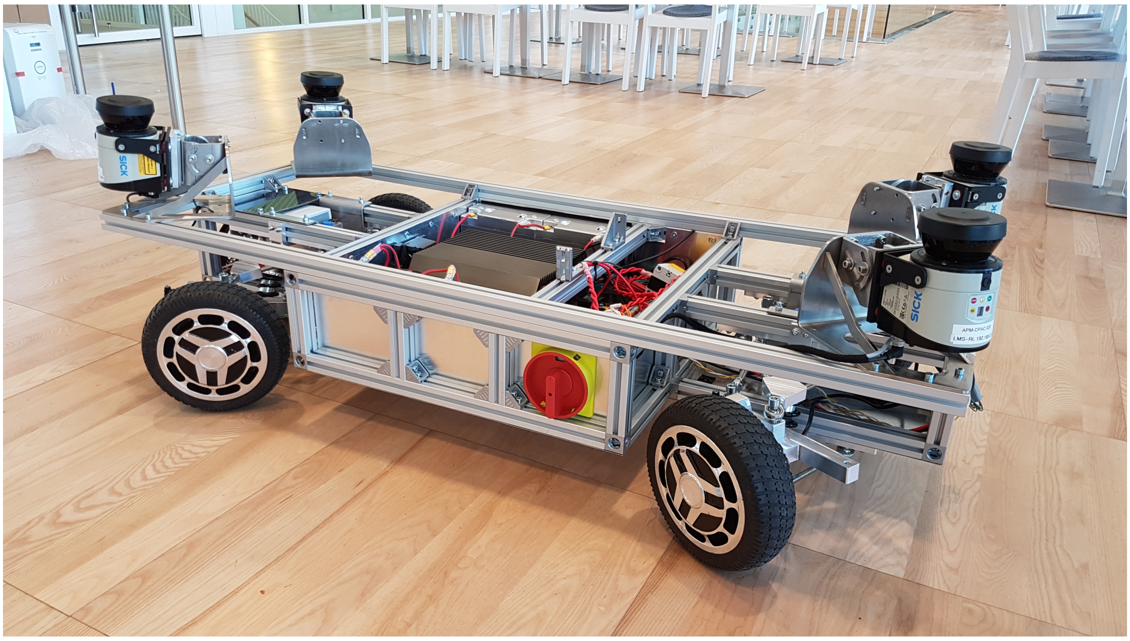
CHALMERS
UNIVERSITY OF TECHNOLOGY



Driveline control for over-actuated electric autonomous ground vehicles

Master's thesis in System, Control and Mechatronics

JACOB BRICKNER
SIMON LILJEQVIST

Department of Signals and Systems
CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden 2017

Master's thesis EX073/2017

Driveline control for over-actuated electric autonomous ground vehicles

JACOB BRICKNER
SIMON LILJEQVIST

Department of Signals and Systems
Division of Systems and Control
Chalmers University of Technology
Gothenburg, Sweden 2017

Driveline control for over-actuated electric autonomous ground vehicles
JACOB BRICKNER
SIMON LILJEQVIST

Supervisor: Michael Pettersson, CPAC Systems AB
Examiner: Knut Åkesson, Signals and Systems, Chalmers University of Technology

Master's thesis EX073/2017
Department of Signals and Systems
Division of Systems and Control
Chalmers University of Technology
SE-412 96 Gothenburg
Telephone +46 31 772 1000

Typeset in LaTeX
Gothenburg, Sweden 2017

Driveline control for over-actuated electric autonomous ground vehicles
JACOB BRICKNER
SIMON LILJEQVIST
Department of Signals and Systems
Chalmers University of Technology

Abstract

In this thesis, which was carried out at CPAC Systems AB, a driveline control system was designed for an existing small sized autonomous vehicle. The vehicle was 4WD and had Ackermann steering on both the front and rear axle and the references for the control system was heading and velocity, hence the system was over-actuated. A control structure with three layers was used; a high level controller calculated the global reference forces, the second layer allocated each actuators effort and the last layer controlled each actuator. Several different control allocation algorithms were compared before a quadratic programming approach was selected.
The driveline control system was implemented and tested in MATLAB® Simulink® on a derived four-wheel model and on the existing test vehicle. The results showed that dividing the control structure into several layers enabled straight forward design of the control allocation problem and also easy tuning. The selected control allocation algorithm showed promising results in the tests. However, given that the vehicle only was traveling at low velocities it was not pushed close to its limits and therefore a simpler control allocation algorithm could have been used that would result in the same behaviour.

