control signal and the state variables, which is necessary for formulating the semi-active suspension control problem.

Previous works have shown potential when using MPC with preview for semi-active suspension. In [3], MPC based on a full-car model was implemented and verified through simulations with the objective to improve ride comfort and handling. The full-car model is suitable since it can represent the vehicle dynamics that is most relevant for the suspension. The full-car model describes the motions of the car body: heave, roll and pitch, and the vertical motion of the four wheels.

The drawback of MPC in general is that the computational time required to solve for the optimal control signals tends to be large. The additional difficulty with semi-active suspensions is that the constraints on the damper force, due to the fact that the semi-active damper can only dissipate energy, makes the optimization problem nonlinear. This increases the computational complexity of the problem significantly. For semi-active suspension with MPC to be useful in practice, good performance in simulation is not enough. A real-time implementation also has to be possible and that has not yet been achieved when using a full-car model to describe the vehicle dynamics in the MPC.

1.1 Purpose

The purpose of the thesis is to utilize preview of the road height profile in a semi-active suspension in order to improve ride comfort. That is, to minimize vehicle body accelerations as much as possible with an improved driving experience as a result.

1.2 Objective

The objective of the thesis is to formulate and implement an MPC for semi-active suspension that utilizes road preview. The MPC should be based on a full-car model and the implementation should be able to run in real-time. An additional objective is to test the implementation in a real car. Upon completion of the project, the following research questions should be answered:

1) What is the best way to utilize road preview together with MPC for semi-active suspension in order to improve ride comfort?
2) What simplifications or assumptions have to be made in order to make a real-time implementation feasible?
3) How does the performance of a real-time implementation differ from an implementation that does not consider the real-time constraints?

1.3 Scope

The scope of the thesis is focused around the design and implementation of a MPC for semi-active suspension with road preview. In order to implement the controller in a real car, additional work is required that will be outside of the scope. The preview information that is fed to the controller is assumed to be pre-processed, always available and covering the entire distance from the back wheels until the maximum range of the vision system. Communication with sensors and actuators in the car is assumed to be in place.

A realistic car model to test and evaluate the controller on is available and will be used during the project. The performance of the controller during more dynamic manoeuvring of the car, such as turning or breaking, will not be considered even though a semi-active suspension can affect the performance of the vehicle in these manoeuvres. Such scenarios will not be considered because during such manoeuvring, handling needs to be prioritized and the controller will be designed to prioritize comfort. Another reason is that the controller will not be optimal when turning, since the calculated path of the road preview will not be followed.

The controller will be tuned and evaluated by assessing vehicle body accelerations rather than the subjective feeling of ride comfort. It makes sense to minimize the accelerations since that is what the driver and the passengers feels.

1.4 Outline of the report

Chapter 2 contains theory about relevant vehicle dynamics and introduces the different kinds of suspensions. It also introduces the control problem that needs to be solved when trying to improve ride comfort with semi-active suspension.

Chapter 3 contains theory about linear MPC and nonlinear MPC.

Chapter 4 describes the control design of the proposed controllers and the real-time implementation. It also contains a review of related research.

Chapter 5 presents simulation testing of the implemented controllers and obtained results.

Chapter 6 describes further additions to the controller in order to make it work in a vehicle implementation and presents experimental results.

Chapter 7 contains a discussion of the obtained results together with suggestions for future work and the conclusion of the thesis.

2

Vehicle model and control problem

This chapter covers the modelling of the vertical full-car model starting with the fundamental quarter-car model, which is the building block for both the half-car and full-car suspension models. Then the different kinds of suspensions (passive, semi-active and active) are described. Finally, the semi-active suspension control problem is described.

2.1 Quarter-car model

The quarter-car model describes the vertical dynamics of the interactions between the road, the wheel and the body. The wheel mass is connected to the road through the tire, which is almost a direct connection. Hence, this mass is referred to as the unsprung mass. The body mass is connected to the wheel through a spring and a damper and is therefore referred to as the sprung mass. The quarter-car model has two degrees of freedom, one associated with the sprung mass and one with the unsprung mass. A passive quarter-car suspension model is shown in Figure 2.1. The variables and constants can be found in Table 2.1.

Table 2.1: Description of the parameters used in the passive quarter-car model.

Symbol	***Quantity***	***Unit***
k_s	Spring constant	N/m
k_t	Spring constant, tire	N/m
m_s	Sprung mass	kg
m_u	Unsprung mass	kg
b_s	damping coefficient	$N/m/s$
z_s	Vertical displacement of sprung mass, heave	m
z_u	Vertical displacement of unsprung mass	m
z_r	Vertical displacement of road	m

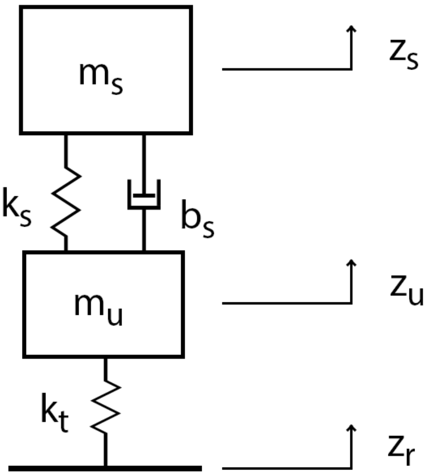


Figure 2.1: Passive quarter-car model. The parameters for this model can be seen in Table 2.1.

The dynamic equations of the passive quarter-car model describes the forces acting on the sprung mass (2.1) and the forces acting on the unsprung mass (2.2) and are formulated as:

$$m_s\ddot{z}_s = -k_s(z_s - z_u) - b_s(\dot{z}_s - \dot{z}_u), \tag{2.1}$$

$$m_u\ddot{z}_u = k_s(z_s - z_u) + b_s(\dot{z}_s - \dot{z}_u) - k_t(z_u - z_r). \tag{2.2}$$

2.2 Full-car model

This section describes the full-car model that is modelled with an actuator that can exert a force on each wheel. Using an additional force as control signal is a common way to model an active or a semi-active suspension. The full-car model is based on four quarter-car models that are connected to the same sprung mass and it has seven degrees of freedom: heave, roll, pitch and vertical displacement of the four unsprung masses. There is also a half-car model that is based on two quarter-car models, connected to the same sprung mass. The half-car model has four degrees of freedom: heave, pitch and vertical displacement of the two unsprung masses. The modelling of the half-car model and the full-car model is very similar, so only the full-car model is derived here. An illustration of the full-car model is shown in Figure 2.2 and the variables and constants can be found in Table 2.2.

Table 2.2: Parameters for full-car model, $i \in \{f, r\}$, $j \in \{l, r\}$ which stands for $i \in \{front, rear\}$, $j \in \{left, right\}$.

Symbol	***Quantity***	***Unit***
m_s	Sprung mass	kg
m_{uij}	Unsprung mass	kg
I_x	Roll inertia	kgm^2
I_y	Pitch inertia	kgm^2
z_s	Vertical displacement of sprung mass, heave	m
z_{sij}	Vertical displacement of sprung mass at each corner	m
θ	Roll	rad
φ	Pitch	rad
z_{uij}	Vertical displacement of unsprung mass	m
z_{rij}	Vertical displacement of road	m
k_{sij}	Spring constant, sprung mass	N/m
k_{tij}	Spring constant, tire	N/m
b_{sij}	Damping coefficient sprung mass	N/m/s
t_f	COG-distance front	m
t_r	COG-distance rear	m
l_l	COG-distance left	m
l_r	COG-distance right	m
F_{ij}	Force acting on each unsprung mass	N
F_{tij}	Tire force	N
u_{ij}	Control signal	N

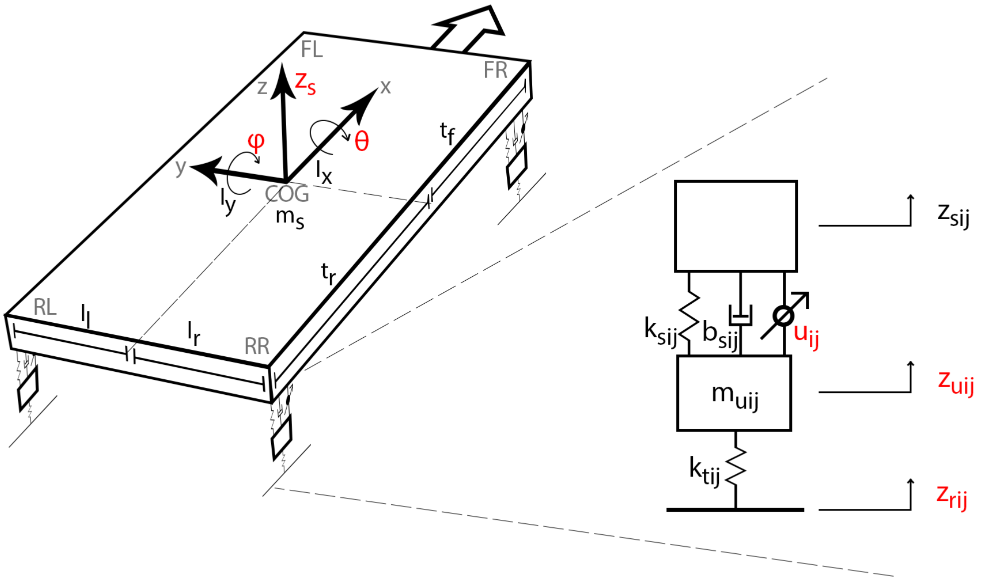


Figure 2.2: Full-car model with additional force between sprung and unsprung mass as control signal. The parameters for this model can be seen in Table 2.2.

In order to describe the dynamics of the full-car model the following equations are derived. Equations of equilibrium:

$$m_s\ddot{z}_s = -F_{fl} - F_{fr} - F_{rl} - F_{rr}, \tag{2.3}$$

$$m_{ufl}\ddot{z}_{ufl} = F_{fl} - F_{tfl}, \tag{2.4}$$

$$m_{ufr}\ddot{z}_{ufr} = F_{fr} - F_{tfr}, \tag{2.5}$$

$$m_{url}\ddot{z}_{url} = F_{rl} - F_{trl}, \tag{2.6}$$

$$m_{urr}\ddot{z}_{urr} = F_{rr} - F_{trr}, \tag{2.7}$$

$$I_x\ddot{\theta} = (F_{fr} + F_{rr})l_r - (F_{fl} + F_{rl})l_l, \tag{2.8}$$

$$I_y\ddot{\varphi} = (F_{fl} + F_{fr})t_f - (F_{rl} + F_{rr})t_r. \tag{2.9}$$

Forces acting on the sprung mass, located at each corner of the sprung mass:

$$F_{fl} = k_{sfl}(z_{sfl} - z_{ufl}) + b_{sfl}(\dot{z}_{sfl} - \dot{z}_{ufl}) + u_{fl}, \tag{2.10}$$

$$F_{fr} = k_{sfr}(z_{sfr} - z_{ufr}) + b_{sfr}(\dot{z}_{sfr} - \dot{z}_{ufr}) + u_{fr}, \tag{2.11}$$

$$F_{rl} = k_{srl}(z_{srl} - z_{url}) + b_{srl}(\dot{z}_{srl} - \dot{z}_{url}) + u_{rl}, \tag{2.12}$$

$$F_{rr} = k_{srr}(z_{srr} - z_{urr}) + b_{srr}(\dot{z}_{srr} - \dot{z}_{urr}) + u_{rr}. \tag{2.13}$$

The vertical displacement at each corner z_{ij} can be described by the vertical displacement at COG, roll and pitch:

$$z_{sfl} = z_s - t_f sin(\varphi) + l_l sin(\theta), \tag{2.14}$$

$$z_{sfr} = z_s - t_f sin(\varphi) - l_r sin(\theta), \tag{2.15}$$

$$z_{srl} = z_s + t_r sin(\varphi) + l_l sin(\theta), \tag{2.16}$$

$$z_{srr} = z_s + t_r sin(\varphi) - l_r sin(\theta). \tag{2.17}$$

The vertical displacement rate at each corner are calculated by taking the derivative of (2.14) - (2.17):

$$\dot{z}_{sfl} = \dot{z}_s - t_f cos(\varphi)\dot{\varphi} + l_l cos(\theta)\dot{\theta}, \tag{2.18}$$

$$\dot{z}_{sfr} = \dot{z}_s - t_f cos(\varphi)\dot{\varphi} - l_r cos(\theta)\dot{\theta}, \tag{2.19}$$

$$\dot{z}_{srl} = \dot{z}_s + t_r cos(\varphi)\dot{\varphi} + l_l cos(\theta)\dot{\theta}, \tag{2.20}$$

$$\dot{z}_{srr} = \dot{z}_s + t_r cos(\varphi)\dot{\varphi} - l_r cos(\theta)\dot{\theta}. \tag{2.21}$$

Forces from the tires acting on the unsprung masses:

$$F_{tfl} = k_{tfl}(z_{ufl} - z_{rfl}), \tag{2.22}$$

$$F_{tfr} = k_{tfr}(z_{ufr} - z_{rfr}), \tag{2.23}$$

$$F_{trl} = k_{trl}(z_{url} - z_{rrl}), \tag{2.24}$$

$$F_{trr} = k_{trr}(z_{urr} - z_{rrr}). \tag{2.25}$$

2.2.1 Linearization

In order to design an MPC based on convex optimization to control the dampers of the car, the full-car model needs to be linear. The model is linearized by using the small angle approximation, i.e. $sin(\phi) \approx \phi$ and $cos(\phi) \approx 1$. This is a good approximation since in all normal road conditions, the roll and pitch angles of the car will be relatively small. Then, (2.14)-(2.17) can be rewritten as:

$$z_{sfl} = z_s - t_f\varphi + l_l\theta, \tag{2.26}$$
$$z_{sfr} = z_s - t_f\varphi - l_r\theta, \tag{2.27}$$
$$z_{srl} = z_s + t_r\varphi + l_l\theta, \tag{2.28}$$
$$z_{srr} = z_s + t_r\varphi - l_r\theta, \tag{2.29}$$

and (2.18)-(2.21) as:

$$\dot{z}_{sfl} = \dot{z}_s - t_f\dot{\varphi} + l_l\dot{\theta}, \tag{2.30}$$
$$\dot{z}_{sfr} = \dot{z}_s - t_f\dot{\varphi} - l_r\dot{\theta}, \tag{2.31}$$
$$\dot{z}_{srl} = \dot{z}_s + t_r\dot{\varphi} + l_l\dot{\theta}, \tag{2.32}$$
$$\dot{z}_{srr} = \dot{z}_s + t_r\dot{\varphi} - l_r\dot{\theta}. \tag{2.33}$$

2.2.2 State space representation

When linearized, the full-car model can be described by a state space representation. A state vector $\mathbf{x}$, a control signal vector $\mathbf{u}$ and a disturbance input vector $\mathbf{w}$ are defined as:

$$\mathbf{x} = \begin{bmatrix} z_s, & \theta, & \varphi, & z_{ufl}, & z_{ufr}, & z_{url}, & z_{urr}, & \dot{z}_s, & \dot{\theta}, & \dot{\varphi}, & \dot{z}_{ufl}, & \dot{z}_{ufr}, & \dot{z}_{url}, & \dot{z}_{urr} \end{bmatrix}^T, \tag{2.34}$$

$$\mathbf{u} = \begin{bmatrix} u_{fl}, & u_{fr}, & u_{rl}, & u_{rr} \end{bmatrix}^T, \tag{2.35}$$

$$\mathbf{w} = \begin{bmatrix} z_{rfl}, & z_{rfr}, & z_{rrl}, & z_{rrr} \end{bmatrix}^T, \tag{2.36}$$

where z_s is heave position and is defined as deviation from initial position. θ and φ are the roll angle and the pitch angle of the car body. z_{ufl}, z_{ufr}, z_{url} and z_{urr} are the positions for the unsprung masses which are defined as the displacement from their initial positions. The remaining states are the derivatives of the first seven state variables. The initial values for the state vector are defined as:

$$\mathbf{x}(0) = \begin{bmatrix} 0, & 0, & 0, & 0, & 0, & 0, & 0, & 0, & 0, & 0, & 0, & 0, & 0, & 0 \end{bmatrix}^T. \tag{2.37}$$

The plant to be controlled can be written on state space form by using (2.3)-(2.9), (2.10)-(2.13), (2.22)-(2.25), (2.26)-(2.29) and (2.30)-(2.33):

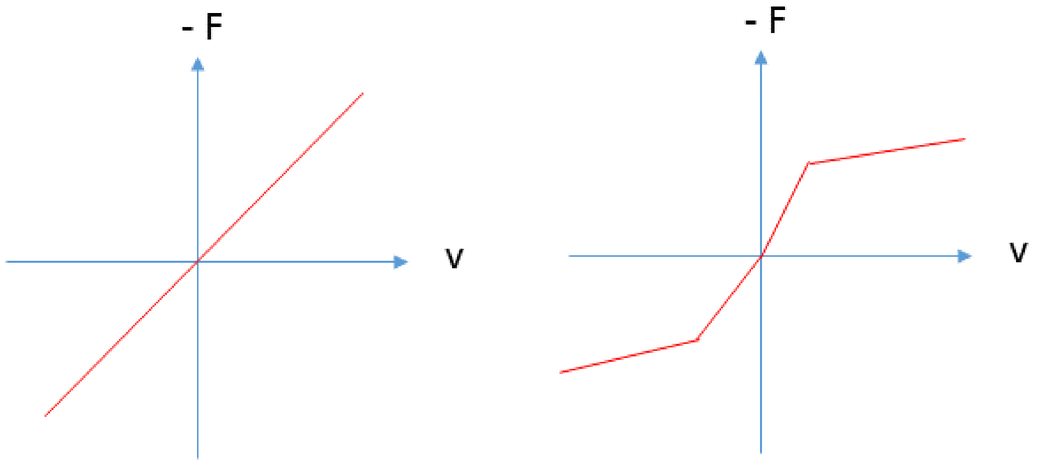
$$\dot{\mathbf{x}} = A\mathbf{x} + B_u\mathbf{u} + B_w\mathbf{w}, \tag{2.38}$$
$$\mathbf{y} = C\mathbf{x} + D\mathbf{u}, \tag{2.39}$$

where A, B, C and D are appropriate matrices.

2.3 Passive suspension

In the full-car model, described in section 2.2, the forces from the springs and dampers in the car are described. The control signals, which are additional forces acting on the vehicle body is used to model an active or semi-active suspension. The additional forces are not present in a passive suspension, which only consists of springs and passive dampers. The purpose of the spring is to absorb shock energy due to unevenness in the road profile and convert it into potential energy while the purpose of the damper is to dissipate energy, i.e. absorb and damp shock impulses caused by unevenness of the road.

With passive suspensions, the relationship between the force and the deflection velocity of the damper can be approximated as linear with the function $F = -bv$, where F is the force generated by the damper, v is the deflection velocity and b is the damping coefficient. For a real damper, this approximation is only valid for deflection velocities close to zero. Figure 2.3 shows a side by side comparison of the linear approximation and the more realistic damper characteristics.



(a) Linear damper characteristics. **(b)** Nonlinear damper characteristics.

Figure 2.3: Force exerted by a passive damper as a function of the deflection velocity with simplified and realistic characteristics.

2.4 Active and semi-active suspension

The initial motivations for active and semi-active damping came from the difficulties in tuning passive suspensions due to the trade-off between comfort and handling. The trade-off is less significant for active and semi-active suspensions since the force that is exerted by the actuators can be controlled. The biggest performance increase compared to passive suspension is achieved by using active suspension because the force that is applied by the actuators is independent of the velocity of the damper which can be seen in Figure 2.4. That means a car with active suspension can exert an independent force on the suspension to vertically control the movement of each wheel in relation to the vehicle body.

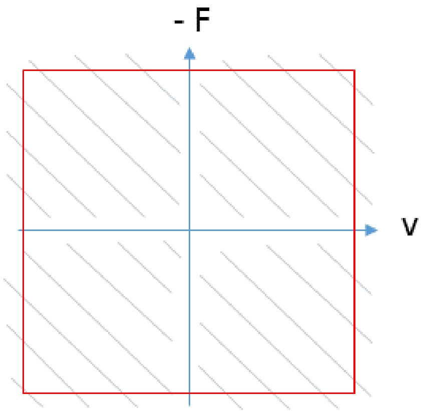
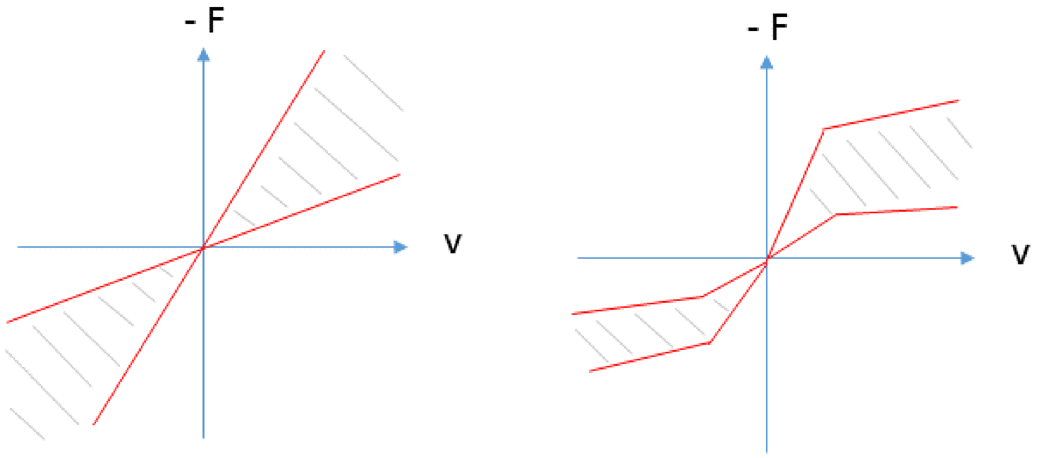


Figure 2.4: Constraints of the force that can be exerted for active suspension given as a function of the deflection velocity of the damper.