Keywords: Control allocation, over-actuated vehicle, quadratic programming, four-wheel driven, Ackermann steering, electric vehicle, autonomous vehicle.

Acknowledgements

We would like to thank our supervisor and examiner Knut Åkesson at Chalmers University of Technology for his thoughts and feedback, especially regarding this report. We would also like to thank our supervisor at CPAC Systems AB Michael (Mippen) Pettersson for his help on general engineering principles, especially he reminded us to try simple solutions before developing them further. He also provided invaluable experience on embedded systems and open source hardware. Professor Peter Forsberg at CPAC Systems AB provided many ideas and knowledge on everything from the control allocation approach to how to rewire circuits in an electrical motor. He also provided invaluable feedback on our thesis and early hand-ins, without his help this thesis would not have been possible. We also would like to thank Professor Paolo Falcone from the discussion on control allocation and Professor Fredrik Bruzelius from the discussion on tire models.

Finally, we thank our respectively families and friends for their support and encouragement during hard and stressful times, both during this master thesis and also during our education to become engineers.

Jacob Brickner and Simon Liljeqvist, Gothenburg, 2017

Nomenclature

Constants and variables

α	Tires slip angle
α_i	Constants in a low pass filter
β	Tires side slip angle
$\boldsymbol{\delta}$	Control effort
$\boldsymbol{\delta}_{\mathrm{cmd}}$	Commanded control effort
$\boldsymbol{\delta}_{\mathrm{cmdll}}$	Commanded control effort
$\boldsymbol{\delta}_{\mathrm{cmd}n}$	n'th commanded control effort in daisy chaining
$\boldsymbol{\delta}_{\mathrm{d}}$	Desired control effort
$\mathbf{v}$	Virtual command vector with F_{x} and M_{z}
$\overline{\boldsymbol{\delta}}$	Upper limit of the control effort
$\underline{\boldsymbol{\delta}}$	Lower limit of the control effort
$\boldsymbol{B}$	Effectiveness matrix
$\boldsymbol{B}^{\#}$	Weighted pseudo inverse of the effectiveness matrix
$\boldsymbol{B}^{\dagger}$	Pseudo inverse of the effectiveness matrix
$\boldsymbol{I}$	Identity matrix
$\boldsymbol{r}$	Reference vector
$\boldsymbol{W}$	Weighting matrix
$\boldsymbol{W}_{\boldsymbol{\delta}}$	Weighting matrix control effort
$\boldsymbol{W}_{\mathrm{v}}$	Weighting matrix virtual command
$\boldsymbol{W}_{\mathrm{Fx}}$	Weighting factor for the force in the x-direction
$\boldsymbol{W}_{\mathrm{Mz}}$	Weighting factor for the rotational force in the z-direction
$\ddot{\theta}$	Heading acceleration
$\dot{\theta}$	Heading velocity
γ	Weighting factor
$\hat{\theta}$	Estimated heading of the vehicle
τ_{ij}	Motor torque
θ	Heading
θ_{r}	The reference heading
$\theta_{\mathrm{steering}}$	Explicit ganging virtual steering command
θ_{ij}	Steering angle at the wheels
θ_i	Steering angle at the front or rear axle
$\theta_{i,\mathrm{r}}$	Steering angle reference at the front or rear axle
a	Distance, front axle to COG
a_{x}	Acceleration in x direction
a_{y}	Acceleration in y direction
b	Distance, rear axle to COG

B_{y}	Parameter in Pacejka's magic formula
C_{α}	Cornering stiffness
C_{erpm}	ERPM to velocity mapping constant
c_{roll}	Roll resistance coefficient
c_{wind}	Wind resistance coefficient
C_{y}	Parameter in Pacejka's magic formula
D_{y}	Parameter in Pacejka's magic formula
$e(t)$	A PID controllers error
e_{cmd}	The error on the commanded control effort
E_{y}	Parameter in Pacejka's magic formula
F_{x}	Force in the longitudinal direction
$f_{\mathrm{x}i}$	Longitudinal force generated by a tire in a bicycle model
$F_{\mathrm{x}ij}$	Longitudinal force generated by a tire in a four wheel model, in the vehicles coordinate system
$f_{\mathrm{x}ij}$	Longitudinal force generated by a tire in a four wheel model, in the tires coordinate system
f_{y}	Lateral force generated between a tire and the ground in the y-direction
$F_{\mathrm{z,nominal}}$	Nominal force between a tire and the ground
$F_{\mathrm{z}ij}$	Normal force between a tire and the ground
$G_{\mathrm{Motors}}(s)$	A motors transfer function in the Laplace domain
$G_{\mathrm{Steering}}(s)$	A motors transfer function in the Laplace domain
I_{z}	Inertia around z-axis
i_{ij}	Motor current
K_{d}	Derivative gain
K_{i}	Integral gain
K_{p}	Proportional gain
K_{T}	Torque constant
m	Mass of vehicle
M_{z}	Moment around the z-axis
N_{Poles}	Number of poles in the motor
$PKY1$	Parameter in Pacejka's magic formula
$PKY2$	Parameter in Pacejka's magic formula
R	Turning radius
r	Wheel radius
$r(t)$	A PID controllers reference
R_{gear}	Gear ratio
RPM_{Wheel}	Revolutions per minute at the wheels
t	Track width
T_{wheels}	Explicit ganging virtual torque command
T_{i}	Torque at a wheel in a bicycle model
$T_{ij,\mathrm{r}}$	Torque reference at a tire
$Teeth_{\mathrm{Motor}}$	Gear teeth on the motor
$Teeth_{\mathrm{Wheel}}$	Gear teeth on the wheel
$u(t)$	A PID controllers control signal
$v_{\mathrm{x,r}}$	The reference velocity in the x direction
$v_{\mathrm{x,wheel}}$	A tires velocity in the x-direction

v_{x}	Velocity in x direction
$v_{\mathrm{y,wheel}}$	A tires velocity in the y-direction
v_{y}	Velocity in y direction
$w_{\mathrm{steering}i}$	Weighting factor for a steering angle
$w_{\mathrm{wheel}ij}$	Weighting factor for a wheel
x	The vehicle states
x_{wheel}	A tires x-direction
y_{wheel}	A tires y-direction

Abbreviations

4WD	Four wheel drive
AV	Autonomous vehicles
BBB	BeagleBone Black
BLDC	Brushless DC
CA	Control allocation
CAN	Controller Area Network
COG	Center of Gravity
ERPM	Electrical revolutions per minute
ESC	Electronic speed controller
IIR	Infinite impulse response (filter)
LP	Linear programming
MPC	Model Predictive Control
MPCA	Model Predictive Control Allocation
PID	Proportional-Integral-Derivative
PWM	Pulse Width Modulation
QCAT	Control Allocation Toolbox
QP	Quadratic programming
ROS	Robotic Operating System
RPM	Revolutions per minute
VESC	Electronic speed control by Benjamin Vedder

Contents

List of Figures

List of Tables

1 Introduction

The mining industry today is becoming safer and more efficient as the technology constantly evolves. However, there are still areas that clearly prevents further progress in productivity, mainly because of the dangers involved. The biggest of these areas is the discontinuation of operation as result of the toxic fumes that arises after drilling and blasting, which makes the environment too hazardous for humans [1]. Hence, the parts of the mines which have to high levels of toxic fumes needs to be evacuated. The time before the workforce can enter again is often called re-entry time and the standard way to make it shorter is to use ventilation systems.
The progress made in sensor technology, computational power and algorithms have created the possibility to use autonomous vehicles (AV). These have the advantage of being capable of operating in toxic environments, therefore they only need to leave the blast zone area, this results in a very short re-entry time, [2].
An investigation, [3], showed several advantages of using autonomous haul trucks instead of manually driven. The productivity were increased by 21.3 %, the fuel consumption were decreased by 5.3 % and the tire wear decreased by 7.6 %. One big reason for the increased productivity was that the autonomous haul trucks could operate continuously in contrast to the human workers which needed breaks.
Autonomous vehicles also have the potential to improve the safety of today´s traffic environment as human drivers are responsible for almost 90 % of all accidents with large trucks [4]. Autonomous vehicles are not subjected to many of the things which humans are, for example tiredness and lack of focus on the driving. Autonomous vehicles are therefore one way to possibly decrease the number of accidents.
Both the society and mining companies can benefit if the transition to using autonomous vehicles happens quickly, one way to improve the speed of this development is to make it easier to verify and test the functionality of algorithms for autonomous driving. Presently at CPAC Systems AB, testing of the algorithms are either carried out in simulations or on full-scale trucks, creating a development gap. This gap can be made smaller by using a small test vehicle which allows to test algorithms in a non idealized environment. If the vehicle is electrical the testing can be carried out indoors, this will be an efficient way to test.
The relative small size of electric motors have made it possible to in a space efficient manner have one motor per wheel. This allows the wheels torques to depend on the conditions and physical limits for each wheel instead of the vehicle as a whole which can improve the handling and performance. However increasing the number of actuators in a system can result in that there are several ways to combine the efforts of the actuators to achieve the same total effort. These kind of systems are over-actuated and the way the efforts of the actuators are distributed is called

control allocation (CA).