With semi-active suspensions, it is possible to change the damping force in the suspension. However, semi-active suspensions can only dissipate energy from the system, not add energy. This means that the controlled force always has the opposite direction to the velocity of the damper and cannot be chosen independently, as it can with active suspensions. Figure 2.5 shows both simplified and realistic characteristics of the force exerted by a semi-active damper and it can be seen that the approximation of the lower and upper bounds as linear functions, (a), is only similar to the more realistic, piece-wise linear bounds, (b), when the deflection velocity of the damper is small. Note that for comfort oriented vehicles, the force that is exerted during rebound (positive velocity) is generally larger than during compression (negative velocity).



(a) Simplified damper characteristics.

(b) Realistic damper characteristics.

Figure 2.5: Constraints on the force that can be exerted by semi-active suspension given as a function of the deflection velocity of the damper.

2.5 Primary and secondary ride

Ride comfort can be divided into primary ride and secondary ride where primary ride is associated with body accelerations with frequencies up to 4 Hz while secondary ride is associated with body accelerations with frequencies higher than 4 Hz, [4]. General body and suspension movements when driving over large bumps, long dips and low frequency undulations in the road results in primary ride. This accounts for both the rebound and the compression phase and is primarily controlled by the springs, dampers and anti-roll bars. Secondary ride is associated with driving on roads with smaller surface imperfections such as concrete and broken tarmac. This results in vibrations of higher frequencies and how well they are damped depends on the tire properties and bushing together with the suspension. To investigate the primary ride of a vehicle it is common to measure accelerations of the body and the wheel deflection. On the other hand, to investigate the secondary ride properties accelerations in the steering wheel and in the seat rail are usually measured. A semi-active suspension can improve primary ride and can also minimize the accelerations on the lower end of the secondary ride spectrum, i.e up to around 10 Hz. Accelerations with even higher frequencies are mainly absorbed by other parts of the suspension and the car seats.

2.6 Semi-active suspension control problem

The semi-active suspension control problem for improving ride comfort is about controlling the force exerted by the semi-active dampers in such a way that the heave, roll and pitch accelerations of the vehicle body are minimized. While doing this, the dissipative constraints of the semi-active dampers needs to be enforced, which means that the force needs to be within the bounds shown in Figure 2.5. Additionally, a preview of the road disturbance should be incorporated into the control problem in order to further improve the possible performance increase over a passive suspension. In a real car, there are also hard limits on the travel of the suspension that needs to be incorporated. A suitable control strategy for incorporating all the described aspects is MPC.

3

Model Predictive Control

This chapter gives a theoretical background of linear MPC and nonlinear MPC. First, linear MPC is explained and the formulation of a quadratic optimization problem is derived in order to get an easier understanding of how nonlinear MPC works.

3.1 Receding horizon idea

The core of model predictive control is to utilize the receding horizon idea. This principle can be seen in Figure 3.1 and is described as follows:

1. At time instant k, based on the current state, the controller uses a process model to predict the process response over a finite prediction horizon, N.

2. An optimization problem is solved for the optimal sequence of control signals over the prediction horizon based on a defined cost function of the predicted process response, subject to constraints.

3. The first element in the control sequence is applied to the process at the current time instant and the rest of the control sequence is discarded.

4. Return to step 1 for the next time instant, $k + 1$.

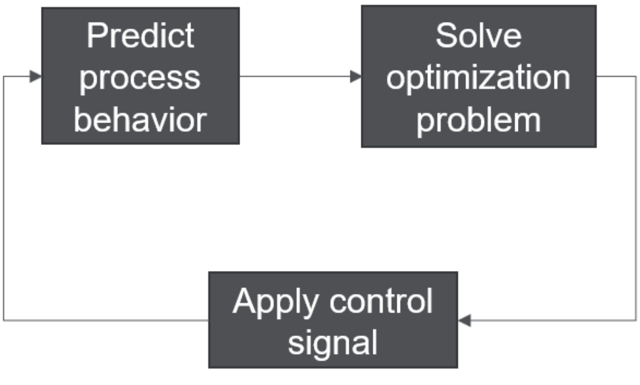


Figure 3.1: Receeding horizon idea.

3.2 System model

In order to design a controller, the system matrices needs to be discretized. This can be done by a zero order hold discretization, where it is assumed that the input signal is constant until the next sampling instant. Assuming that the plant model is continuous and described as in (2.38)-(2.39), it can be rewritten as:

$$\dot{\mathbf{x}} = A\mathbf{x} + \underbrace{\begin{bmatrix} B_u & B_w \end{bmatrix}}_{B} \underbrace{\begin{bmatrix} \mathbf{u} \\ \mathbf{w} \end{bmatrix}}_{\mathbf{v}}, \tag{3.1}$$

$$\mathbf{y} = C\mathbf{x} + D\mathbf{u}. \tag{3.2}$$

Then, the discretized matrices are:

$$A_d = e^{AT}, \tag{3.3}$$

$$B_d = A^{-1}(e^{AT} - I)B, \tag{3.4}$$

$$C_d = C, \tag{3.5}$$

$$D_d = D, \tag{3.6}$$

and the discretized plant model is:

$$x[k+1] = A_d x[k] + B_d v[k], \tag{3.7}$$

$$y[k] = C_d x[k] + D_d u[k]. \tag{3.8}$$

Equation (3.7)-(3.8) can be rewritten into the following discrete linear state equations:

$$x[k+1] = Ax[k] + B_u u[k] + B_w w[k], \tag{3.9}$$

$$y[k] = Cx[k] + Du[k], \tag{3.10}$$

where A is the state transition matrix, B_u is the control signal matrix, B_w is the disturbance input matrix, C is the output matrix, D is the feedforward matrix, $x[k[$ is the state vector, $u[k]$ is the control signal vector, $w[k]$ is the disturbance input vector and $y[k]$ is the output vector. The subscript d, used to clarify that the matrices are discrete, are from now on neglected and the matrices are assumed to be discrete.

3.3 State predictions

The state predictions are given by (3.9) and are written over the horizon as:

$$
\begin{aligned}
\hat{x}[k+1|k] &= A\hat{x}[k|k] + B_u u[k|k] + B_w w[k|k], &(3.11)\\
\hat{x}[k+2|k] &= A\hat{x}[k+1|k] + B_u u[k+1|k] + B_w w[k+1|k]\\
&= A(A\hat{x}[k|k] + B_u u[k|k] + B_w w[k|k]) + B_u u[k+1|k] + B_w w[k+1|k]\\
&= A^2\hat{x}[k|k] + AB_u u[k|k] + AB_w w[k|k] + Bu[k+1|k] + B_w w[k+1|k], &(3.12)\\
&\vdots\\
\hat{x}[k+N|k] &= A^N\hat{x}[k|k] + A^{N-1}B_u u[k|k] + A^{N-1}B_w w[k|k] + A^{N-2}Bu[k+1|k]\\
&\quad + A^{N-2}B_w w[k+1|k] + ... + B_u u[k+N-1|k] + B_w w[k+N-1|k]. &(3.13)
\end{aligned}
$$

3.4 Output predictions

From the state predictions given in Section 3.3 and by the model given in (3.10), the following output predictions can be derived:

$$
\begin{aligned}
\hat{y}[k+1|k] &= C\hat{x}[k+1|k] + Du[k+1|k]\\
&= C(A\hat{x}[k|k] + B_u u[k|k] + B_w w[k|k]) + Du[k+1|k]\\
&= CA\hat{x}[k|k] + CB_u u[k|k] + CB_w w[k|k] + Du[k+1|k], &(3.14)\\
\hat{y}[k+2|k] &= C\hat{x}[k+2|k] + Du[k+2|k]\\
&= C(A^2\hat{x}[k|k] + AB_u u[k|k] + AB_w w[k|k] + B_u u[k+1|k]\\
&\quad + B_w w[k+1|k]) + Du[k+2|k]\\
&= CA^2\hat{x}[k|k] + CAB_u u[k|k] + CAB_w w[k|k] + CB_u u[k+1|k]\\
&\quad + CB_w w[k+1|k]) + Du[k+2|k], &(3.15)\\
&\vdots\\
\hat{y}[k+N|k] &= CA^N\hat{x}[k|k] + CA^{N-1}B_u u[k|k] + CA^{N-1}B_w w[k|k]\\
&\quad + CA^{N-2}B_u u[k+1|k] + CA^{N-2}B_w w[k+1|k] + ...\\
&\quad + CB_u u[k+N-1|k] + CB_w w[k+N-1|k] + Du[k+N|k]. &(3.16)
\end{aligned}
$$

The output predictions can be rewritten into matrix form as:

$$\underbrace{\begin{bmatrix} \hat{y}[k+1|k] \\ \hat{y}[k+2|k] \\ \vdots \\ \hat{y}[k+N|k] \end{bmatrix}}_{\hat{y}} = \underbrace{\begin{bmatrix} CA \\ CA^2 \\ \vdots \\ CA^N \end{bmatrix}}_{\Phi} \hat{x}[k|k] + \underbrace{\begin{bmatrix} CB_u & D & 0 & \dots & 0 \\ CAB_u & CB_u & D & 0 & \vdots \\ \vdots & \ddots & \ddots & \ddots & 0 \\ CA^{N-1}B_u & CA^{N-2}B_u & \dots & CB_u & D \end{bmatrix}}_{\Gamma_u} \underbrace{\begin{bmatrix} u[k|k] \\ u[k+1|k] \\ \vdots \\ u[k+N|k] \end{bmatrix}}_{\hat{u}}$$
$$+ \underbrace{\begin{bmatrix} CB_w & 0 & \dots & 0 \\ CAB_w & CB_w & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ CA^{N-1}B_w & CA^{N-2}B_w & \dots & CB_w \end{bmatrix}}_{\Gamma_w} \underbrace{\begin{bmatrix} w[k|k] \\ w[k+1|k] \\ \vdots \\ w[k+N-1|k] \end{bmatrix}}_{\hat{w}} \tag{3.17}$$

Equation (3.17) can be written short as:

$$\hat{y} = \Phi x[k] + \Gamma_u \hat{u} + \Gamma_w \hat{w}. \tag{3.18}$$

In order to minimize the output of the system, a quadratic optimization problem is formulated on the form

$$\underset{\hat{u}}{min} \;\; \hat{y}^T Q \hat{y} + \hat{u}^T R \hat{u}, \tag{3.19}$$

where Q is the output penalty matrix and R the control signal penalty matrix.
By substituting (3.18) into (3.19) the following cost function is derived:

$$\hat{y}^T Q \hat{y} + \hat{u}^T R \hat{u} = \tag{3.20}$$
$$(\Phi x[k] + \Gamma_u \hat{u} + \Gamma_w \hat{w})^T Q (\Phi x[k] + \Gamma_u \hat{u} + \Gamma_w \hat{w}) + \hat{u}^T R \hat{u} = \tag{3.21}$$
$$(x[k]^T \Phi^T + \hat{u}^T \Gamma_u^T + \hat{w}^T \Gamma_w^T) Q (\Phi x[k] + \Gamma_u \hat{u} + \Gamma_w \hat{w}) + \hat{u}^T R \hat{u} = \tag{3.22}$$
$$\begin{aligned} & x[k]^T \Phi^T Q \Phi x[k] + x[k]^T \Phi^T Q \Gamma_u \hat{u} + x[k]^T \Phi^T Q \Gamma_w \hat{w} + ... \\ & + \hat{u}^T \Gamma_u^T Q \Phi x[k] + \hat{u}^T \Gamma_u^T Q \Gamma_u \hat{u} + \hat{u}^T \Gamma_u^T Q \Gamma_w \hat{w} + ... \\ & + \hat{w}^T \Gamma_w^T Q \Phi x[k] + \hat{w}^T \Gamma_w^T Q \Gamma_u \hat{u} + \hat{w}^T \Gamma_w^T Q \Gamma_w \hat{w} + \hat{u}^T R \hat{u}. \end{aligned} \tag{3.23}$$

Omitting the terms not dependent on the optimization variable $\hat{u}$ gives:

$$\begin{aligned} & x[k]^T \Phi^T Q \Gamma_u \hat{u} + \hat{u}^T \Gamma_u^T Q \Phi x[k] + \hat{u}^T \Gamma_u^T Q \Gamma_u \hat{u} + ... \\ & + \hat{u}^T \Gamma_u^T Q \Gamma_w \hat{w} + \hat{w}^T \Gamma_w^T Q \Gamma_u \hat{u} + \hat{u}^T R \hat{u} = \end{aligned} \tag{3.24}$$
$$\hat{u}^T \underbrace{(\Gamma_u^T Q \Gamma_u + R)}_{H} \hat{u} + 2 \underbrace{(x[k]^T \Phi^T Q \Gamma_u + \hat{w}^T \Gamma_w^T Q \Gamma_u)}_{f} \hat{u}. \tag{3.25}$$

Problem (3.19) can be rewritten as:

$$\underset{\hat{u}}{min} \;\; \frac{1}{2} \hat{u}^T H \hat{u} + f \hat{u}. \tag{3.26}$$

Now, the problem is formulated in such a way that a quadratic programming (QP) solver can be used. In order for the problem to be convex, the Hessian matrix (H) needs to be positive definite. The typical optimization problem to solve in an MPC problem is formulated as:

$$\underset{\hat{u}}{min}\ \frac{1}{2}\hat{u}^T H\hat{u} + f\hat{u} \tag{3.27}$$

$$\begin{aligned} \text{s.t. }\ & A\hat{u} \leq b && (3.28)\\ & lb \leq \hat{u} \leq ub, && (3.29) \end{aligned}$$

where the constraints can be formulated as linear combinations of the optimization variables and also by upper and lower bounds on the optimization variables.

3.5 Nonlinear Model Predictive control

Nonlinear MPC can be used if the vehicle model is nonlinear or if the constraints are nonlinear. The following nonlinear set of differential equations are considered for the continuous time NMPC formulation, which is described in more detail in [5]:

$$\dot{\mathbf{x}} = f(\mathbf{x}(t), \mathbf{u}(t)),\ \mathbf{x}(0) = \mathbf{x}_0 \tag{3.30}$$

$$\begin{aligned} \text{s.t. }\ & \mathbf{u}(t) \in \mathcal{U},\ \forall t \geq 0 && (3.31)\\ & \mathbf{x}(t) \in \mathcal{X},\ \forall t \geq 0, && (3.32) \end{aligned}$$

where $\mathbf{u}$(t) $\in \mathcal{U} \subseteq \mathbb{R}^m$ and $\mathbf{x}$(t) $\in \mathcal{X} \subseteq \mathbb{R}^n$ denotes the input and the state vectors, respectively. $\mathcal{U}$ stands for the set of feasible input values and $\mathcal{X}$ for the set of feasible state values.

Then the finite horizon open-loop optimal control problem can be formulated as:

$$\underset{\bar{\mathbf{u}}(\cdot)}{min}\ J(\mathbf{x}(t), \bar{\mathbf{u}}(\cdot); T_c, T_p) := \int_t^{t+T_p} F(\bar{\mathbf{x}}(\tau), \bar{\mathbf{u}}(\tau))d\tau \tag{3.33}$$

$$\begin{aligned} \text{s.t. }\ & \dot{\bar{\mathbf{x}}}(\tau) = f(\bar{\mathbf{x}}(\tau), \bar{\mathbf{u}}(\tau)),\ \bar{\mathbf{x}}(t) = x(t) && (3.34)\\ & \bar{\mathbf{u}}(\tau) \in \mathcal{U},\ \forall\tau \in [t,\ t + T_c] && (3.35)\\ & \bar{\mathbf{u}}(\tau) = \bar{\mathbf{u}}(\tau + T_c),\ \forall\tau \in [t + T_c,\ t + T_p] && (3.36)\\ & \bar{\mathbf{x}} \in \mathcal{X},\ \forall\tau \in [t,\ t + T_p], && (3.37) \end{aligned}$$

where J is some specified cost function, T_c is the control horizon, T_p prediction horizon and $T_c \leq T_p$. The function F is called the stage cost and most commonly defined as a quadratic function. The internal variables in the controller are denoted by a bar $(\bar{\mathbf{x}}, \bar{\mathbf{u}})$ to distinguish between the real system and the system model. $\bar{\mathbf{x}}(\cdot)$ is the solution of Equation 3.34 with input $\bar{\mathbf{u}}(\cdot)$: [t, T_p] $\rightarrow \mathcal{U}$ and initial condition $\mathbf{x}$(t).

3.5.1 Numerical integration methods

In order to solve for $\dot{x}$ in (3.30) numerically, an integration method must be used. Two commonly used integration methods are explicit and implicit methods, which are described in detail in [6]. An explicit method calculates the next system states based on the current states while an implicit method calculates the future system states based on both current and future states.

Explicit method:

$$x(t + \Delta t) = f(x(t)). \tag{3.38}$$

Implicit method:

$$x(t + \Delta t) = g(x(t)) + f(x(t + \Delta t)). \tag{3.39}$$

Implicit methods require an extra computation step compared to explicit methods and are usually more difficult to implement, but have a larger stability region. An explicit method requires a small step size to prevent instability while an implicit method can use a comparably larger step size while the solution still converges. An explicit integration method is not well suited to handle highly nonlinear equations since it would require a very small step size and hence increase the computation time drastically. On the other hand, implicit methods can handle highly nonlinear equations without reducing the step size to an impractically small value.

3.5.2 Shooting methods

Direct single shooting and direct multiple shooting are techniques used to discretize the time-continuous formulation and are described in [7]. A shooting method solves a boundary value problem by reducing it to the solution of an initial value problem. Direct single shooting is a sequential approach where the differential variables are considered as quantities dependent on the independent control variables. An initial value problem is solved in each iteration of the optimization algorithm to compute the derivatives with respect to control variables, the objective function and the constraints. On the other hand, direct multiple shooting is a simultaneous method. The interval over which an solution is sought is divided into smaller intervals and an initial value problem is solved in each interval. To form a solution over the whole interval additional matching conditions are imposed.

Single shooting suffers from several drawbacks. The solution of the initial value problem must exist on the interval and for a highly nonlinear or unstable ODE, the initial guess must be close to the actual but unknown solution. Sometimes this initial value may be impossible to find. In terms of numerical stability and the significant improvement in distribution of nonlinearity, multiple shooting is superior to single shooting.

4
Control design

In this chapter the control design is presented. At first, related research in the field is presented which is a followed by a description of the proposed approach for the design of the controller. Two different controllers were designed in order to answer the research questions: one full complexity controller without the limitations of a real-time implementation and one controller that can be run in real time. The latter controller thus has limitations on sampling time, prediction horizon and the complexity of the model.

4.1 Related research

A natural choice of control strategy for utilizing road preview in an optimal way is MPC, but other control strategies have been used as well. In [8], design of a Linear Quadratic Regulator (LQR) for active suspension utilizing road preview was conducted. A quarter-car model was considered and constraints on dynamic tire load variation, body accelerations and suspension working space were set and minimized in the cost function. The results showed that a fast active controller mainly tries to improve the components in the cost function that corresponds to a higher natural frequency, i.e. the tire deflection. A slow active controller cannot reduce this cost, but works well for improving ride comfort and suspension work space. In [9], MPC is applied to a half-car model with active suspension using preview of the road height profile. The results from the study showed a significant improvement in the performance for both ride comfort and handling for active suspensions compared to passive suspensions. The MPC solution was also robust against the occurrence of a significant amount of noise in the road preview. The authors claims MPC to be a superior control method for this kind of problem compared to LQR because of the ability to incorporate hard constraints in the design process. A LQR cannot explicitly incorporate hard constraints, they are minimized in the cost function instead. For active and semi-active suspensions, it is important that the limits of the suspension deflection, actuator displacements and rate of the actuator displacements are not exceeded and therefore it is important to incorporate such hard constraints in the control design, which is discussed in [10]. In that study, road preview was used together with MPC for active suspension using a reduced full-car model which is a model where the wheel dynamics are neglected. A low bandwidth actuator was considered and therefore the fast dynamics of the wheels were not taken into account, hence the reduced full-car model. The same authors implemented the controller in real car in [11] and thus managed to run the controller in real time.