1.1 Main task

The main task of this thesis was to design, implement and evaluate a complete driveline control system, see Figure 1.1, suitable for an existing over-actuated vehicle, see Section 2. A model-based development approach had to be used, hence the complete control system had to be designed in MATLAB® Simulink® [5]. The control system had to control the vehicle such that it followed a reference heading and a reference velocity.

The driveline control system had to be implemented on a vehicle in the software Robotic Operating System (ROS). Finally the choice of driveline control system had to be tested by performing test scenarios with the vehicle. The tests had to be evaluated with the simulation results to point out the advantages and disadvantages with the implemented driveline control system to be able to answer if the selected approach was good.

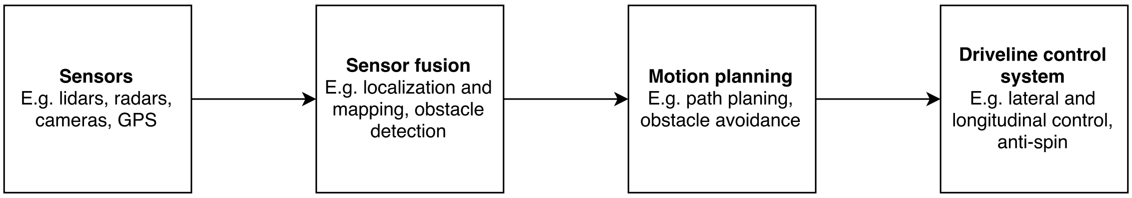

Sensors
E.g. lidars, radars, cameras, GPS
Sensor fusion
E.g. localization and mapping, obstacle detection
Motion planning
E.g. path planing, obstacle avoidance
Driveline control system
E.g. lateral and longitudinal control, anti-spin

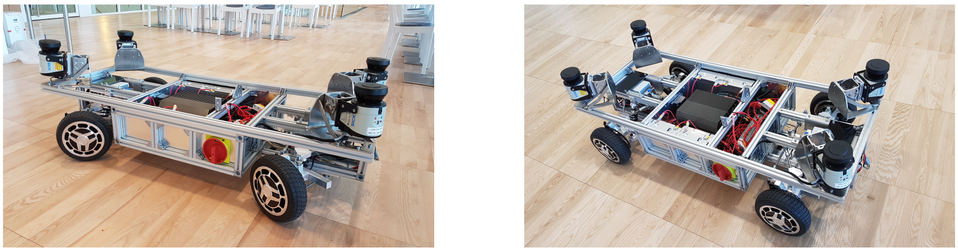
Figure 1.1: *The figure shows a block diagram of a typical structure of the data flow for autonomous vehicles, depending on which approach is used the subsystems could merge. This thesis focus on the last subsystem, the driveline control system.*

1.2 Thesis outline

Chapter 2 provides an introduction to the existing vehicle with descriptions on all hardware components and software. In Chapter 3 the requirements on the closed-loop system are stated. Chapter 4 describes the approach used for the control system and how it is divided into several layers. In Chapter 5 is a simple tire model derived and also a bicycle model and a four wheel model of the vehicle. In Chapter 6 is each layer in the control system described in depth including the relevant theory. Chapter 7 describes how a simulation environment was designed to evaluate the control system for the previously mentioned four wheel model. How the control system were implemented and tuned in the vehicle is described in Chapter 8. In Chapter 9 is the designed control system discussed and future recommendations are stated.

2
Test vehicle

As described in the introduction the driveline control system was implemented and evaluated on an existing small sized vehicle. The vehicle is described in this chapter for creating the general understanding of the platform, which is useful for the future chapters. First the mechanical and electrical hardware are described and finally the sensors, computers and software that was used. Before this thesis began the mechanical hardware, tires, electrical motors and steering system was already assembled. During this project the rest of the systems which are described below was added and integrated. The test vehicle can be seen in Figure 2.1.