Control of semi-active suspensions using MPC and a full-car model was proposed in [3]. However, preview of the road profile was not used. Instead, the authors proposed an observer approach in order to estimate the road disturbances. Ride comfort and handling were considered by the performance indices of heave acceleration through the vehicle center of gravity and roll angle, respectively. A trade-off was made between heave acceleration and roll since the two objectives are conflicting. This trade-off was chosen by a load transfer ratio which was measured by lateral accelerations. If the vehicle is exposed to lateral load transfer or roll motion, the controller aims to minimize the roll in the cost function. On the opposite, if there is low roll motion and low lateral load transfer the controller minimizes the heave acceleration in the cost function. The dissipative constraints of the semi-active dampers were transformed into input constraints as:

$$0 \leq b_{min_{ij}} \leq b_{ij}(\cdot) \leq b_{max_{ij}}, \tag{4.1}$$

where $b_{ij}(\cdot)$ is a variable damping coefficient, $i \in \{f = front,\ r = rear\}$ and $j \in \{l = left,\ r = right\}$. The other symbols correspond to those defined in Table 2.2 for the full-car model. The damping force $F_{d_{ij}}$ was defined as:

$$F_{d_{ij}} = b_{nom_{ij}}\dot{z}_{def_{ij}} + u_{ij} = b_{ij}(\cdot)\dot{z}_{def_{ij}} = \frac{b_{min_{ij}} + b_{max_{ij}}}{2}(\dot{z}_{s_{ij}} - \dot{z}_{u_{ij}}) + u_{ij}. \tag{4.2}$$

By using (4.1) and (4.2), the following input constraints were defined:

$$b_{min_{ij}}(\dot{z}_{s_{ij}} - \dot{z}_{u_{ij}}) \leq F_{d_{ij}} \leq b_{max_{ij}}(\dot{z}_{s_{ij}} - \dot{z}_{u_{ij}}),\ \text{if}\ \dot{z}_{s_{ij}} \geq \dot{z}_{u_{ij}}, \tag{4.3}$$

$$b_{max_{ij}}(\dot{z}_{s_{ij}} - \dot{z}_{u_{ij}}) \leq F_{d_{ij}} \leq b_{min_{ij}}(\dot{z}_{s_{ij}} - \dot{z}_{u_{ij}}),\ \text{if}\ \dot{z}_{s_{ij}} < \dot{z}_{u_{ij}}, \tag{4.4}$$

and because the force was expressed as:

$$\begin{aligned} F_{d_{ij}} &= \frac{b_{min_{ij}} + b_{max_{ij}}}{2}(\dot{z}_{s_{ij}} - \dot{z}_{u_{ij}}) + u_{ij} \\ &= \frac{b_{min_{ij}}(\dot{z}_{s_{ij}} - \dot{z}_{u_{ij}}) + b_{max_{ij}}(\dot{z}_{s_{ij}} - \dot{z}_{u_{ij}})}{2} + u_{ij}, \end{aligned} \tag{4.5}$$

Equation (4.3) and (4.4) could be rewritten as:

$$\frac{b_{min_{ij}} - b_{max_{ij}}}{2}(\dot{z}_{s_{ij}} - \dot{z}_{u_{ij}}) \leq u_{ij} \leq \frac{b_{max_{ij}} - b_{min_{ij}}}{2}(\dot{z}_{s_{ij}} - \dot{z}_{u_{ij}}),\ \text{if}\ \dot{z}_{s_{ij}} \geq \dot{z}_{u_{ij}}, \tag{4.6}$$

$$\frac{b_{max_{ij}} - b_{min_{ij}}}{2}(\dot{z}_{s_{ij}} - \dot{z}_{u_{ij}}) \leq u_{ij} \leq \frac{b_{min_{ij}} - b_{max_{ij}}}{2}(\dot{z}_{s_{ij}} - \dot{z}_{u_{ij}}),\ \text{if}\ \dot{z}_{s_{ij}} < \dot{z}_{u_{ij}}. \tag{4.7}$$

By using the conventional full-car model (same as the one described in Section 2.2) the constraints on the damping force exerted by the semi-active dampers depended on the direction of the velocity of the dampers. In order to solve for the optimal control sequence in the MPC formulation, either (4.6) or (4.7) needed to be chosen for each time instant. This was formulated by introducing binary variables which led to the formulation of a mixed-integer quadratic programming (MIQP) problem. The result of the study in [3] showed that the proposed controller performed very close to a MPC with preview, even though the road disturbance was estimated. However, the controller was only verified through simulations and the limitations of a real-time implementation was not considered.

Since the constraints of the semi-active dampers depends on the sign of the suspension deflection velocity, it is necessary to switch between the appropriate constraints when optimizing over the control horizon. This results in an optimization problem that is NP-hard and hence cannot be solved in polynomial time. Thus, handling the constraints of the semi-active dampers to solve the optimization problem in real time is the biggest challenge with a real-time implementation. Some simplifications or assumptions have to be made to achieve this task. In [12], both active and semi-active suspensions using MPC with road preview have been considered. Two methods to reduce the computational complexity of the optimization problem were proposed. The first method predicts damper velocities for a passive vehicle, chooses an appropriate linear damper force and approximates the relative damper velocities. Constraints for the damper force are calculated over the prediction horizon which leads to a linear optimization problem. If the resulting force is not within the bounds, maximum or minimum damping force is applied. In the second method that is proposed, no constraints on the damping force are taken into account when calculating the optimal damping force. The force is then clipped to the limits of the damper characteristics. Both methods performed as good as the reference solution, which incorporates nonlinear constraints on the damper force, for the given road disturbances.

MPC with realistic constraints for semi-active suspension are also presented in [13], where a nonlinear half-car model was employed. To overcome the problem with the large computational time for the MPC for semi-active suspension the authors utilizes a so called fast MPC method. The objective is to find the control signal as a function of the state variables. This cannot be calculated online due to the long computational time which is why the function is approximated using a set membership approach. Assuming that the state variables are bounded in some region, offline calculations are performed for a set of initial conditions. Then the only online calculation that needs to be done is a simple evaluation of the approximated function that is calculated offline. However, a model with only four states was used, i.e. two decoupled quarter-car models to represents a half-car model. For a more complex model with a large number of states, control signals and disturbance inputs the necessary number of offline calculations are much more difficult to cover.

In [14] the authors presents a MPC solution to the problem of controlling a quarter-car model with semi-active suspension. This solution is stated by the authors as the best possible solution to this problem and is used as a reference solution, assuming that future road disturbances are known. To overcome the problem with the large computational time to solve the optimization problem at each time instant in the MPC solution a rule based controller was proposed that mimics the behavior of the

MPC.

In [15] the authors proposed another method to tackle the problem with the large computational time in a real time implementation by performing the computations offline instead. Their solution uses a lidar system that measures the road profile in front of the vehicle and a Fast Fourier Transform (FFT) is applied to analyze the frequency content of the road. The road is divided into different classes of discrete events such as potholes and bumps and random roads, based on the frequency content. The MPC optimization problem is solved offline for all classes and the results are stored in lookup tables. During online operation, the control signal is recalled from the lookup table based on the current state values and the frequency content of the road. A half-car model was used in this study and the proposed method showed a better performance regarding ride comfort compared to passive suspension and other control methods such as Skyhook-control and H-infinity control.

4.2 Proposed approach

Based on previous, recently performed research, the chosen control strategy for the semi-active control problem is MPC based on a full-car model. It is an effective approach to utilize the known disturbance from the road in the design of the controller while incorporating hard constraints on the control signal. In previous research there are few studies where a full-car model has been used because of the increase in computation time a more complex vehicle model introduces. However, in order to incorporate dynamics for all relevant body movements (heave, roll and pitch), a full-car model is appropriate.

Since the force that is exerted by the semi-active damper depends on the velocity of the damper and the upper and lower constraints switch place when the velocity of the damper changes direction, it results in an optimization problem with large time complexity. In [3], the authors set up a MIQP problem by switching between the constraints depicted in (4.6) and (4.7) with the use of binary variables. This optimization problem is NP-hard and therefore not suitable for a real-time implementation that can be run sufficiently fast.

Instead of using the conventional full-car model, where the variable force from the semi-active damper is modelled as an additional force between the sprung mass and the unsprung mass, we propose a full-car model with the damping coefficient as control signal. This results in a nonlinear model where the previous assumption of small angle approximation is not enough to make the model linear since the control signal is multiplied with some of the states. The benefit of formulating the problem in such a way is that the constraints on the damping coefficient only have a lower and an upper bound and do not depend on the direction of the velocity of the damper. However, linear MPC can no longer be used for such a model which introduces the need for NMPC.

4.3 NMPC Solver

In order to make a real-time implementation feasible, an optimization problem needs to be solved online and that requires a lot of computational power. Due to increasing popularity for using MPC in systems with fast dynamics there are solvers which are tailored to MPC style optimization problems. As stated in the proposed approach, NMPC will be used in this thesis. Therefore, a solver that can handle NMPC problems needs to be used. Two frameworks that targets this type of problem are: FORCES Pro [16] and ACADO Toolkit [17]. FORCES Pro is a commercial framework for formulation of MPC and nonlinear programs (including NMPC) with its own structure exploiting solver. ACADO Toolkit is an open-source framework for formulating MPC and NMPC problems that is utilizing qpOASES [18], a QP solver, as its main low level solver. In order to make a real-time implementation feasible, the chosen solver software needs to be able to run in real-time on the hardware. ACADO Toolkit is, like qpOASES, based on C and C++ and it has support for exporting source code that can be run on the hardware and FORCES Pro is also capable of generating code for that purpose.

This thesis is done in collaboration with Volvo Car Corporation which means that an academic license is more difficult to obtain for commercial solvers (FORCES Pro). Therefore, with respect to costs and the time it would take to arrange a regular license, an open-source solution is highly preferred. Since ACADO Toolkit is open-source, able to handle NMPC and supports code generation it is the solver that is chosen in this thesis.

4.3.1 ACADO Toolkit

In order to use ACADO Toolkit, the problem should be formulated on the following form, as described in [19]:

$$\underset{\substack{x_0,\ldots,x_N\\u_0,\ldots,u_{N-1}}}{min} \sum_{k=0}^{N-1} \left|\left|h(x_k,u_k)-\tilde{y}_k\right|\right|^2_{W_k} + \left|\left|h_N(x_N)-\tilde{y}_N\right|\right|^2_{W_N} \tag{4.8}$$

$$\begin{aligned}
\text{s.t. } x_0 &= \hat{x}_0 && (4.9)\\
x_{k+1} &= F(x_k,u_k,z_k),\ for\ k=0,...,N-1 && (4.10)\\
x_k^{lb} \leq x_k &\leq x_k^{ub},\ for\ k=0,...,N && (4.11)\\
u_k^{lb} \leq u_k &\leq u_k^{ub},\ for\ k=0,...,N-1 && (4.12)\\
r_k^{lb} \leq r_k(x_k,u_k) &\leq r_k^{ub},\ for\ k=0,...,N-1 && (4.13)\\
r_N^{lb} \leq r_N(x_N) &\leq r_N^{ub}, && (4.14)
\end{aligned}$$

where the differential state is denoted by $x \in \mathbb{R}^{n_x}$, the control signal by $u \in \mathbb{R}^{n_u}$, the algebraic variable by $z \in \mathbb{R}^{n_z}$, and the current state measurement by $\hat{x}_0 \in \mathbb{R}^{n_x}$. Furthermore, $h \in \mathbb{R}^{n_y}$ and $h_N \in \mathbb{R}^{n_{y,N}}$ are reference functions and $W_k \in \mathbb{R}^{n_y \times n_y}$ and $W_N \in \mathbb{R}^{n_{y,N} \times n_{y,N}}$ are weighting matrices. Time-varying references are denoted by $\tilde{y}_k \in \mathbb{R}^{n_y}$ and $\tilde{y}_N \in \mathbb{R}^{n_y}$. Equation (4.11) and (4.12) denote upper and lower bounds on states and control signals, respectively, which also can change along the horizon. Path and point constraints are defined by (4.13) and (4.14) with $r_k \in \mathbb{R}^{n_{r,k}}$ and $r_N \in \mathbb{R}^{n_{r,N}}$. The function F in Equation (4.10) defines a discretized ODE or DAE.

ACADO Toolkit solves an NMPC problem by first discretizing the ODE or DAE by multiple or single shooting techniques and built in explicit or implicit integrators. Then it runs a real-time Gauss-Newton algorithm that returns a QP problem on the same form as (3.27). The structure exploiting QP solver using interior point method described in [20] can be used to solve the problem or alternatively, the QP can be condensed and passed to dense linear algebra solver *qpOASES* which employs an active set method.

4.4 Full complexity NMPC

In previous work, the MPC for semi-active suspension has been based on a full-car model like the one described in Section 2.2 where the control signal is the additional force exerted by the semi-active damper. This results in an optimization problem where the model is linear but the constraints are nonlinear and dependent on the deflection velocity of the damper. This thesis propose a different approach and reformulates the model to use the damping coefficient as control signal instead of the force. This results in a nonlinear plant model but the constraints on the damping coefficient is linear. In this section a Nonlinear MPC using such a model, without taking the real-time constraints into account, is described. This full complexity NMPC is referred to as the Reference NMPC from now on.

4.4.1 Vehicle model

The plant model used in the MPC is the same as the one described in Section 2.2 with the difference that the damping coefficient is controlled instead of the force that is exerted by the semi-active dampers. A visual representation of the model can be seen in Figure 4.1. The variables and constants can be found in Table 2.2, with the difference that the control signal, u_{ij}, now has the unit $N/m/s$.

Forces acting on the sprung mass, located at each corner of the sprung mass:

$$F_{fl} = k_{sfl}(z_s - t_f sin\varphi + l_l sin\theta - z_{ufl}) + u_{fl}(\dot{z}_s - t_f cos(\varphi)\dot{\varphi} + l_l cos(\theta)\dot{\theta} - \dot{z}_{ufl}), \tag{4.15}$$

$$F_{fr} = k_{sfr}(z_s - t_f sin\varphi - l_r sin\theta - z_{ufr}) + u_{fr}(\dot{z}_s - t_f cos(\varphi)\dot{\varphi} - l_r cos(\theta)\dot{\theta} - \dot{z}_{ufr}), \tag{4.16}$$

$$F_{rl} = k_{srl}(z_s + t_r sin\varphi + l_l sin\theta - z_{url}) + u_{rl}(\dot{z}_s + t_r cos(\varphi)\dot{\varphi} + l_l cos(\theta)\dot{\theta} - \dot{z}_{url}), \tag{4.17}$$

$$F_{rr} = k_{srr}(z_s + t_r sin\varphi - l_r sin\theta - z_{urr}) + u_{rr}(\dot{z}_s + t_r cos(\varphi)\dot{\varphi} - l_r cos(\theta)\dot{\theta} - \dot{z}_{urr}). \tag{4.18}$$

Tire forces acting on the unsprung masses:

$$F_{tfl} = k_{tfl}(z_{ufl} - z_{rfl}), \tag{4.19}$$
$$F_{tfr} = k_{tfr}(z_{ufr} - z_{rfr}), \tag{4.20}$$
$$F_{trl} = k_{trl}(z_{url} - z_{rrl}), \tag{4.21}$$
$$F_{trr} = k_{trr}(z_{urr} - z_{rrr}). \tag{4.22}$$

Accelerations of the sprung mass:

$$\ddot{z}_s = \frac{1}{m_s}(-F_{fl} - F_{fr} - F_{rl} - F_{rr}), \tag{4.23}$$
$$\ddot{\theta} = \frac{1}{I_x}((F_{fr} + F_{rr})l_r - (F_{fl} + F_{rl})l_l), \tag{4.24}$$
$$\ddot{\varphi} = \frac{1}{I_y}((F_{fl} + F_{fr})t_f - (F_{rl} + F_{rr})t_r). \tag{4.25}$$

Accelerations of the unsprung masses:

$$\ddot{z}_{ufl} = \frac{1}{m_{ufl}}(F_{fl} - F_{tfl}), \tag{4.26}$$
$$\ddot{z}_{ufr} = \frac{1}{m_{ufr}}(F_{fr} - F_{tfr}), \tag{4.27}$$
$$\ddot{z}_{url} = \frac{1}{m_{url}}(F_{rl} - F_{trl}), \tag{4.28}$$
$$\ddot{z}_{urr} = \frac{1}{m_{urr}}(F_{rr} - F_{trr}). \tag{4.29}$$

It is eligible to include the rate of change of the control signal in the cost function since that gives the possibility to make the controller more or less aggressive. Hence, the model is augmented so that the change of the control signals are used as control variables, i.e. $\Delta u_{fl}, \ , \Delta u_{fr}, \ , \Delta u_{rl}$ and Δu_{fl}. This is desirable because too rapid changes of the damping coefficient can introduce unwanted high frequency accelerations that the controller will not predict if the sampling time is too long.

Rate of change of the control signals are defined as:

$$\dot{u}_{fl} = \frac{\Delta u_{fl}}{\Delta t}, \tag{4.30}$$
$$\dot{u}_{fr} = \frac{\Delta u_{fr}}{\Delta t}, \tag{4.31}$$
$$\dot{u}_{rl} = \frac{\Delta u_{rl}}{\Delta t}, \tag{4.32}$$
$$\dot{u}_{rr} = \frac{\Delta u_{rr}}{\Delta t}. \tag{4.33}$$

The state vector, $\mathbf{x}$, for this model has 18 states, the 14 states of a normal full-car model and the four control signals:

$$\mathbf{x} = [z_s, \ \theta, \ \varphi, \ z_{ufl}, \ z_{ufr}, \ z_{url}, \ z_{urr}, \ \dot{z}_s, \ \dot{\theta}, \ \dot{\varphi}, \ \dot{z}_{ufl}, \ \dot{z}_{ufr}, \ \dot{z}_{url}, \ \dot{z}_{urr}, \ u_{fl}, \ u_{fr}, \ u_{rl}, \ u_{rr}]^T. \tag{4.34}$$

The control vector, $\mathbf{u}$, contains the rate of change of each control signal:

$$\mathbf{u} = [\Delta u_{fl},\ \Delta u_{fr},\ \Delta u_{rl},\ \Delta u_{rr}]^T. \tag{4.35}$$

The road height is represented in the model as a known disturbance and the disturbance vector, $\mathbf{w}$, contains the height of the road under each wheel:

$$\mathbf{w} = [z_{rfl},\ z_{rfr},\ z_{rrl},\ z_{rrr}]^T. \tag{4.36}$$

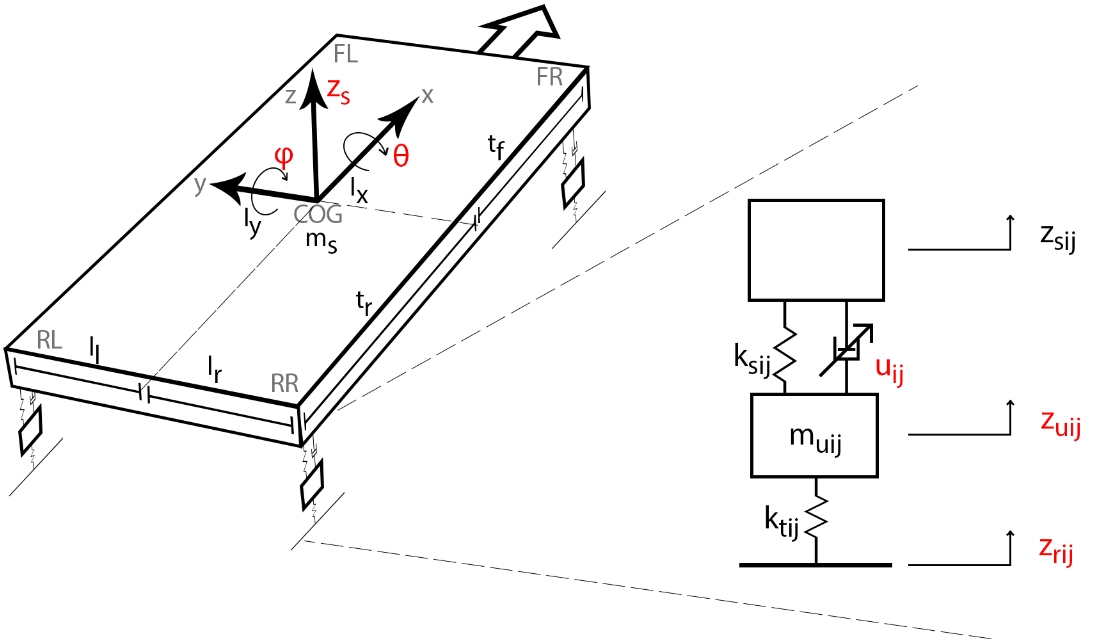


Figure 4.1: Full-car model with damping coefficient as control signal.

4.4.2 Optimization problem formulation

The optimization problem was set up to fit into the ACADO Toolkit problem formulation, described in Section 4.3.1. The objective is to minimize the square of the terms in the vector h and the vector h_N. The terms in the h vector is minimized over the entire horizon until the second to last step and the terms in the h_N vector is minimized in the last prediction step. The terms in h_N can only depend on state variables. The resulting optimization problem is:

$$\underset{\substack{x_0,\dots,x_N\\u_0,\dots,u_{N-1}}}{min} \sum_{k=0}^{N-1} \left|\left|h(x_k, u_k)\right|\right|_{W_k}^2 + \left|\left|h_N(x_N)\right|\right|_{W_N}^2 \tag{4.37}$$

$$\begin{aligned}
\text{s.t. } & x_0 = \hat{x}_0 && (4.38)\\
& x_{k+1} = F(x_k, u_k, w_k),\ for\ k = 0, ..., N-1 && (4.39)\\
& u_{fl}^{lb} \leq u_{fl} \leq u_{fl}^{ub} && (4.40)\\
& u_{fr}^{lb} \leq u_{fr} \leq u_{fr}^{ub} && (4.41)\\
& u_{rl}^{lb} \leq u_{rl} \leq u_{rl}^{ub} && (4.42)\\
& u_{rr}^{lb} \leq u_{rr} \leq u_{rr}^{ub} && (4.43)
\end{aligned}$$

$$\begin{aligned}
h &= [\ddot{z}_s,\ \ddot{\theta},\ \ddot{\varphi},\ \Delta u_{fl},\ \Delta u_{fr},\ \Delta u_{rl},\ \Delta u_{rr}]^T && (4.44)\\
hN &= [\ddot{z}_s,\ \ddot{\theta},\ \ddot{\varphi}]^T. && (4.45)
\end{aligned}$$

The cost function is set to primarily minimize the accelerations for heave, roll and pitch. However, the rate of change of the damping coefficients is also in the cost function. An upper and a lower bound are set on the damping coefficients. The damping coefficient can only take positive values.