(a) *The test vehicle seen from the side.*

(b) *The test vehicle seen from above.*

Figure 2.1: *The test vehicle on which the driveline control system were implemented and tested. The lidar sensors placed in each corner are not used in this thesis.*

2.1 Mechanical hardware

The vehicle is developed to be similar in functionality to the autonomous and electric Volvo prototype HX01 load carrier [6]. It is built from an easy adaptable metal frame that holds all the mechanical and electrical hardware together. The four wheels are connected to the frame by a suspension system. Each wheel pair is connected to each other by an Ackermann steering system [7]. The main body of the test vehicle is where most of the electrical systems are mounted. The vehicle can be seen in Figure 2.1a and 2.1b, the main vehicle parameters can be seen in Table 5.1.

2.2 Electrical hardware

The driveline of the vehicle consists of four electric Brushless DC (BLDC) motors that are integrated in the wheels. All motors are controlled individually by four

VESC motor controllers, which is an electronic speed controller (ESC) [8]. The signals is send to one of them by a Serial interface which in turn sends the other three commands over a Controller Area Network (CAN) interface. The Ackermann steering systems is connected to DC motors that are Pulse Width Modulation (PWM) controlled by the motor driver Sabertooth 2x25 [9]. The power to all the electrical systems comes from four 12 V batteries connected such that 24 and 48 V can be used with a common ground. A flow chart to visualize the main electrical systems is shown in Figure 2.2. The sensors used in the thesis is described in the following section.

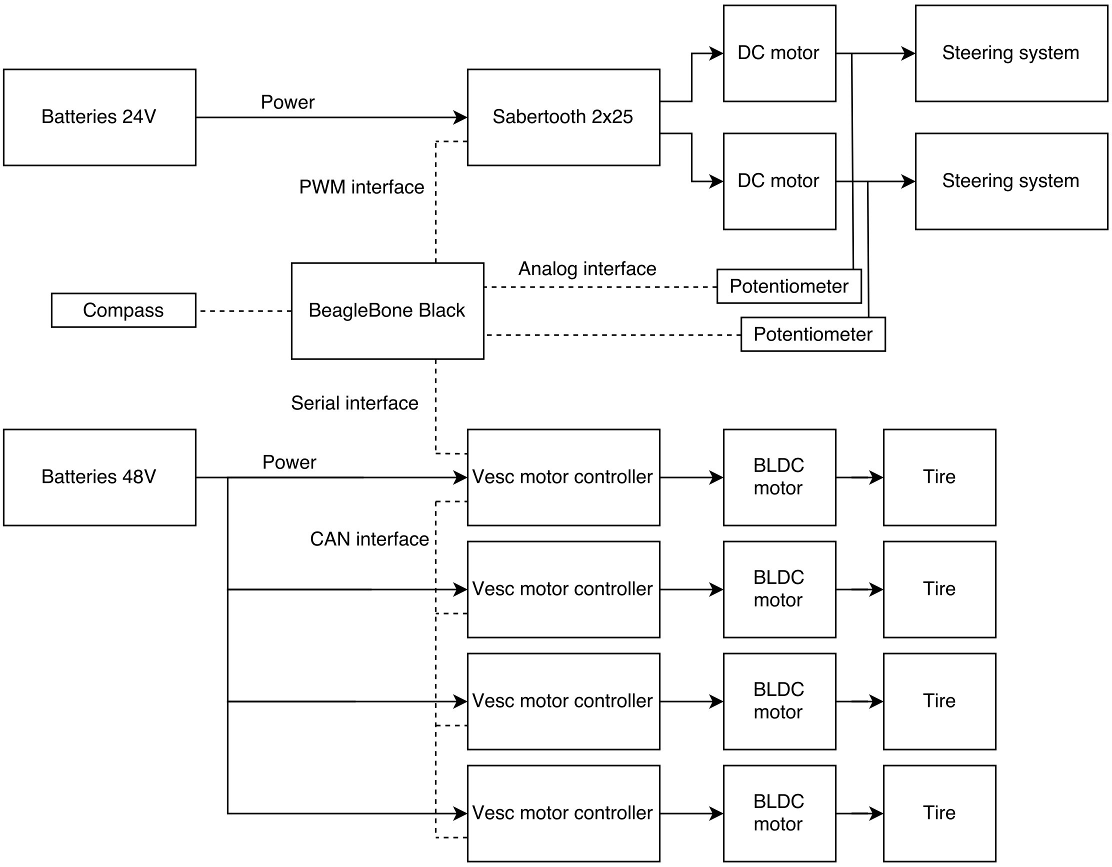


Figure 2.2: *Flow chart of the vehicles main electrical systems. The sensors and the embedded computer i.e. the BeagleBone Black.*