The function $F(x_k, u_k, w_k)$ is the discretized version of the continuous model, described in Section 4.4.1. The integration and discretization of the continuous model is made with the implicit Runge-Kutta 4 integrator and multiple shooting technique built into ACADO Toolkit.

4.4.3 Tuning

The controller was tuned to primarily minimize the body accelerations. The weighting for the damping coefficient change rate was smaller than the weights for the body accelerations to prioritize lowering body accelerations, but with a high enough weight to avoid too fast changes of the damping coefficients.

The sampling time was chosen to be 10 milliseconds and the prediction horizon was 45 samples, which means that the prediction time was 0.45 seconds. For an "optimal" controller one could of course choose an even lower sampling time and a longer prediction horizon, but a trade off between good performance and convenient simulation times was made. However, further increasing the horizon did not improve performance significantly and in most cases it did not improve the performance at all.

By examination of the values that the terms in the cost function could take, it was seen that heave acceleration took values between -10 and $+10\, m/s^2$, roll and pitch acceleration were also typically in the range -10 to $+10\, rad/s^2$ and the damping coefficients could take values in the range 0 to 10000 $N/m/s$. In order to make the magnitude of the terms in the cost function similar, the damping coefficients $(u_{fl}, u_{fr}, u_{rl}, u_{rr})$ were scaled by a factor 1000.An overview of the tuning parameters can be found in Table 4.1.

The weights were set so that the heave, roll and pitch accelerations were weighted equally. The reason for this is that the values they take have similar magnitude and there is no clear reason to prioritize one over the other. Setting an equal weight on each of the accelerations means that the controller will minimize them proportionally to their current and predicted magnitude.

Table 4.1: Tuning parameter values for the Reference NMPC.

Parameter	**Value**	**Description**
Δt	0.01	Sampling time (s)
N	45	Prediction horizon (steps)
W_k	$diag(1,1,1,0.7,0.7,0.7,0.7)$	Weight matrix
W_N	$diag(1,1,1)$	Weight matrix (last step)

4.5 Real-time NMPC

The same approach as for the Reference NMPC was used when designing the real-time controller, but with the additional constraint that it needs to be able to run in real time. In order to lower the execution time, the model was reduced by removing the dynamics of the wheels and connecting the suspension directly to the road. The sampling time was also increased which gave the additional benefit that the number of prediction steps could be reduced without reducing the prediction time.

4.5.1 Vehicle model

In [10], the authors proposed a reduced full car model, which is a model where the wheel dynamics are neglected. One important benefit by using a reduced full-car model is the decreased computational complexity, since the number of states are reduced. However, neglecting the wheel dynamics gives a less accurate representation of the real system. The vehicle model with a reduced number of states and damping coefficient as control signal can be seen in Figure 4.2. The small angle approximation is also used to remove more complexity from the model. The variables and constants can be found in Table 2.2, with the difference that the control signal, u_{ij}, now has the unit $N/m/s$.

Forces acting on the sprung mass, located at each corner of the sprung mass:

$$F_{fl} = k_{sfl}(z_s - t_f\varphi + l_l\theta - z_{rfl}) + u_{fl}(\dot{z}_s - t_f\dot{\varphi} + l_l\dot{\theta} - \dot{z}_{rfl}), \tag{4.46}$$

$$F_{fr} = k_{sfr}(z_s - t_f\varphi - l_r\theta - z_{rfr}) + u_{fr}(\dot{z}_s - t_f\dot{\varphi} - l_r\dot{\theta} - \dot{z}_{rfr}), \tag{4.47}$$

$$F_{rl} = k_{srl}(z_s + t_r\varphi + l_l\theta - z_{rrl}) + u_{rl}(\dot{z}_s + t_r\dot{\varphi} + l_l\dot{\theta} - \dot{z}_{rrl}), \tag{4.48}$$

$$F_{rr} = k_{srr}(z_s + t_r\varphi - l_r\theta - z_{rrr}) + u_{rr}(\dot{z}_s + t_r\dot{\varphi} - l_r\dot{\theta} - \dot{z}_{rrr}). \tag{4.49}$$

Accelerations of the sprung mass:

$$\ddot{z}_s = \frac{1}{m_s}(-F_{fl} - F_{fr} - F_{rl} - F_{rr}), \tag{4.50}$$

$$\ddot{\theta} = \frac{1}{I_x}((F_{fr} + F_{rr})l_r - (F_{fl} + F_{rl})l_l), \tag{4.51}$$

$$\ddot{\varphi} = \frac{1}{I_y}((F_{fl} + F_{fr})t_f - (F_{rl} + F_{rr})t_r). \tag{4.52}$$

As for the Reference NMPC, it was desirable to have the rate of change of the damping coefficient in the cost function which are defined as:

$$\dot{u}_{fl} = \frac{\Delta u_{fl}}{\Delta t}, \tag{4.53}$$

$$\dot{u}_{fr} = \frac{\Delta u_{fr}}{\Delta t}, \tag{4.54}$$

$$\dot{u}_{rl} = \frac{\Delta u_{rl}}{\Delta t}, \tag{4.55}$$

$$\dot{u}_{rr} = \frac{\Delta u_{rr}}{\Delta t}. \tag{4.56}$$

This reduced model results in a state vector, $\mathbf{x}$, with 10 states instead of 18. Heave, roll, pitch and their respective velocities as well as the control signals:

$$\mathbf{x} = [z_s,\ \theta,\ \varphi,\ \dot{z}_s,\ \dot{\theta},\ \dot{\varphi},\ u_{fl},\ u_{fr},\ u_{rl},\ u_{rr}]^T. \tag{4.57}$$

The control vector, $\mathbf{u}$, contains the rate of change of the damping coefficients:

$$\mathbf{u} = [\Delta u_{fl},\ \Delta u_{fr},\ \Delta u_{rl},\ \Delta u_{rr}]^T. \tag{4.58}$$

Since the model is now also dependent on the vertical velocity of the road, the known disturbance vector, $\mathbf{w}$, contains this information as well as the road height:

$$\mathbf{w} = [z_{rfl},\ z_{rfr},\ z_{rrl},\ z_{rrr},\ \dot{z}_{rfl},\ \dot{z}_{rfr},\ \dot{z}_{rrl},\ \dot{z}_{rrr}]^T. \tag{4.59}$$

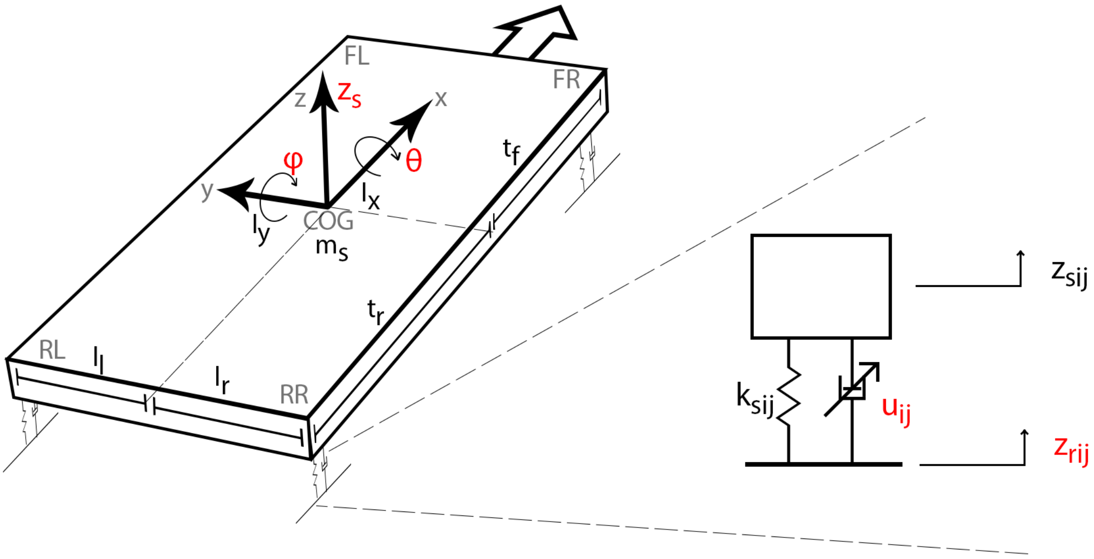


Figure 4.2: Reduced vehicle model with damping coefficient as control signal. The suspension is connected directly between the sprung mass and the road.

4.5.2 Optimization problem formulation

The optimization problem structure and the cost function is exactly the same as for the Reference NMPC. See Section 4.4.2 for more details. The optimization problem is:

$$\min_{\substack{x_0,\dots,x_N\\u_0,\dots,u_{N-1}}} \sum_{k=0}^{N-1} \left|\left|h(x_k,u_k)\right|\right|^2_{W_k} + \left|\left|h_N(x_N)\right|\right|^2_{W_N} \tag{4.60}$$

$$\text{s.t.}\ \ x_0 = \hat{x}_0 \tag{4.61}$$

$$x_{k+1} = F(x_k, u_k, w_k),\ for\ k = 0, ..., N-1 \tag{4.62}$$

$$u_{fl}^{lb} \leq u_{fl} \leq u_{fl}^{ub} \tag{4.63}$$

$$u_{fr}^{lb} \leq u_{fr} \leq u_{fr}^{ub} \tag{4.64}$$

$$u_{rl}^{lb} \leq u_{rl} \leq u_{rl}^{ub} \tag{4.65}$$

$$u_{rr}^{lb} \leq u_{rr} \leq u_{rr}^{ub} \tag{4.66}$$

$$h = [\ddot{z}_s,\ \ddot{\theta},\ \ddot{\varphi},\ \Delta u_{fl},\ \Delta u_{fr},\ \Delta u_{rl},\ \Delta u_{rr}]^T \tag{4.67}$$

$$hN = [\ddot{z}_s,\ \ddot{\theta},\ \ddot{\varphi}]^T. \tag{4.68}$$

4.5.3 Tuning

The goal of the tuning of the Real-time NMPC was the same as for the Reference NMPC, which was to minimize body accelerations. In order to meet the real-time constraints, the sampling time was increased since this enabled shortening of the prediction horizon without reducing the prediction time. It also gave the controller more time to compute the next control signal before the next sampling instant. The sampling time for the Real-time NMPC was set to 30 milliseconds and the number of prediction steps was set to 20. This meant that the prediction time was 0.6 seconds. An overview of the tuning parameters can be found in Table 4.2.

Table 4.2: Tuning parameter values for the Real-time NMPC.

Parameter	**Value**	**Description**
Δt	0.03	Sampling time (s)
N	20	Prediction horizon (steps)
W_k	$diag(1, 1, 1, 0.2, 0.2, 0.2, 0.2)$	Weight matrix
W_N	$diag(1, 1, 1)$	Weight matrix (last step)

4.6 Real-time implementation

One of the main goals of the thesis was to make the controller run in real time. To do that, the controller had to be run on a piece of hardware running a real-time operating system. The hardware that was used was a MicroAutobox II DS1401 [21] from dSpace with a 900 MHz processor. The dSpace compiler supports plain C and C++ code but it is also possible to generate code from a Simulink model by using Simulink coder and the RTI1401 code generation tool from dSpace. The RTI1401 tool does however only support C code and not C++ code.

The controller was implemented using ACADO Toolkit, which automatically generates a C code implementation of the MPC solver. To be able to interface the solver with the rest of the functionality, which was made in Simulink, the solver was compiled into an S-function that was inserted into the model. When doing the implementation in this way, the RTI1401 tool could be used to generate C code for the entire model and upload it to the Autobox. The underlying QP solver that was used with ACADO Toolkit was qpOASES. However, the qpOASES solver implementation is done in C++ and the RTI1401 tool only supports C code. Therefore, a plain C code translation of the qpOASES solver, *qpOASES_e* [22], was used instead.

5
Simulation testing and results

This chapter describes how the controller performance was tested in simulations against a realistic vehicle model. Evaluation metrics are introduced and a verification of the vehicle models is done. Test scenarios to test the performance of the controllers are described as well as the procedure how to test the real-time performance. Finally, the results of the simulations are presented.

5.1 Realistic vehicle model

Simulations to test the controller were performed as a co-simulation with Simulink and CarMaker, [23], which is a virtual testing environment for automobiles and light-duty vehicles. The controllers were evaluated against a realistic vehicle model, validated on a Volvo S90. This vehicle model considers all dynamics that affects the movement of the car body while the controllers are based upon much simpler models. The solver used in the continuous model for the co-simulation with CarMaker was ODE1, which uses the Euler method, with a fixed step size of 0.001 seconds. When the controllers were run, the damper forces in the model were replaced with external damper forces, calculated from the control signals and the suspension deflection velocities from CarMaker. The forces were calculated using the ideal damper model, $F = b \cdot v$. The reference vehicle to compare the controllers with was a car with a passive suspension with nonlinear characteristics for the springs and the dampers.

5.2 Evaluation metrics

To evaluate the performance of the controllers in simulation, the primary parameters to consider were the body accelerations since low body accelerations are considered as good ride comfort. To evaluate the behaviour between the different controllers as well as comparisons with a passive car, the Root Mean Squared (RMS) error of the body accelerations were used. It is defined as:

$$x_{rms} = \sqrt{\frac{1}{n}(x_1^2 + x_2^2 + \dots x_n^2)}. \tag{5.1}$$

The FFT of the body accelerations was also calculated in order to analyze the frequency content of the accelerations. The controllers were evaluated by a visual evaluation of how the frequency spectrum differed between the controllers and the passive car.

5.3 Model verification

This section aims to verify the vehicle models by comparing them to the realistic vehicle model in CarMaker. The state variables from the full-car model and the reduced full-car model in Simulink were compared to the state variables from the CarMaker model. The road profile used in the test consists of a bump that is rotated with an angle of 45 degrees which triggers a combined motion of heave, roll and pitch. The front left wheel hits the bump first, then the front right wheel followed by the rear wheels in the same order. The passive CarMaker model used in this test had passive suspension with a fixed damping coefficient for each damper and was using the same ideal damper model as the two full-car models. The full-car model that was verified is described in Section 4.4.1 and the reduced full-car model is described in Section 4.5.1. The road profile used in the test is shown in Figure 5.1.

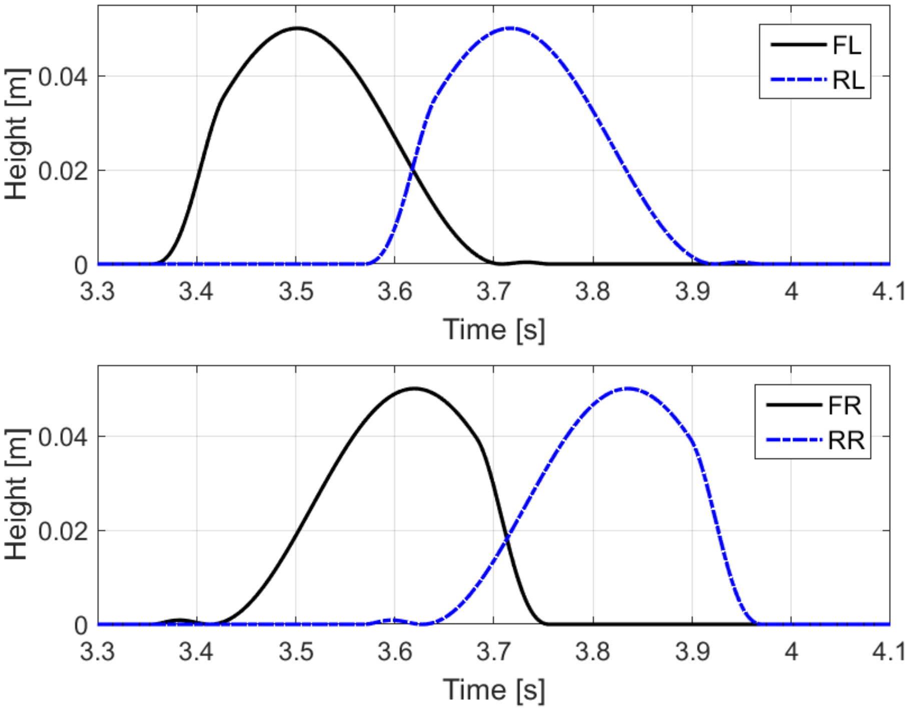
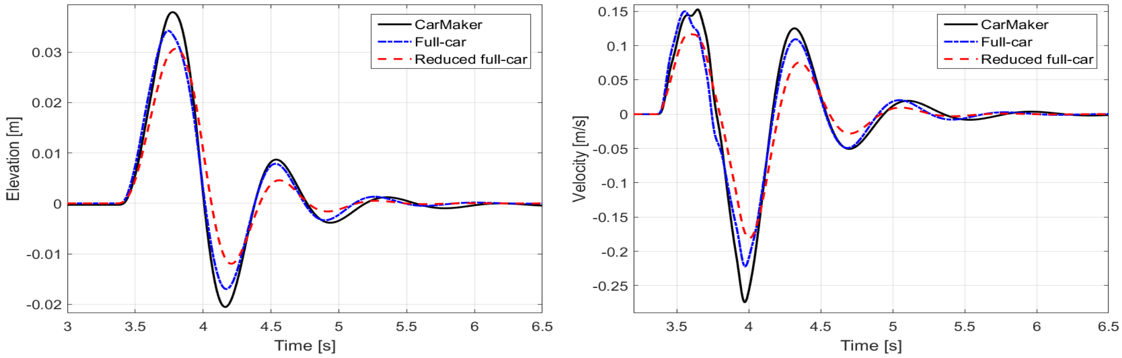


Figure 5.1: Road profile used to verify the full-car model and the reduced full-car model. The black curves represents the road profile under the front left wheel and the front right wheel, respectively. The blue curves represents the road profile under the rear left and rear right wheels.

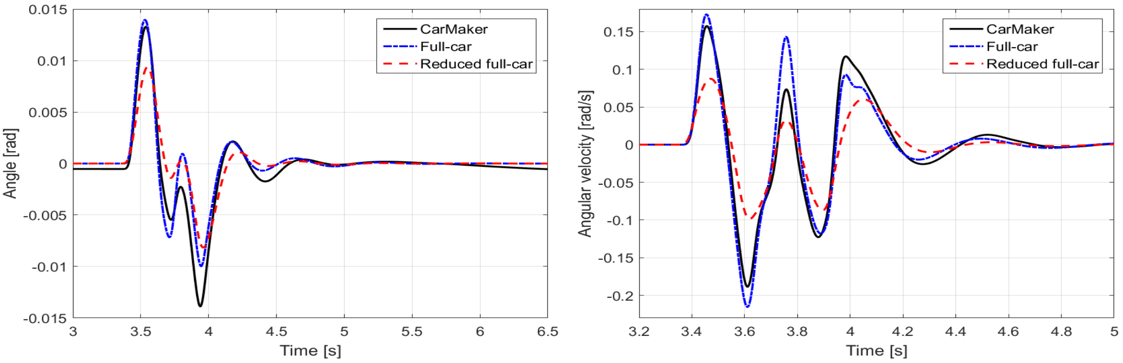
As can be seen in Figure 5.2, 5.3, 5.4 and 5.5 the behaviour of the vehicle when passing over a rotated bump is very similar when comparing the full-car model to the reduced full-car model. The full-car model and the reduced full-car model are simplified models which only consider vertical dynamics compared to the vehicle model in CarMaker which describes the full dynamics of a real car. Which is why some differences between the models are inevitable. However, the dynamics are very similar when assessing the shapes of the curves and the main difference is the amplitude of the curves.



(a) Heave of vehicle body.

(b) Heave velocity of vehicle body.

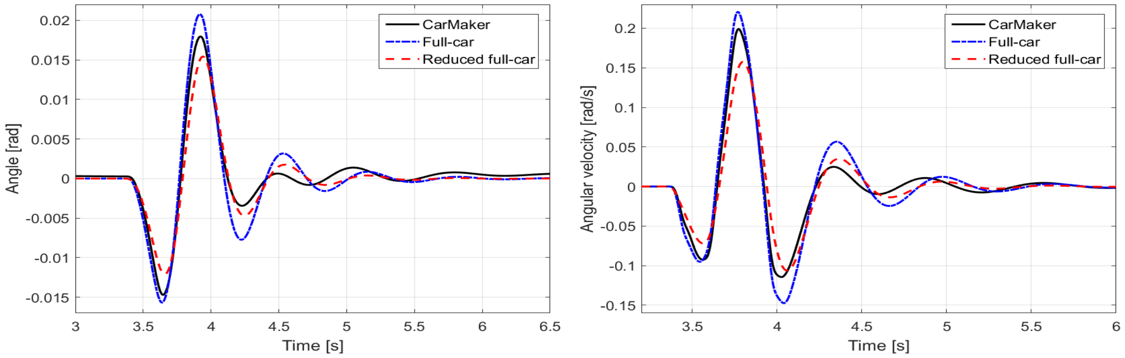
Figure 5.2: Comparison of vehicle behaviour between full-car model, reduced full-car model and vehicle model in CarMaker with respect to heave and heave velocity of the vehicle body.



(a) Roll of vehicle body.

(b) Roll velocity of vehicle body.

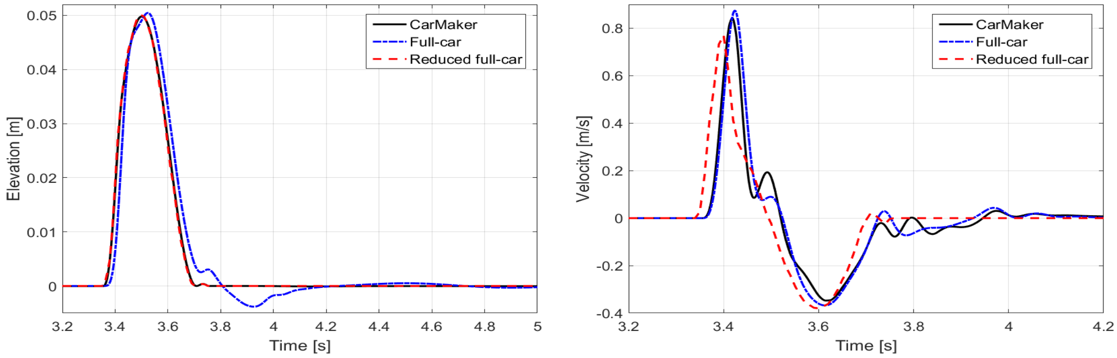
Figure 5.3: Comparison of vehicle behaviour between full-car model, reduced full-car model and vehicle model in CarMaker with respect to roll and roll velocity of the vehicle body.



(a) Pitch of vehicle body.

(b) Pitch velocity of vehicle body.

Figure 5.4: Comparison of vehicle behaviour between full-car model, reduced full-car model and vehicle model in CarMaker with respect to pitch and pitch velocity of the vehicle body.



(a) Heave of front left wheel.

(b) Heave velocity of front left wheel.

Figure 5.5: Comparison of vehicle behaviour between full-car model, reduced full-car model and vehicle model in CarMaker with respect to heave and heave velocity of the front left wheel. Since there are no wheels in the reduced full-car model, the spring and the damper are connected directly to the road and the red curves correspond to the road height and the rate of change of the road height.

5.4 Testing controller performance

In order to test the performance of the controllers, four difference test scenarios were created with the aim to test different combinations of heave, roll and pitch of the car.

5.4.1 Scenario 1: bumps

This test case aims to test the controllers ability to handle irregularities in the road in the shape of small bumps and a hole and can be seen in Figure 5.6. Since the two front wheels hit the bumps simultaneously, the resulting motion of the vehicle body are mainly pitch and heave. Some roll is also induced because the COG of the car is not perfectly centered. The first two bumps are placed close to each other and then comes a third larger bump followed by a hole.

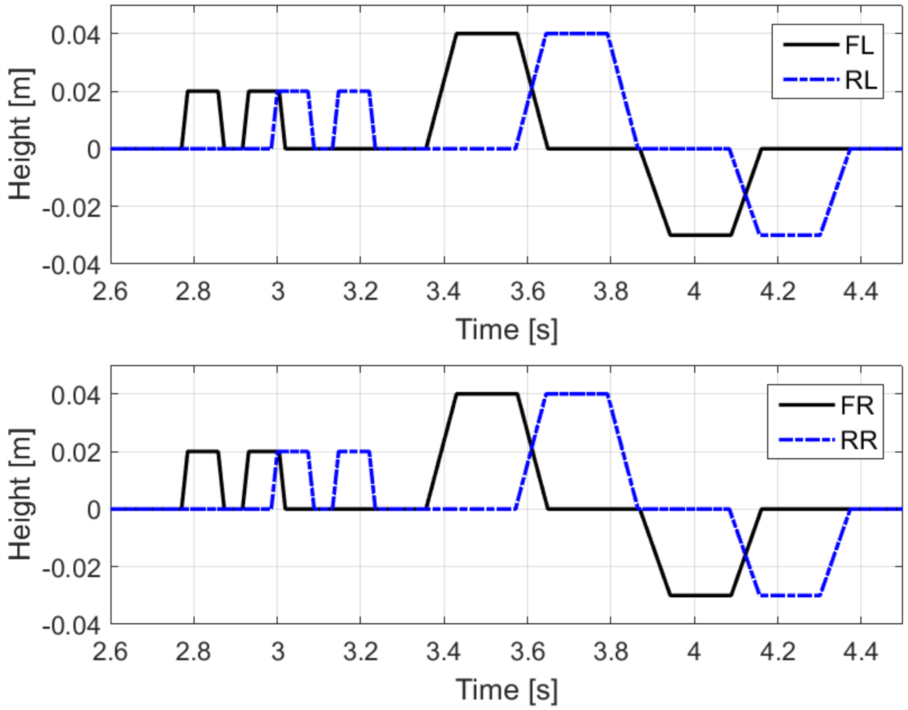


Figure 5.6: Road profile for test case which consists of two small bumps following one larger bump and a hole. The car hit the bumps with the front wheels at the same time. Note that the road profile is zoomed in on the area of interest. Before and after the described disturbance in the road profile the height is zero.

Figure 5.7 shows the resulting body accelerations (heave-, roll- and pitch) from simulation over the road profile. As can be seen, the amplitude of the peaks in the accelerations are decreased by the controllers over almost the whole test case compared to the passive car. The Reference NMPC and the Real-time NMPC perform very similar for this test scenario with respect to how they lower the body accelerations. The RMS-error of the body accelerations is shown in Table 5.1. It can be seen that the Reference NMPC has the lowest RMS-error and the Real-time controller has a slightly higher RMS-error. However, both the controllers have a significantly lower RMS-error than the passive car.

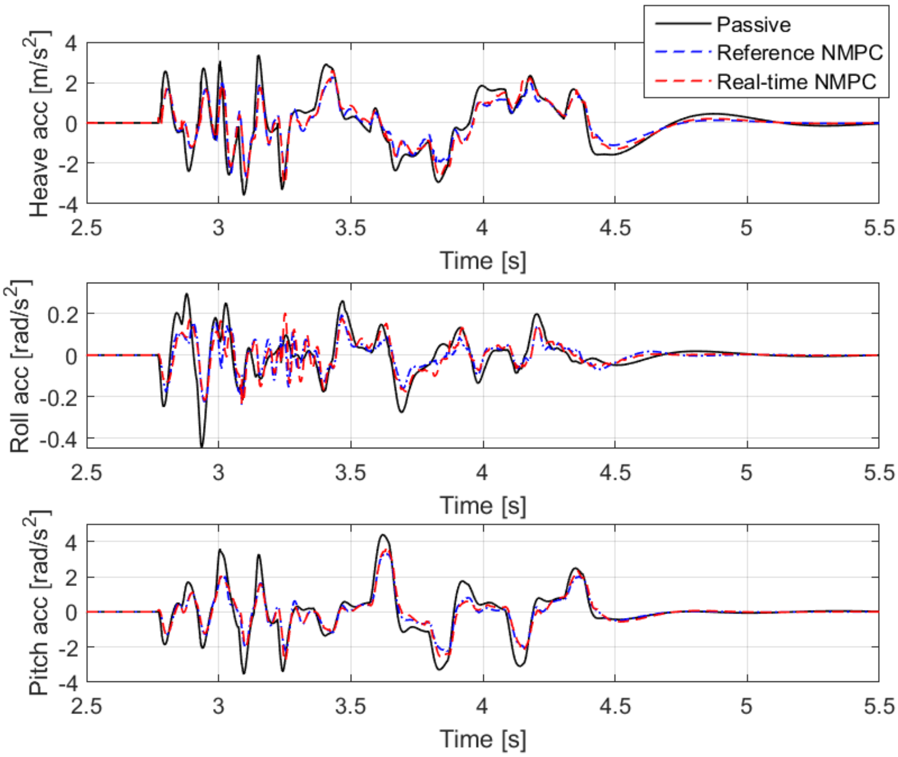


Figure 5.7: Heave, roll and pitch accelerations of the vehicle body.

Table 5.1: RMS errors for the body accelerations during Scenario 1.

	Heave acc	***Roll acc***	***Pitch acc***
Passive:	0.71	0.054	0.73
Reference NMPC:	0.5	0.035	0.5
Real-time NMPC:	0.56	0.039	0.53

The frequency content of the body accelerations is shown in Figure 5.8. As can be seen, the amplitude of most frequencies are lowered for both the Reference NMPC and the Real-time NMPC compared to the passive car.

Figure 5.9 shows the control signal at the front left wheel and the control signal at the rear right wheel for the Reference NMPC and the Real-time NMPC. The control signals are very similar which is expected since the two front wheels hits the bumps and the hole simultaneously and the same for the rear wheels. The damping coefficient is decreased when the wheel hit the bump in order to let the wheel travel with low damping. Then the damping coefficient is increased in order to dampen out the oscillations followed after hitting the bump. This behaviour is as expected and the control signals stays between the specified bounds.

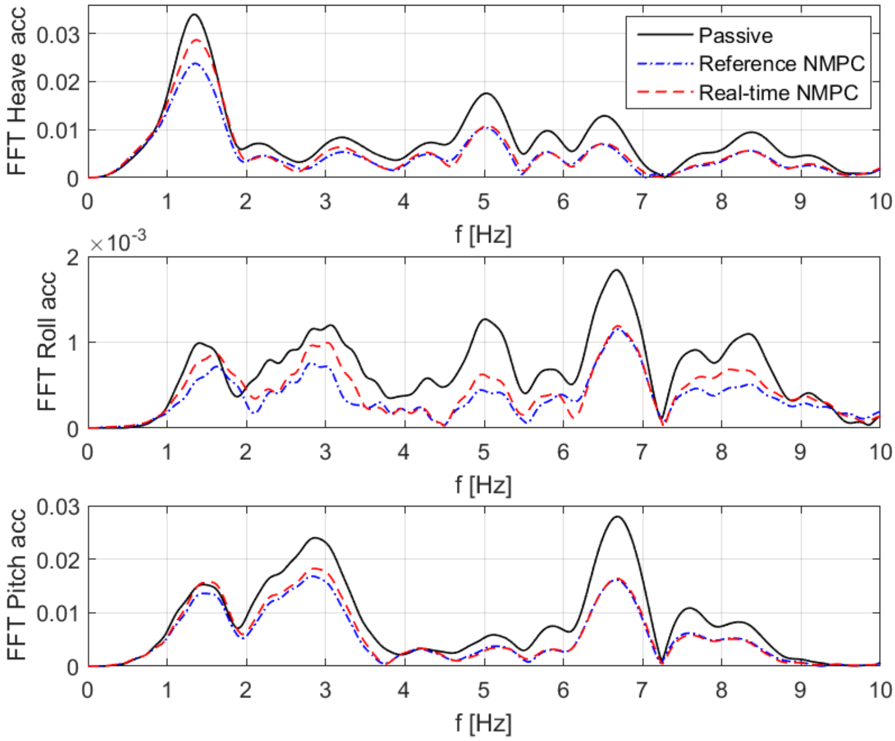


Figure 5.8: Frequency spectrum of heave, roll and pitch accelerations of the vehicle body.

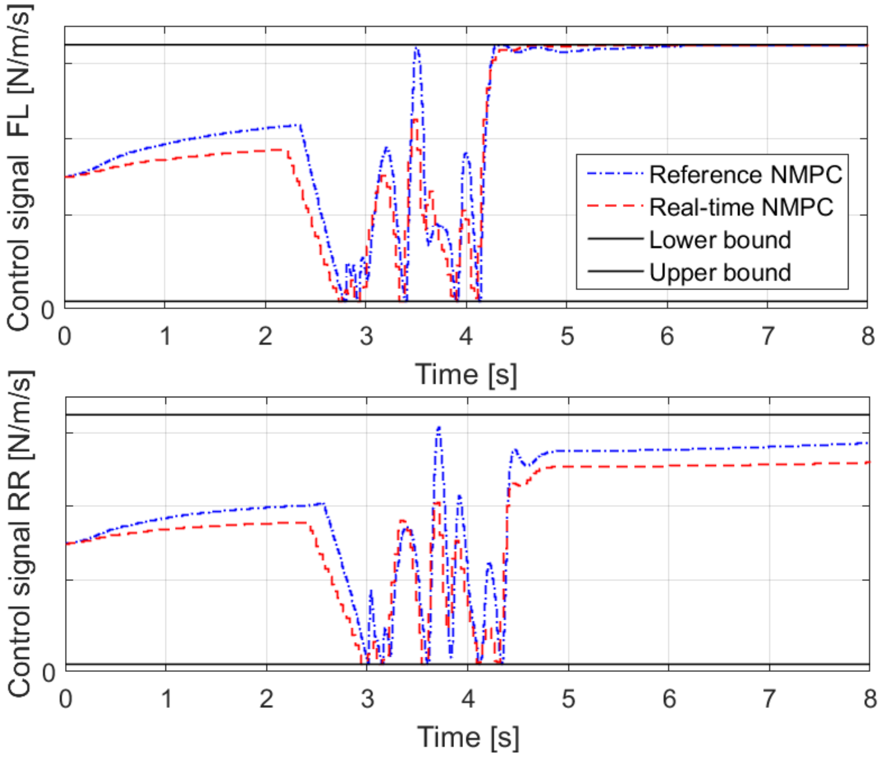


Figure 5.9: Control signals at the front left wheel and the rear right wheel for the Real-time NMPC and the Reference NMPC.

5.4.2 Scenario 2: one side disturbances

In this scenario, shown in Figure 5.10, the car is exposed to disturbances on the right side of the car in the shape of one hole, one small bump and another larger hole. Note that the right hand wheels are exposed to this road profile while the road profile on the left hand side is a flat surface. The resulting vehicle body motion is a combined roll and pitch motion with an additional heave motion. This scenario is interesting because it induces both roll and pitch motions which means that a full-car model is required to accurately predict the process response in the MPC. A smaller model, like the half-car model, would not be sufficient for predicting the process response in this scenario.

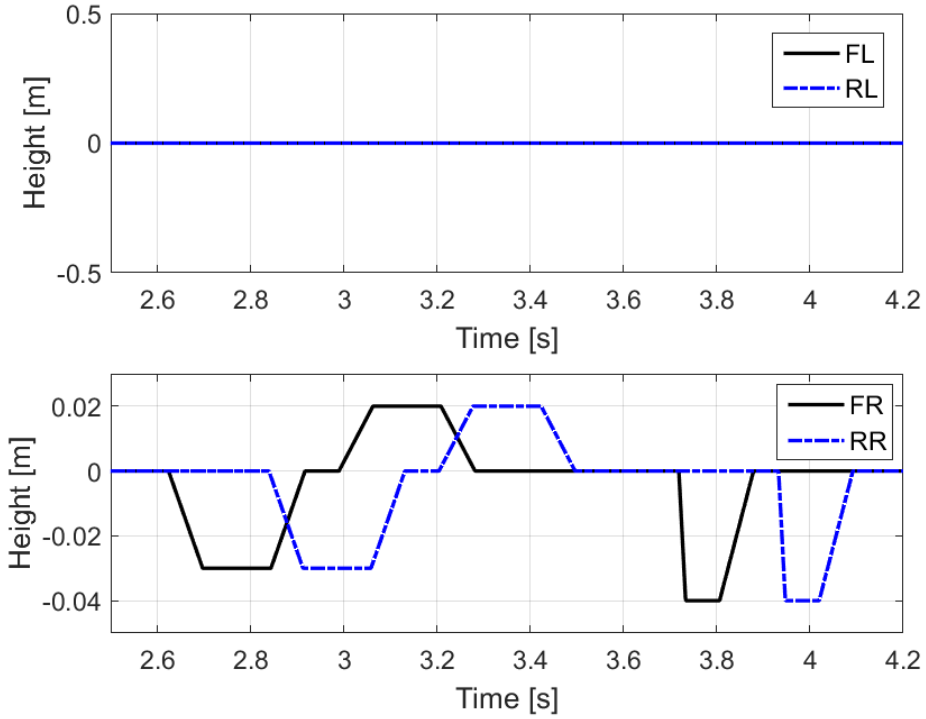


Figure 5.10: Road profile for test case with disturbance on only the right hand side in terms of two holes and one small bump. Only the wheels on the right side of the car hits the holes and the bump while the road profile under the tires on the left side is flat. Note that the road profile is zoomed in on the area of interest. Before and after the described disturbance in the road profile the height is zero.

The resulting body accelerations for this test scenario are shown in Figure 5.11 and the RMS-errors of the body accelerations are presented in Table 5.2. Note that the Reference NMPC has the lowest RMS-error for all body accelerations, the Real-time slightly higher RMS-error than the Reference NMPC and the passive car has the highest RMS-errors for all body accelerations.

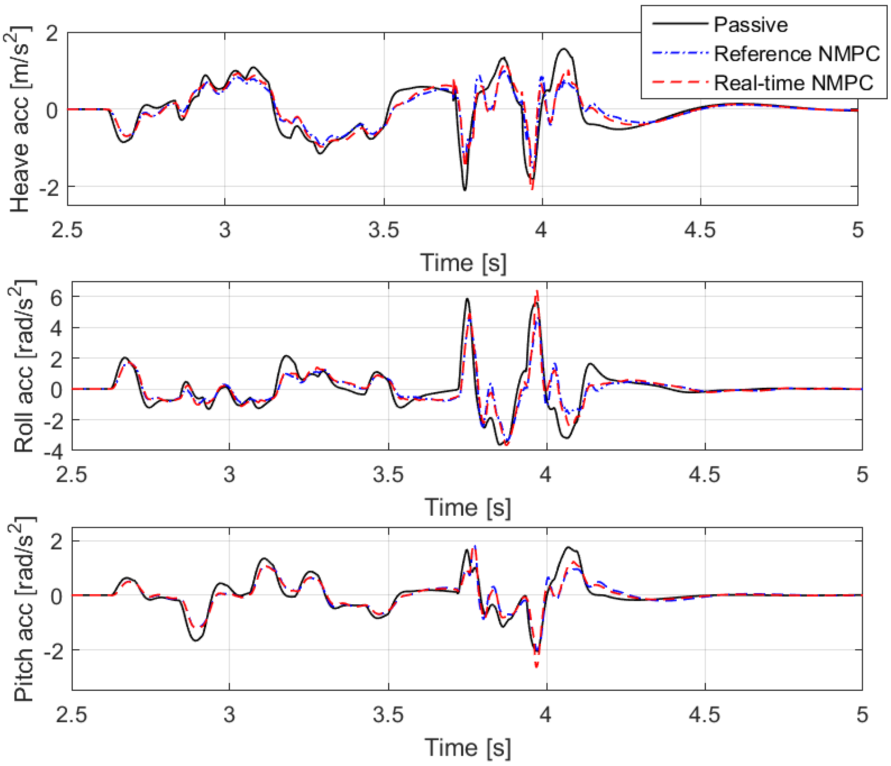


Figure 5.11: Heave, roll and pitch accelerations of the vehicle body.

Table 5.2: RMS errors for the body accelerations during Scenario 2.

	Heave acc	***Roll acc***	***Pitch acc***
Passive:	0.33	0.72	0.32
Reference NMPC:	0.24	0.53	0.25
Real-time NMPC:	0.26	0.58	0.26

The frequency content of the body accelerations is presented in Figure 5.12. Note that especially the amplitude of frequencies over around 4 Hz are reduced for the Reference NMPC and the Real-time NMPC compared to the passive car. Those frequencies corresponds to the secondary ride, which is described in Section 2.5 and are frequencies that are less comfortable.

The control signals for the Reference NMPC and the Real-time NMPC at the front right wheel and at the rear left wheel are shown in Figure 5.13. Since the road profile only consist of holes on the right side of the car, the control signals between the left and the right side differs. The damping coefficient for the front right damper changes between low and high values according to when the wheels hit the obstacle. On the other hand, the control signal (damping coefficient) for a wheel which travels on the flat surface is mainly high with a decrease when the car hits the larger hole.

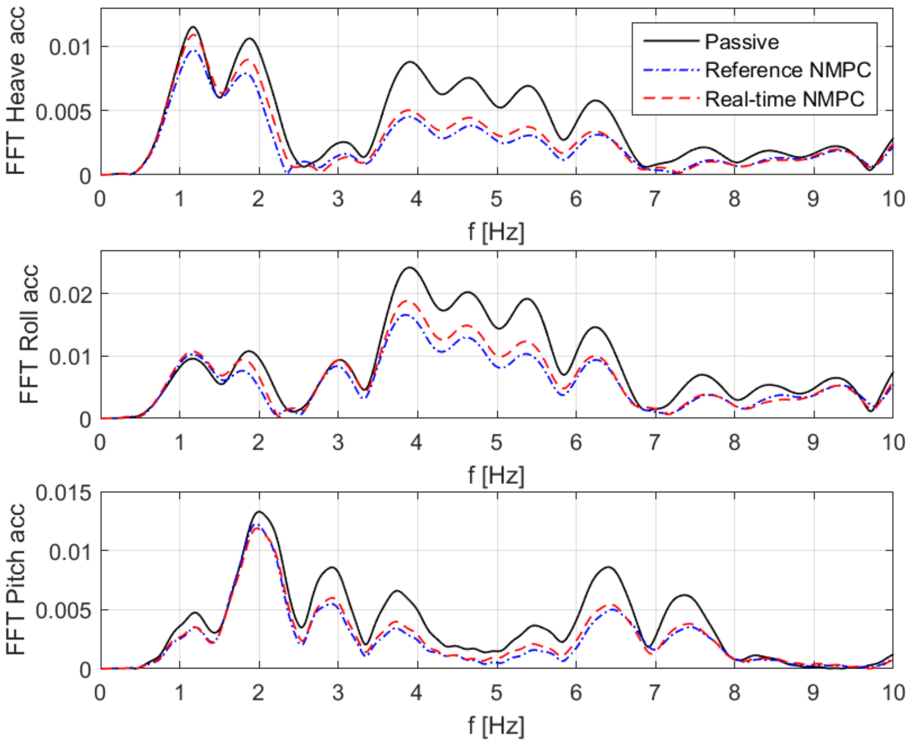


Figure 5.12: Frequency spectrum of heave, roll and pitch accelerations of the vehicle body.

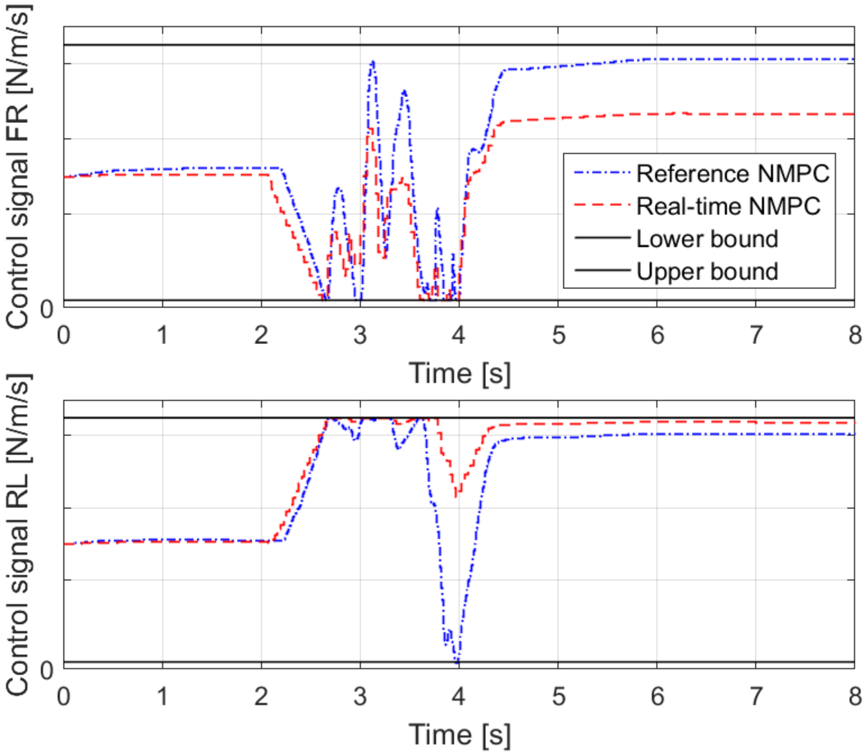


Figure 5.13: Control signals for the Real-time NMPC and the Reference NMPC.

5.4.3 Scenario 3: long dip

This test case consists of a long dip and the road profile can be seen in Figure 5.14. This scenario aims to test the behaviour of the controller when the road profile is changing significantly in height during the test case. A small bump is introduced after approximately 3.1 seconds when the car is driving downhill and small hole at around 5.2 seconds when the car is driving uphill. This is done to investigate how the controller handles disturbances while driving uphill or downhill.

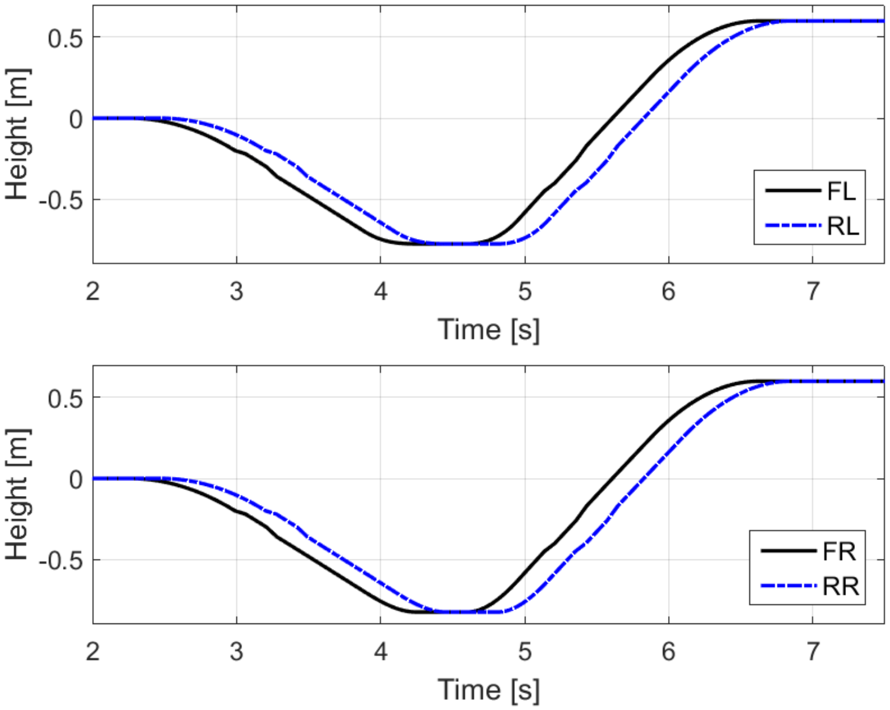


Figure 5.14: Road profile for test case with a long dip with a small bump on the way downhill and a small hole on the way uphill. Note that the road profile is zoomed in on the area of interest. Before and after the described disturbance in the road profile the height is zero.

Figure 5.15 shows the heave-, roll- and pitch accelerations and the RMS-errors of the body accelerations are presented in Table 5.3. The body accelerations are reduced for the Reference NMPC and the Real-time NMPC compared to the passive car. The reference controller performs slightly better than the real-time controller, except for roll acceleration where they perform just as good.

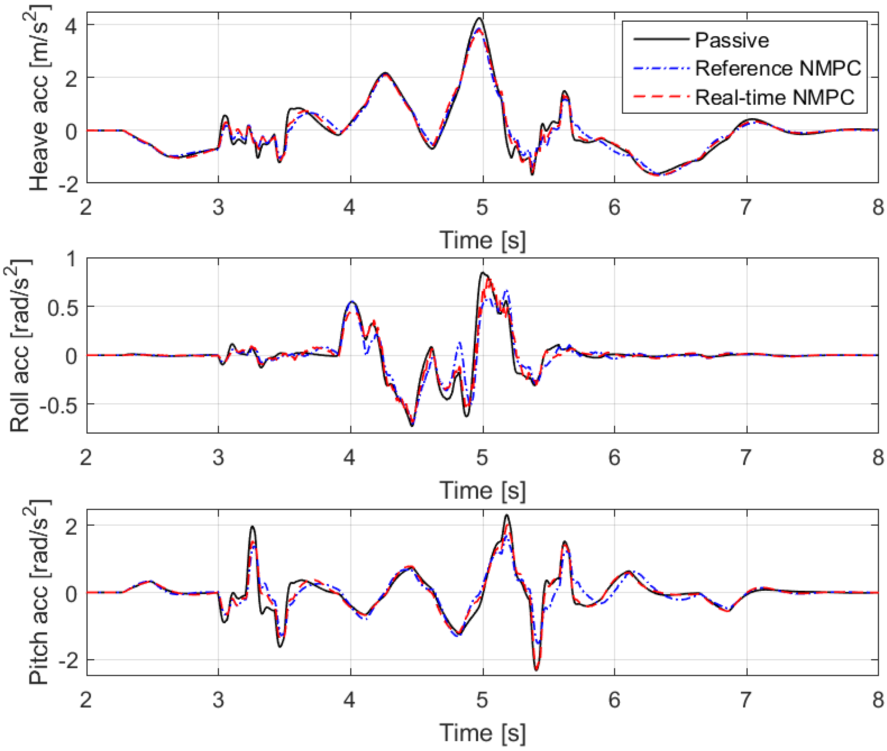


Figure 5.15: Heave, roll and pitch accelerations of the vehicle body.

Table 5.3: RMS errors for the body accelerations during Scenario 3.

	Heave acc	***Roll acc***	***Pitch acc***
Passive:	0.9	0.18	0.49
Reference NMPC:	0.83	0.16	0.4
Real-time NMPC:	0.85	0.16	0.45

The frequency spectrum of the body accelerations are shown in Figure 5.16. Since the accelerations are relatively similar for the two controllers and the passive car for this test case the amplitude of the different frequencies are similar. However, the Real-time NMPC reduces the amplitude slightly compared to the passive car and the same for the Reference NMPC compared to the Real-time NMPC.

The control signals at the front right wheel and at the rear left wheel are shown in Figure 5.17. The control signals differs quite significantly between the Reference NMPC and the Real-time NMPC for this scenario. However, as previously stated, the body accelerations are very similar between the two controllers as well as the RMS-errors of the body accelerations.

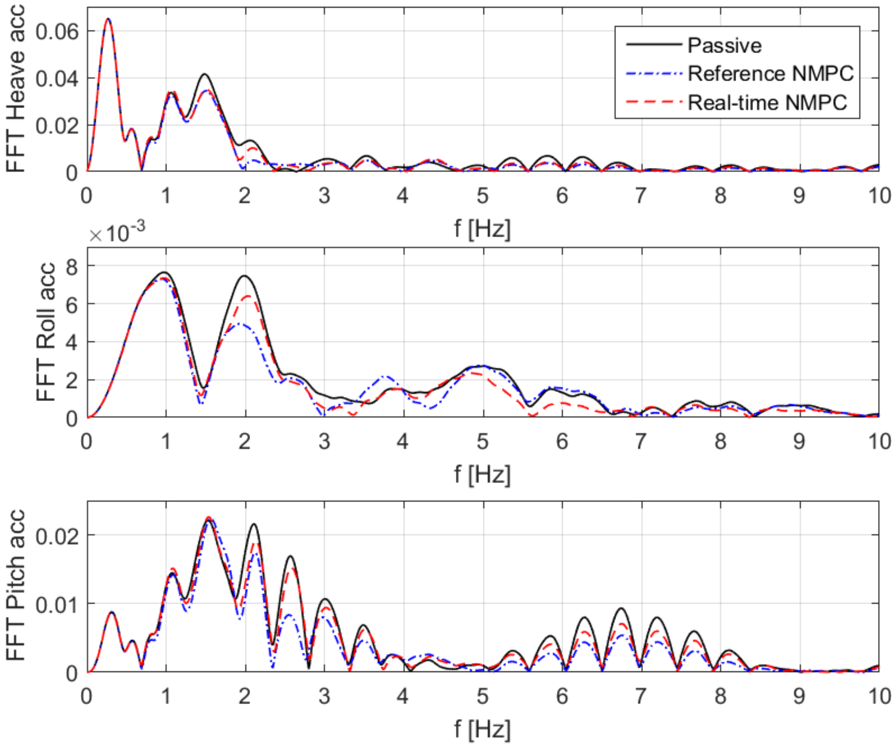


Figure 5.16: Frequency spectrum of heave, roll and pitch accelerations of the vehicle body.

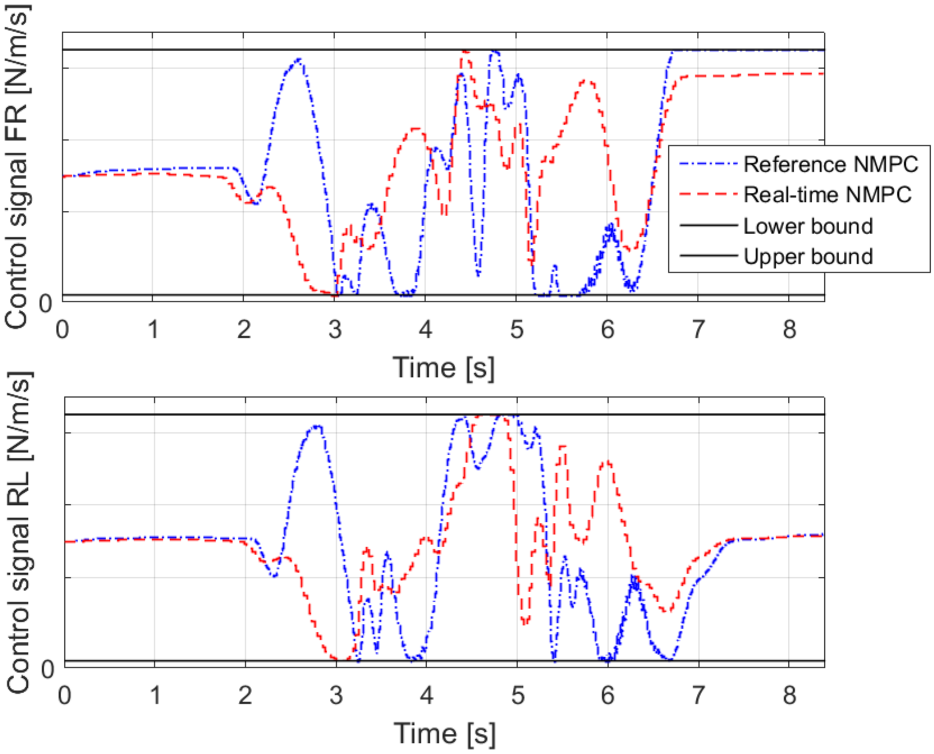


Figure 5.17: Control signals for the Real-time NMPC and the Reference NMPC.

5.4.4 Scenario 4: waves

A road profile is commonly defined as combination of different sine shaped waves with different amplitudes and frequencies. This test case consist of such waves and is shown in Figure 5.18 and aims to test how well the controller handles such a road.

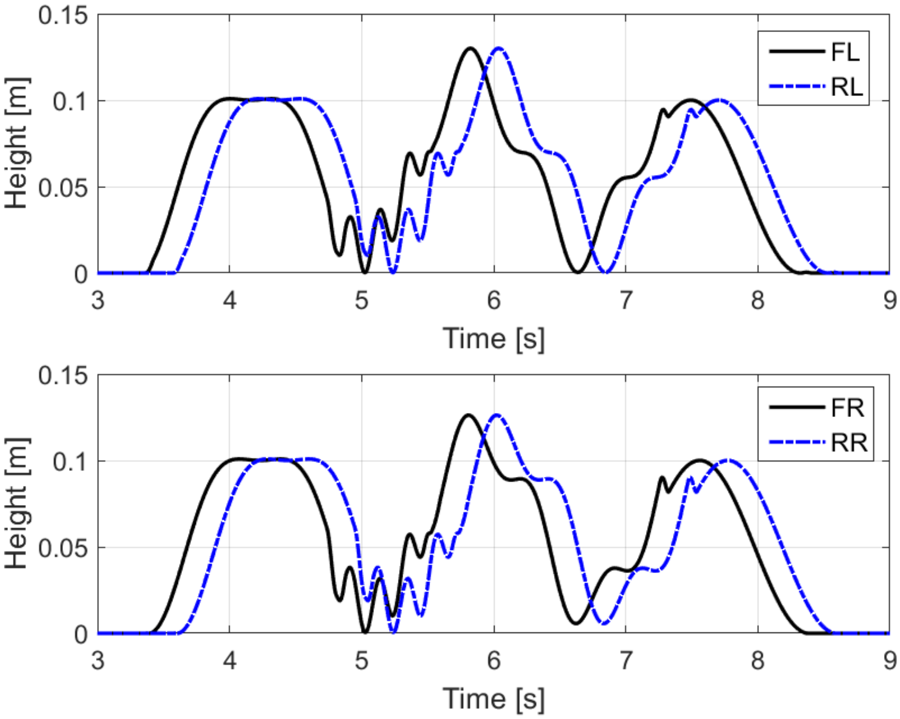


Figure 5.18: Road profile for test case with wave shaped road with different amplitudes and frequencies. Note that the road profile is zoomed in on the area of interest. Before and after the described disturbance in the road profile the height is zero.

Figure 5.19 shows the resulting body accelerations for simulation over such a road profile. It is clear from the figure that both the Reference NMPC and the Real-time NMPC minimize the body accelerations compared to a passive car, especially for waves with a higher frequency. Also when looking at the RMS-errors of the body accelerations presented in Table 5.4, the Reference NMPC has the lowest RMS-errors, the Real-time NMPC slightly higher RMS-erros and the passive car the by far highest RMS-errors.

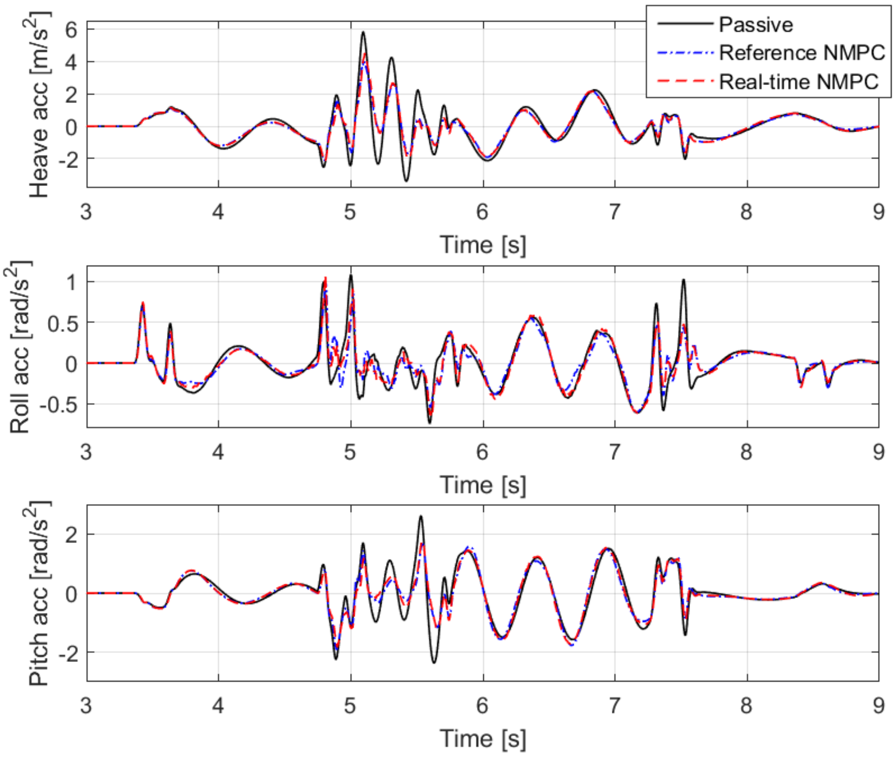


Figure 5.19: Heave, roll and pitch accelerations of the vehicle body.

Table 5.4: RMS errors for the body accelerations during Scenario 4.

	Heave acc	***Roll acc***	***Pitch acc***
Passive:	0.92	0.22	0.58
Reference NMPC:	0.7	0.17	0.51
Real-time NMPC:	0.73	0.19	0.52

The frequency spectrum of the body accelerations is presented in Figure 5.20. The amplitude of frequencies higher than approximately 3-4 Hz are reduced, which improves secondary ride. This behaviour can also be seen in Figure 5.19 where the two controllers have lower peaks in the body accelerations when the car hits wave shaped road disturbances of a higher frequency.

Figure 5.21 shows the control signals for the front left wheel and the rear right wheel. For this test case, the control signals for the Reference NMPC and the Real-time NMPC are similar in shape with a small difference in amplitude.

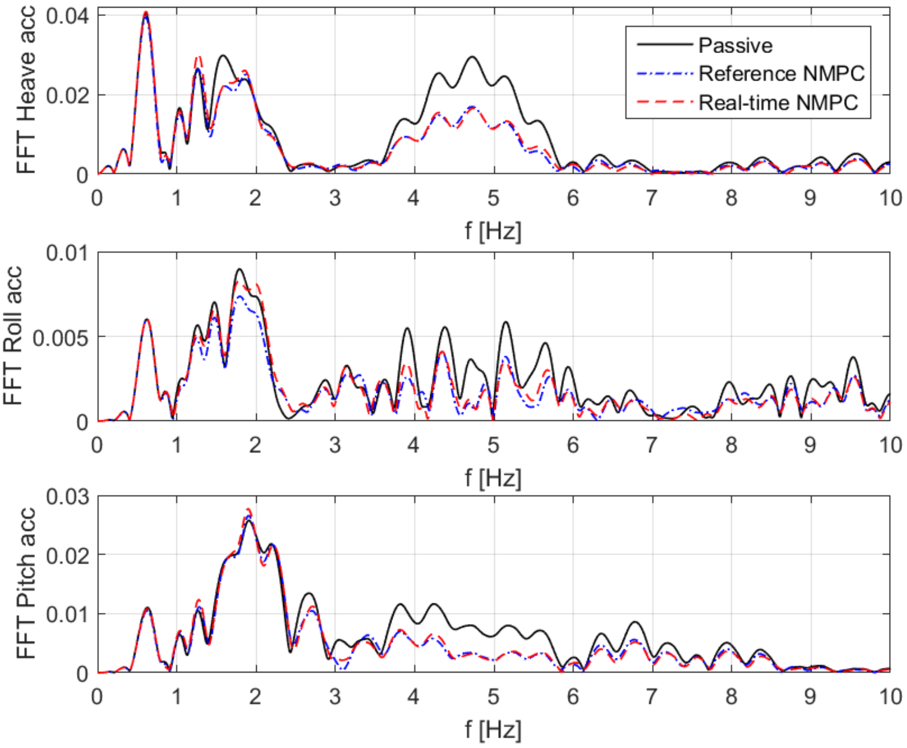


Figure 5.20: Frequency spectrum of heave, roll and pitch accelerations of the vehicle body.

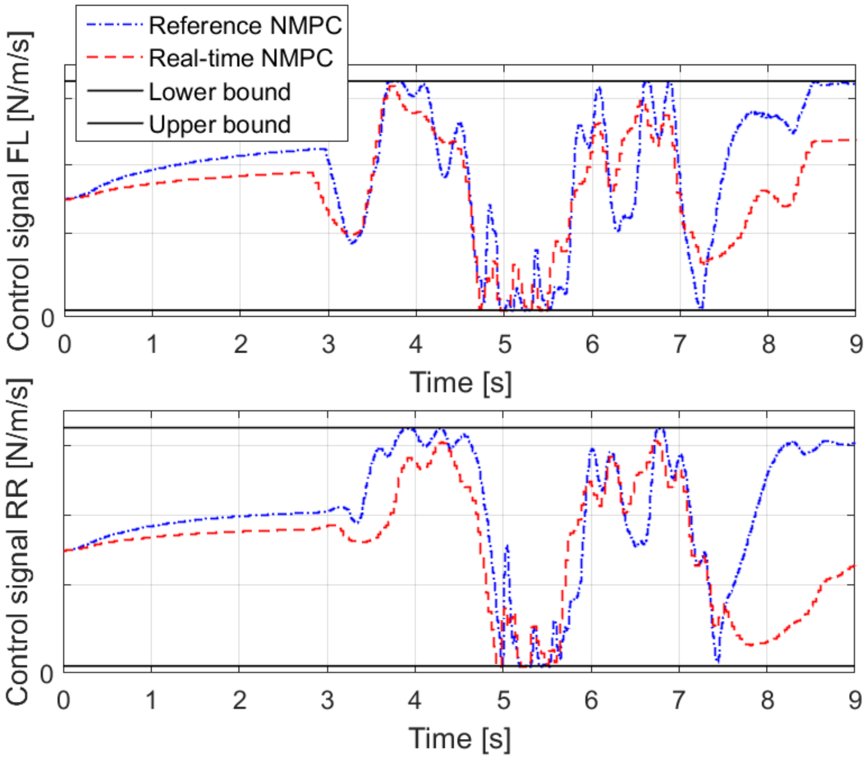


Figure 5.21: Control signals for the Real-time NMPC and the Reference NMPC.

5.5 Measuring execution time

Being able to run the controller in real time means that the entire controller algorithm should take less than one sampling interval to execute on the specific real-time platform. For example if the sampling time is 0.03 seconds, the execution time of the controller on the hardware needs to be less than 0.03 seconds if it should run in real-time.

In order to check the real-time performance of the controller, the execution time of the controller is measured by running it on the MicroAutobox. On the MicroAutobox, the execution time can be monitored via Control Desk [24] from dSpace as the turnaround time of the task that is executing. If no other code besides the controller is running, the turnaround time of the task is the same as the execution time of the controller.

The difficulty in measuring the execution time of an MPC is that the time it takes to solve the optimization problem may vary depending on the current inputs. This means that the execution time has to be measured for many different inputs in order to gain an understanding for what the execution time actually is. Therefore, the controller was run for an entire test scenario and the average and maximum execution time were recorded. This test was carried through by first simulating the scenario with the closed-loop controller in CarMaker and record all the inputs to the controller. The recorded inputs were then played back to the controller running on the MicroAutobox. This made sure that the inputs were realistic and not random. It also removed the need for simulating the car plant on the MicroAutobox or to set up a hardware in the loop test with the MicroAutobox communicating with the computer running CarMaker.

Both the Reference NMPC and the Real-time NMPC were tested to get a comparison between the full state model with 18 states and the reduced state model with 10 states. Three different prediction horizon lengths were tested for both controllers: 10, 20 and 40 samples. By doing this, an understanding of the time complexity when changing the size of the model and the length of the prediction horizon could be gained. The inputs were recorded while running Scenario 1, described in Section 5.4.1, in CarMaker.

Figure 5.22 shows the execution time for the Reference NMPC with horizon set to 20 when feeding the MicroAutobox with the inputs from Scenario 1. As can be seen, the execution time varies a lot. The maximum and average execution times from the measurements are shown in Table 5.5.

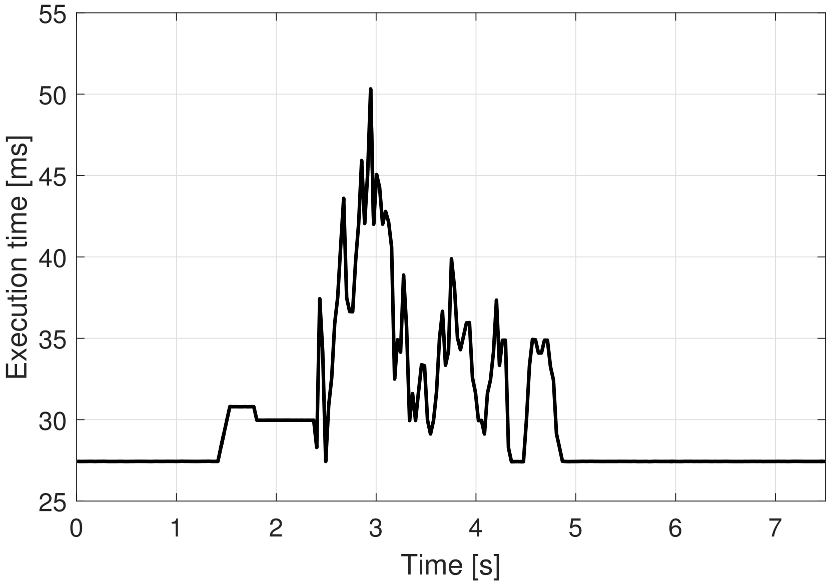


Figure 5.22: Execution time for the Reference NMPC with horizon 20 while running Scenario 1.

Table 5.5: Execution times for the Reference NMPC with 18 states and the Real-time NMPC with 10 states. The controllers were both running Scenario 1 and the execution time is shown for three different prediction horizon lengths (number of samples). The first number is the maximum recorded execution time and the second number, within parenthesis, is the average execution time for the scenario.

Horizon:	10	20	40
Reference NMPC	14.4 (11.4) ms	50.3 (30.2) ms	278.2 (106.1) ms
Real-time NMPC	5.7 (3.5) ms	23.4 (13.6) ms	123.8 (65.84) ms

6
Vehicle implementation and results

This chapter describes implementation of the controller in a real car, including available sensors and nonlinear spring and damper characteristics. The presented solutions targets running the controller in a Volvo S90 equipped with semi-active dampers and a vision system for obtaining the road preview. Finally, some experimental results are shown as a proof of concept that the controller can be run in a real car.

6.1 Available sensors

The sensors that are available in the car are three single axis accelerometers and four level sensors that measure the position and velocity of the suspension at each wheel. With these sensors, the following signals could be obtained with sufficient accuracy: suspension deflection, suspension deflection velocity, heave, roll and pitch accelerations and heave, roll and pitch velocities. The road preview is obtained with a vision system mounted in the car.

The absolute heave position and roll and pitch angles cannot be measured with the available sensors. This is a problem since the full-car model contains these states. However, a workaround is possible under the assumption that the slope of the road is small, the same assumption that was made when linearizing the full-car model. The workaround is closely tied to the output of the pre-processing of the road preview.

6.1.1 Road preview

The road preview that is fed to the controller consists of four road height profiles, one for each wheel. Both the height and the velocity of the road are available from the rear wheels to the maximum range of the vision system. The pre-processing of the road preview is outside the scope of this thesis, but it is important to understand what the road preview looks like when it is received by the controller. The samples in the road height profile is spaced by time, not by distance. The difference in time between two samples in the road preview is one sampling interval. The road height profile for each wheel is rotated and translated to fit into one calculated reference plane. The reference plane is calculated from the average slope of the road within the prediction horizon. The road preview and the initial state in the controller is then adjusted to that reference plane. The adjustment is done by setting the heave position, the roll angle and the pitch angle of the car to zero. Note that at most times this is not the same as the real state of the car body, but the real state cannot be measured. In order to keep the correct relation between the states in the model, the road height profile for each wheel is placed according to the current measurements of suspension deflection and suspension deflection velocity. This way, all the forces are calculated accurately in the model. The heave velocity, roll velocity and pitch velocity are measured accurately and are not affected by the reference plane so they are fed to the controller without modification. Figure 6.1 shows how the reference frame is calculated and how the road preview is rotated and translated before being fed to the controller for one road height profile. When pre-processing the road preview for all four wheels, they are rotated according to the same reference plane, but the starting height is placed individually based on the current suspension deflection for each wheel.

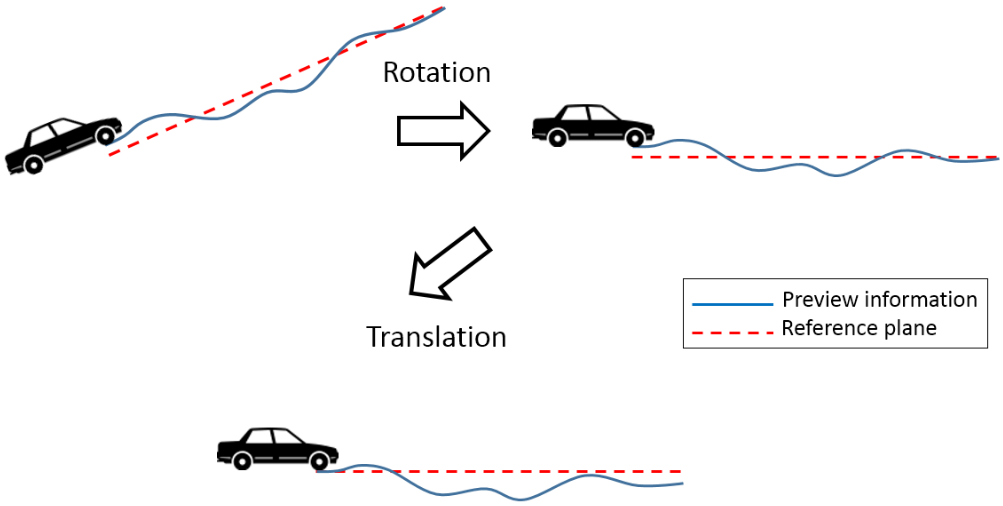


Figure 6.1: Pre-processing of the road preview in the real car before it is fed to the controller.

6.2 Real damper characteristics

The general assumption when modelling dampers is that the characteristics are linear and can be described as:

$$F(v, b) = b \cdot v, \tag{6.1}$$

where F is the force, v is the damper velocity and b is the damping coefficient. The same applies for the damper model used in the designed controllers, where b can be controlled in order to get the desired stiffness. This assumption does not work well with the characteristics of the semi-active dampers in the real car, as can be seen in Figure 6.2. The dashed lines show measured forces from the semi-active dampers for maximum, medium and minimum input current. Using the linear damper characteristics in this case means that the force will be clipped for a lot of cases without the controller knowing and in some cases it will not use the full potential of the actuators.

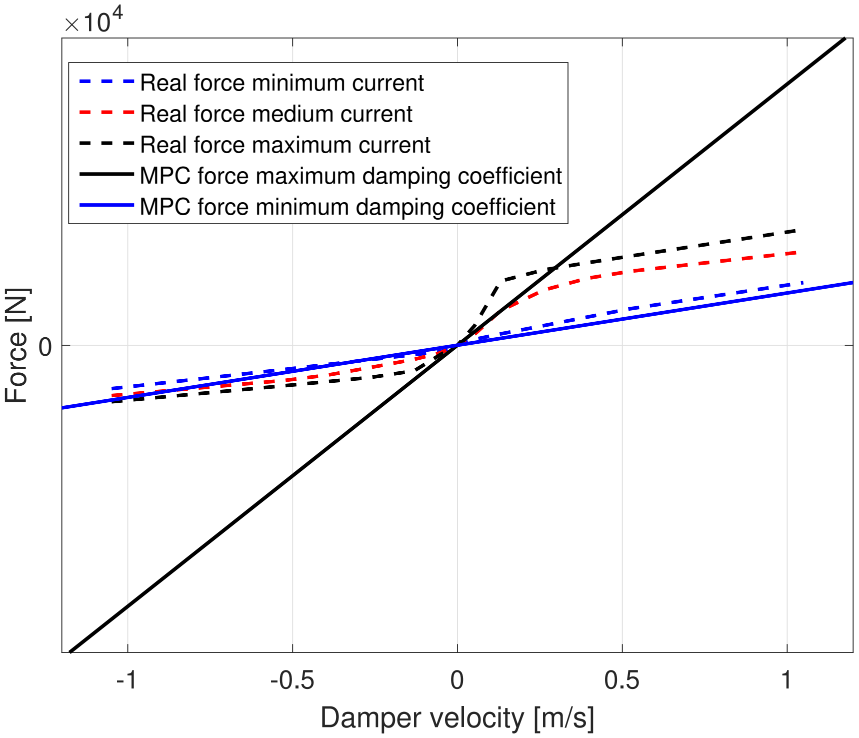


Figure 6.2: Linear damper model (solid lines) showing the force generated with the maximum and minimum damping coefficient fit to the characteristics of the real damper (dashed lines) where the curves show the force generated with maximum, medium and minimum input current.

The solution to this problem was to use the fact that NMPC was used and hence it was not expensive to introduce further nonlinearities. With inspiration drawn from Pacejka's tire model, [25], the following equation for describing the semi-active damper characteristics was found:

$$F(v,i) = a(i)\cdot(arctan(b(i)\cdot v + c(i)) - arctan(c(i))) + d(i)\cdot v, \tag{6.2}$$

where:

$$a(i) = k_a \cdot i + m_a, \tag{6.3}$$
$$b(i) = k_b \cdot i + m_b, \tag{6.4}$$
$$c(i) = k_c \cdot i + m_c, \tag{6.5}$$
$$d(i) = k_d \cdot i + m_d. \tag{6.6}$$

The control signal in this function is current since the real actuators use current as input. The parameters a, b, c and d where fit to the measured data for the maximum, medium and minimum currents. This resulted in three different sets of parameters. The parameter values had to expressed as funcitons depending on the current, i. Linear functions were used to fit how the parameters changed with the current and another round of parameters, $k_{a,b,c,d}$ and $m_{a,b,c,d}$, were found. The resulting damper model is displayed in Figure 6.3.

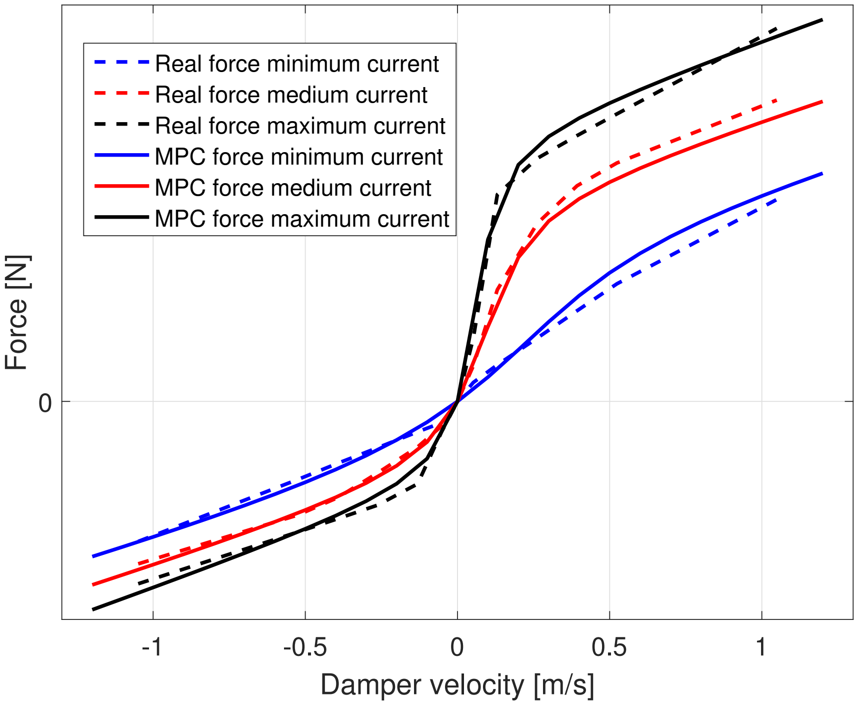


Figure 6.3: Nonlinear damper model (solid lines) fit to the characteristics of the real damper (dashed lines). The curves show the force generated with maximum, medium, and minimum input current.

6.3 Suspension travel limits

An important aspect to consider in a real car is that the travel of the suspension is limited. A car is equipped with so called bump stops or spring buffers to reduce the impact if the limit of the suspension is reached and possibly also save the car from breaking. If the controller has knowledge about these limits it could reduce the risk of hitting the bump stops and the opposite is true if the controller assumes unlimited travel.

One way to implement this in an MPC is to put constraints on the suspension deflection together with slack variables in case the actuator is not able to keep the constraints. This is expensive for a real-time solution since the optimization problem will contain more constraints and more optimization variables.

The proposed solution is to add the limits as a cost by modelling a significant increase of the force from the springs when the travel limit is reached. This is done with the following equation:

$$F(x) = a \cdot (arctan(b \cdot x - c) + arctan(b \cdot x - d) + arctan(c) + arctan(d)), \quad (6.7)$$

where F is the force, x is the suspension deflection and a, b, c and d are parameters chosen so that the function fits to the real characteristics of the spring buffers. The function is shown in Figure 6.4 together with the real characteristics. Note that the slope at the ends of the function is purposely decremented in order to avoid problems with very large numbers in the optimization problem.

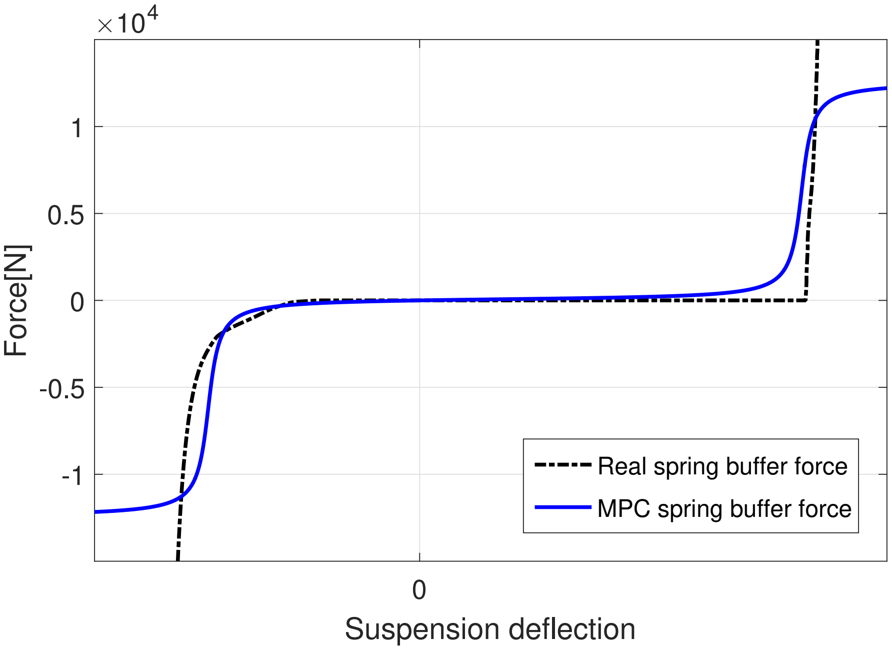


Figure 6.4: Nonlinear spring buffer model (solid lines) fit to the characteristics of the real spring buffer (dashed lines).

6.4 Controller implementation

The controller that was used for the experimental tests, henceforth called the Preview controller is the Real-time NMPC, described in section 4.5, with two additions: the force from the spring buffers, described in Section 6.3, and the nonlinear damper characteristics from Section 6.2. Because the nonlinear damper characteristics were added, the Preview controller outputs the current that goes directly to the semi-active dampers instead of the damping coefficient. The sampling time for the experimental tests was set to 30 milliseconds and the number of prediction steps was set to 12. This meant that the prediction time was 0.36 seconds. An overview of the tuning parameters for the Preview controller can be found in Table 6.1. The magnitude of the new control signals were slightly different, so the weights were adjusted to keep the controller behaviour approximately the same as for the Real-time NMPC.

Table 6.1: Tuning parameter values for the Preview controller.

Parameter	**Value**	**Description**
Δt	0.03	Sampling time (s)
N	12	Prediction horizon (steps)
W_k	$diag(1,1,1,1,1,1,1)$	Weight matrix
W_N	$diag(1,1,1)$	Weight matrix (last step)

6.5 Simulation results

To ensure that the Preview controller still worked well after the modifications, it was tested on Scenario 1, described in Section 5.4.1 and compared to the passive car. For this test, a model of the real actuators was used (instead of an ideal actuator) including signal delays and dynamics for building force. The inputs to the controller and the preview signal were kept ideal.

The resulting performance of the Preview controller was better than the passive car and as good as the Real-time NMPC for heave acceleration and pitch acceleration, which can be seen in Figure 6.5 and Table 6.2. The roll accelerations were, however, a lot worse. This might be a consequence of the delays in the actuator model and because the road in the scenario mainly excites heave and pitch, the prediction of roll in the controller might not be accurate. The control signals for the front left and rear right wheel during the scenario is shown in Figure 6.6.

Table 6.2: RMS errors for the body accelerations during Scenario 1.

	Heave acc	***Roll acc***	***Pitch acc***
Passive:	0.71	0.054	0.73
Real-time NMPC:	0.56	0.039	0.53
Preview controller:	0.57	0.1	0.54

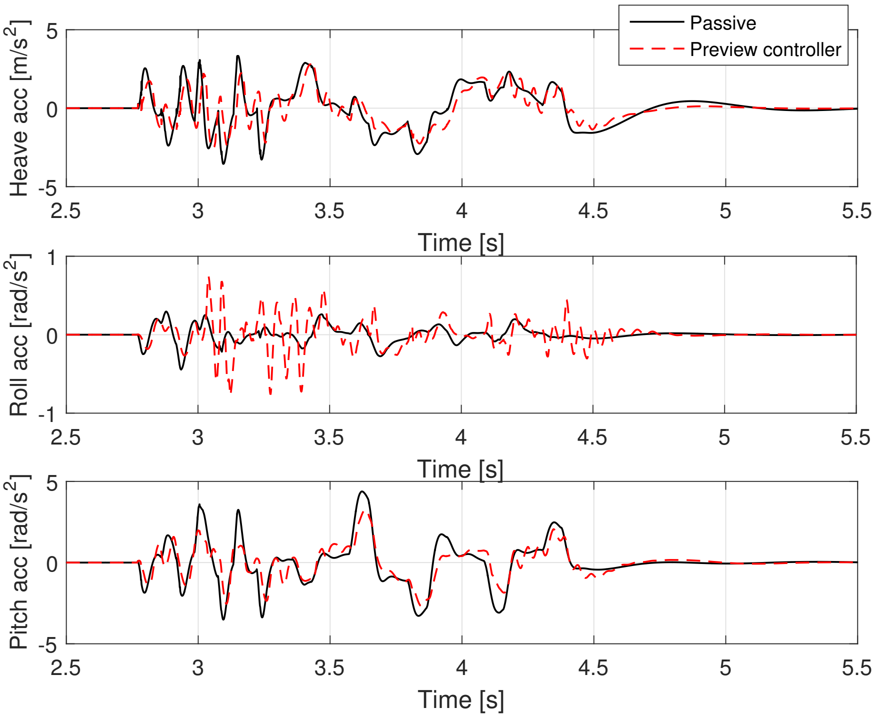


Figure 6.5: Heave, roll and pitch accelerations of the vehicle body for Scenario 1.

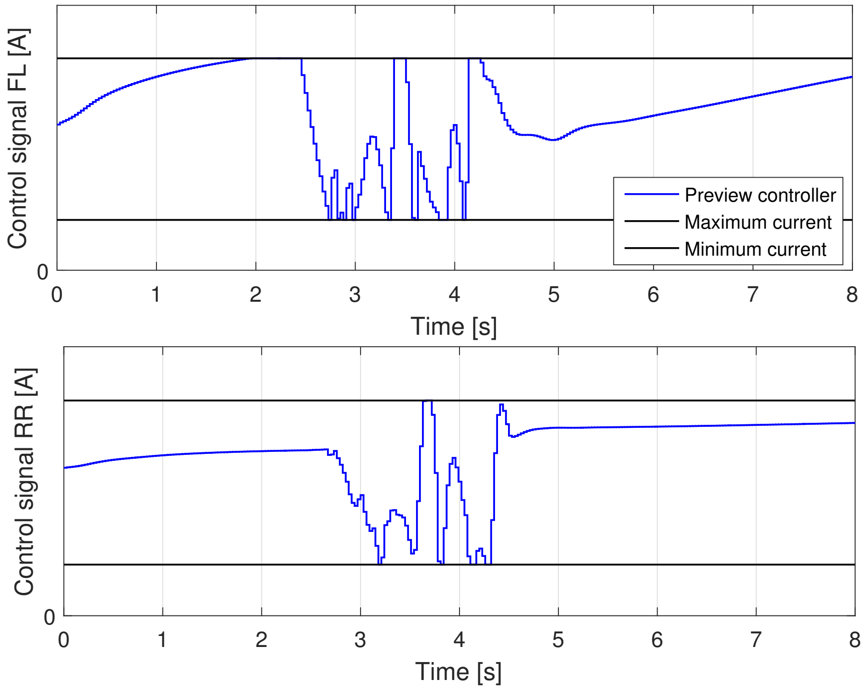


Figure 6.6: Control signals for the front left and rear right wheel for the Preview controller.

6.6 Experimental results

To obtain experimental results, the controller was run on a MicroAutobox mounted in a Volvo S90 equipped with semi-active dampers and a vision system for collecting road preview. The testing was performed on Hällered Proving Ground (HPG) on a road section containing bumps that mainly exited heave and pitch. The road section was divided into three parts, staring with 5 small bumps followed by 6 larger bumps and finally 5 smaller bumps. The car was driven over the bumps twice at a speed of 60 km/h. The first time, the current sent to the dampers were fixed to the medium current, in order to get the behaviour of a passive car. The second time, the damper currents were controlled by the Preview controller.

A comparison of the measured body accelerations between the passive car and the Preview controller is shown in Figure 6.7 and the RMS errors can be found in Table 6.3. For heave and pitch accelerations, the Preview controller is marginally worse than passive on heave and marginally better on pitch. For the roll acceleration, the result is similar to the simulations as roll acceleration is notably worse with the Preview controller.

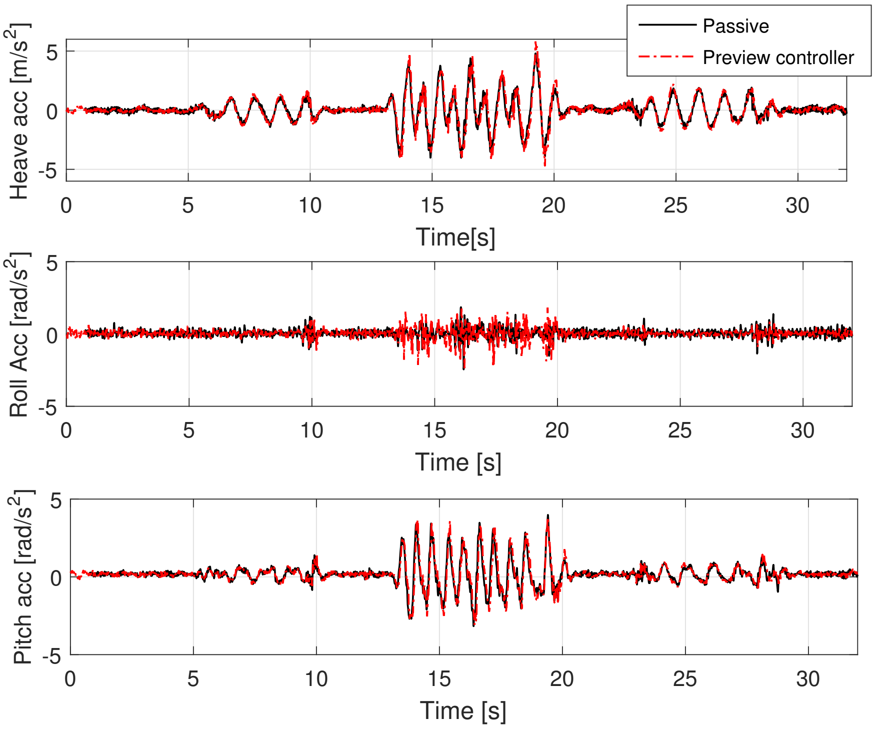


Figure 6.7: Heave, roll and pitch accelerations of the vehicle body measured by the accelerometers in the car.

Table 6.3: RMS errors for the body accelerations during the test run on HPG.

	Heave acc	***Roll acc***	***Pitch acc***
Passive:	1.0647	0.2817	0.7681
Preview controller:	1.0785	0.3411	0.7506

The control signals, seen in Figure 6.8 for the front left and rear right wheels, show that the controller is working during the test run. The full range of the current is used but the upper and lower bounds are kept. An interesting difference compared to the simulation results is that the controller chooses a soft damping when the road disturbance is small.

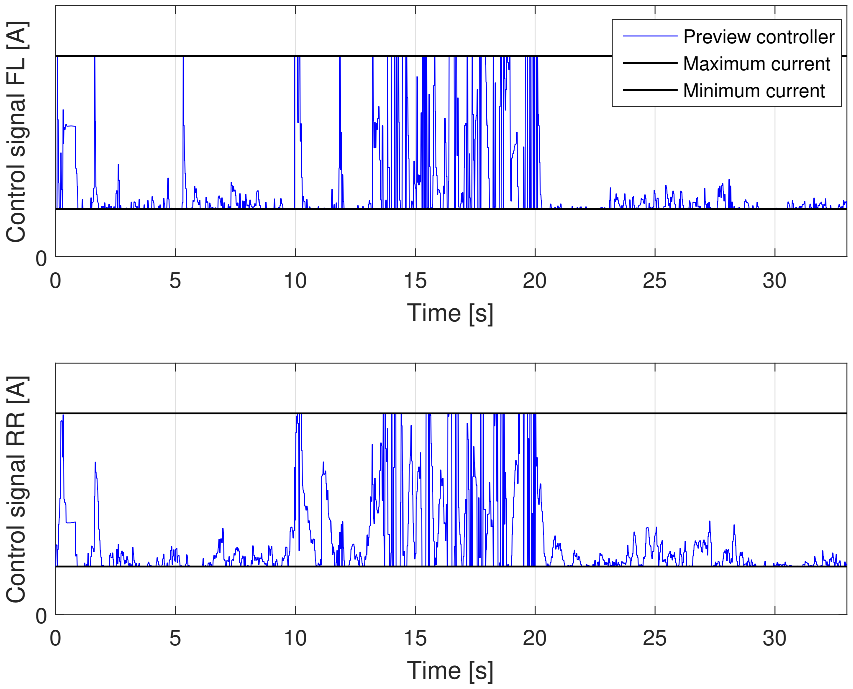


Figure 6.8: Control signals from the Preview controller for front left and rear right wheel.

The experimental results showed that there are things to work on in terms of performance but they also showed potential since the controller was able to function in real-time with real inputs in a real car.

7
Discussion and Conclusion

In this chapter, the objective and the research questions are recalled and discussed in conjunction with the obtained results. Furthermore, actions for future work are suggested. Then conclusions are drawn regarding how well the purpose of the thesis was achieved.

7.1 Discussion

The objective was to achieve an implementation of an MPC that was utilizing road preview to control the semi-active suspensions of a car. There were also the two requirements that (1) the controller should be based on a full-car model capable of describing the movements of heave, roll and pitch and (2) the controller should be able to run in real time. (1) was achieved because the implemented MPC is based on a full-car model. (2) was achieved since by using a reduced full-car model the controller could be run in real time. The reduced full-car model also retained the important correlation between heave, roll and pitch, despite that the tire dynamics were neglected. This claim is supported by the results in Section 5.3 where the full-car model and the reduced full-car model were compared to the CarMaker model and it can be seen that the full-car model and the reduced full-car model behaves very similar to each other.

Since the primary objective was reached, the additional objective to test the controller in a real car could also be achieved. In order to make it possible for the real implementation to work well, further developments to the controller were proposed and the limitations of using real sensors and real semi-active actuators were addressed. To show that the real implementation worked, a proof of concept test was run on Hällered Proving Ground. The test showed that the controller was able to run in real time, even with the extra load of communicating with the camera and the dampers and being fed imperfect sensor data. One interesting difference between how the controller worked in the car compared to in simulations was that the control signal was set to give soft damping when the road disturbances were small. In simulations, the disturbances from the road profile were zero when driving on a flat road which means that the controller stops the oscillations induced by driving over for example a bump by setting a high damping coefficient. However, that is not the case for a real road and to minimize the body accelerations introduced by these small disturbances, the controller chose a soft damping. The test also showed that the controller did not manage to reduce the body accelerations compared to a passive car, but the fact the the simulation results showed great potential suggests that there is a lot of room for improvement for the real car implementation.

7.1.1 Research questions

1) What is the best way to utilize road preview together with MPC for semi-active suspension in order to improve ride comfort?

It was seen during the literature review that the best way to utilize road preview together with semi-active suspension is to use MPC as the control strategy. The reason for this is that MPC makes it possible incorporate the road preview and solve for the optimal control signals while accounting for the dissipative constraints of the semi-active dampers. In order to use the full potential of the controller, the vehicle model that the MPC is based on should incorporate all relevant dynamics of the real vehicle. The full-car model fits this purpose since the vertical dynamics are of interest when considering the suspension. The most important difference from previous work done in the field is the reformulation of the model so that the damping coefficients are used as control signals instead of the forces exerted by the dampers. By doing this, NMPC was required to solve the problem since the model become nonlinear, but in turn it removed the need for an MIQP solver to handle the constraints on the damper force. It also made it possible to tap into the extensive research done to make real-time NMPC solvers and in this case make use of ACADO Toolkit. Using a nonlinear plant model also opened up the possibility to incorporate nonlinear damper and spring characteristics without substantially increasing the computation time, which is a huge advantage. In conclusion, the answer to the question is that the best way that was found was to use a full-car model with damping coefficient as control signal in an NMPC with the objective to minimize body accelerations.

2) What simplifications or assumptions have to be made in order to make a real-time implementation feasible?

In order to achieve a real-time implementation of the controller the main issue is to reduce the computational time. Modelling the force that is exerted by the semi-active actuators as the control signal leads to a linear model with nonlinear constraints on the control signal. This strategy leads to a MIQP problem that is NP-hard and therefore not suitable for a real-time implementation. Therefore, the main assumption in order to make a real-time implementation feasible was to formulate the problem with the damping coefficient as the control signal where a framework for NMPC could be used and the resulting optimization problem became a QP-problem instead. However, other simplifications were required in order to be able to run the controller on a real-time system. The vehicle model was reduced to contain 10 state variables instead of 18 state variables which reduced the computational time by approximately a factor of two. The prediction horizon was shortened in order to make the optimization problem smaller and it was seen that reducing the prediction horizon lowered the computational time quadratically. The sampling time was also increased in order to give the controller more time to to calculate the control signals.

3) How does the performance of a real-time implementation differ from an implementation that does not consider the real-time constraints?

A reference controller was designed without any simplifications due to a real-time implementation, such as limitations on number of state variables, sampling time and length of prediction horizon. The real-time controller was compared to this reference controller and the RMS-errors of the body accelerations were evaluated for both controllers during four different test cases. The results are presented in Section 5.4 which indicates that the limitations of a real-time solution does not decrease the performance a lot. The reference controller has lower RMS-errors for all body accelerations in all test cases, but the real-time controller is only marginally worse in terms of reducing the vehicle body accelerations.

7.2 Future work

The implementation of the controller in the car is implemented as a proof of concept that the controller can be run in real-time while using inputs from the car and the vision system. In order to improve the performance there are still work to be done. The controller assume that the inputs are ideal, which is not the case for the real implementation. Hence, further testing of how the inputs affects the performance of the controller is appropriate. The execution time varies depending on the current input to the controller, which can be seen in Figure 5.22. To guarantee that the controller will always meet the real-time constraints, the sampling time is chosen with enough marginal so that the maximum execution time is lower than the sampling time. However, since the average execution time is significantly lower than the maximum execution time it may be desirable to limit the execution time in order to reduce the sampling time. A solution would be to monitor the controller and employ a fast, backup control strategy for the case when the controller is not able to find a solution within a specified upper bound on the execution time.

7.3 Conclusion

The purpose of the thesis was to improve ride comfort, which is the same as lowering body accelerations, by using preview of the road height profile with semi-active suspension. For this purpose, a nonlinear model predictive controller (NMPC) was designed and implemented. In order to make a real-time implementation feasible two key design concepts were proposed, the definition of damping coefficient as control signal and the reduced full-car model. The common way to formulate the problem to control semi-active suspension is to use a linear vehicle model with nonlinear constraints on the damping force which is the control signal. This results in a MIQP problem that is infeasible for a real-time implementation. This thesis proposed a nonlinear vehicle model with linear constraints on the the damping coefficient which is the control signal. This made it possible to use an existing framework for NMPC, by which a standard QP solver was used. The other key concept that is proposed is to use a reduced full-car model with 6 state variables instead of 14 state variables. This model reduction reduce the complexity of the model by neglecting the wheel dynamics and reduce the computational time significantly. The reduced full-car model was compared and verified against a full-car model and the realistic vehicle model in CarMaker which showed a similar behaviour between the models. The proposed controller was able to run in real-time and this was verified by measuring the execution time when running the controller on a real-time platform. The simulation results where a real-time controller was compared to a passive car and a reference controller without simplifications due to a real-time implementation, showed unanimously that the performance of the real-time controller was slightly worse than the reference controller but significantly better than a car with passive suspension. This indicates that ride comfort can be improved by designing an real-time NMPC utilizing road preview for semi-active suspension.

The controller were implemented and tested in a real-car as a proof of concept that the controller works in real-time while using real inputs from the car as well as the preview information from the vision system. The resulting body accelerations when driving on a track consisting of bumps were very similar to a car with passive suspension. Since the performance of the controller showed great potential in the simulation results future work will be to improve the implementation in a real car where there are things to work on to increase the performance of the controller.

Bibliography

[1] Yufen Zhou and Suren Chen. Vehicle ride comfort analysis with whole-body vibration on long-span bridges subjected to crosswind. *Journal of Wind Engineering and Industrial Aerodynamics*, 155:126–140, 2016.

[2] Pieter Schalk Els, Nicolaas J Theron, Petro E Uys, and Michael John Thoresson. The ride comfort vs. handling compromise for off-road vehicles. *Journal of Terramechanics*, 44(4):303–317, 2007.

[3] Manh-Quan Nguyen, Massimo Canale, Olivier Sename, and Luc Dugard. A model predictive control approach for semi-active suspension control problem of a full car. In *Decision and Control (CDC), 2016 IEEE 55th Conference on*, pages 721–726. IEEE, 2016.

[4] Klaus Wolff, R Kraaijeveld, and J Hoppermans. Objective evaluation of subjective driving impressions. *2008 JSAE Annual Cong. Yokohama, SAE Paper*, (2008-08):0222, 2008.

[5] Rolf Findeisen and Frank Allgöwer. An introduction to nonlinear model predictive control. In *21st Benelux Meeting on Systems and Control*, volume 11, pages 119–141. Technische Universiteit Eindhoven Veldhoven Eindhoven, The Netherlands, 2002.

[6] David F Griffiths and Desmond J Higham. *Numerical methods for ordinary differential equations: initial value problems*. Springer Science & Business Media, 2010.

[7] Tor A Johansen. Introduction to nonlinear model predictive control and moving horizon estimation. *Selected Topics on Constrained and Nonlinear Control*, 1:1–53, 2011.

[8] G Prokop and RS Sharp. Performance enhancement of limited-bandwidth active automotive suspensions by road preview. *IEE Proceedings-Control Theory and Applications*, 142(2):140–148, 1995.

[9] Raman K Mehra, Jayesh N Amin, Karl J Hedrick, Carlos Osorio, and Srinivasan Gopalasamy. Active suspension using preview information and model predictive control. In *Control Applications, 1997., Proceedings of the 1997 IEEE International Conference on*, pages 860–865. IEEE, 1997.

[10] Christoph Göhrle, Andreas Wagner, Andreas Schindler, and Oliver Sawodny. Active suspension controller using mpc based on a full-car model with preview information. In *American Control Conference (ACC), 2012*, pages 497–502. IEEE, 2012.

[11] Christoph Gohrle, Andreas Schindler, Andreas Wagner, and Oliver Sawodny. Design and vehicle implementation of preview active suspension controllers. *IEEE Transactions on Control Systems Technology*, 22(3):1135–1142, 2014.

[12] Christoph Gohrle, Andreas Schindler, Andreas Wagner, and Oliver Sawodny. Model predictive control of semi-active and active suspension systems with available road preview. In *Control Conference (ECC), 2013 European*, pages 1499–1504. IEEE, 2013.

[13] Massimo Canale, Mario Milanese, and Carlo Novara. Semi-active suspension control using "fast" model-predictive techniques. *IEEE Transactions on control systems technology*, 14(6):1034–1046, 2006.

[14] Tom van der Sande, Igo Besselink, and Henk Nijmeijer. Control of a semi-active suspension system for comfort by means of a rule-based controller. In *Decision and Control (CDC), 2014 IEEE 53rd Annual Conference on*, pages 5929–5934. IEEE, 2014.

[15] Mahmoud Ahmed and Ferdinand Svaricek. Preview control of semi-active suspension based on a half-car model using fast fourier transform. In *Systems, Signals & Devices (SSD), 2013 10th International Multi-Conference on*, pages 1–6. IEEE, 2013.

[16] Alexander Domahidi and Juan Jerez. FORCES Professional. embotech GmbH (`http://embotech.com/FORCES-Pro`), July 2014.

[17] B. Houska, H.J. Ferreau, and M. Diehl. ACADO Toolkit – An Open Source Framework for Automatic Control and Dynamic Optimization. *Optimal Control Applications and Methods*, 32(3):298–312, 2011.

[18] H.J. Ferreau, A. Potschka, and C. Kirches. qpOASES webpage. http://www.qpOASES.org/, 2007–2017.

[19] B. Houska, H.J. Ferreau, M. Vukov, and R. Quirynen. ACADO Toolkit User's Manual. http://www.acadotoolkit.org, 2009–2013.

[20] Alexander Domahidi, Aldo U Zgraggen, Melanie N Zeilinger, Manfred Morari, and Colin N Jones. Efficient interior point methods for multistage problems arising in receding horizon control. In *Decision and Control (CDC), 2012 IEEE 51st Annual Conference on*, pages 668–674. IEEE, 2012.

[21] dSPACE. Microautobox hardware. `https://www.dspace.com/en/pub/home/products/hw/micautob/microautobox2.cfm`, 2018.

[22] H.J. Ferreau, A. Potschka, and C. Kirches. qpOASES embedded webpage. https://projects.coin-or.org/qpOASES/wiki/QpoasesEmbedded, 2007–2017.

[23] IPG Automotive. Carmaker: Virtual testing of automobiles and light-duty vehicles. `https://ipg-automotive.com/products-services/simulation-software/carmaker/`, 2018.

[24] dSPACE. Control desk. `https://www.dspace.com/en/inc/home/products/sw/experimentandvisualization/controldesk.cfm`, 2018.

[25] Hans B Pacejka and Egbert Bakker. The magic formula tyre model. *Vehicle system dynamics*, 21(S1):1–18, 1992.