2.3 Sensors

To make it possible to control the vehicle, measurements of the vehicle´s state is required. There is one compass in a form of a magnetometer for measuring the vehicles heading that is around the z-axis, the axis defined for the vehicle can be seen in Figure 5.1. The velocity of the vehicle was estimated with the VESC motor controllers by the use of hall sensors in the motors, see Section 8.2.2.1. The steering angles for the front and rear axle can be estimated by using two potentiometers. These are connected directly to the steering axles and therefore changes the voltage over them when the wheels turn. The voltage can be mapped such that a certain voltage corresponds to a certain steering angle, see Section 8.2.2.3.

2.4 Computer and Software

The test vehicle had an integrated computer to handle all the necessary calculations. The computer was a BeagleBone Black (BBB), it was a one chip community-supported development platform made by BeagleBoard [10]. The platforms processing power allowed it to run the operating system Linux. The electrical systems and the software had an interface on the BBB made in ROS. ROS allowed a system setup that had nodes with code and functions that was executed in parallel. The nodes that built the ROS system could be written in a mix of C++ and Python code.

2.5 Summary of chapter

In this chapter the test vehicle was introduced. The mechanical and electrical hardware of the vehicle was presented together with how the hardware together with the sensors will be used. Finally the computer and software used in the thesis was introduced.

3
Requirements on the closed-loop system

This chapter describes the requirements that exists on the closed-loop driveline control system. The requirements were chosen to make the vehicle behave such that the algorithms that are used earlier in the data flow, see Figure 1.1, can be tested. They have been stated in terms of the performance variables in Figure 3.1.

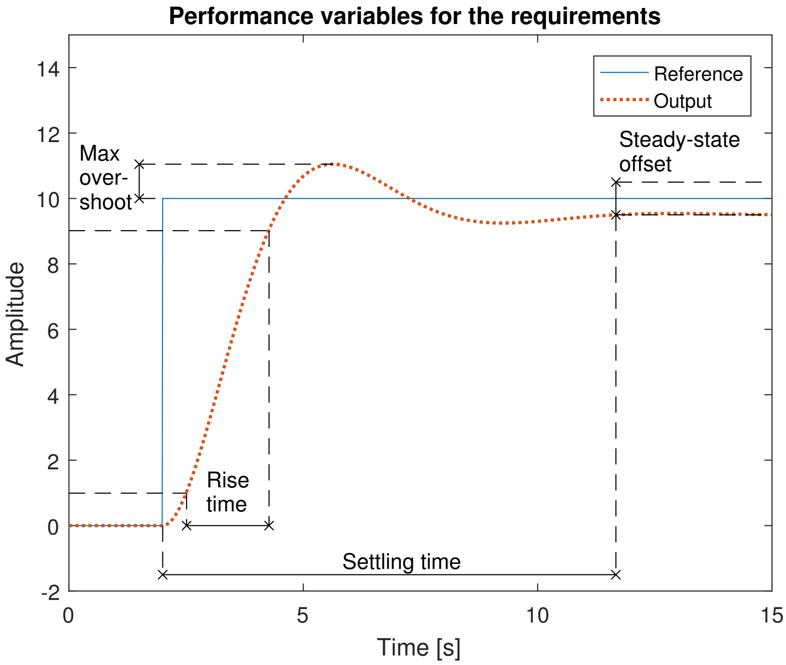


Figure 3.1: *The figure is an illustration of the variables used for setting the closed loop requirements. The rise time is defined as the time it takes for the response to change from 10 to 90 % of the desired value. The settling time is the time it takes for the system to go and stay inside a given offset from the desired value, which is the steady state offset. The maximum overshoot for the response is the amount in percent than the maximum output value is higher than the reference.*

The performance variables are rise time, settling time, maximum overshoot and steady-state offset. They are defined as follows:

- The rise time is the time it takes for the vehicle to change from 10 to 90 % of the reference value.
- The settling time is the time it takes for the vehicle to go from the beginning

of the input step to the first time it enters and thereafter stays inside a given interval of the reference value. Here, the interval was chosen to be same as the steady-state offset.

- The maximum overshoot is how many percent higher the maximum output value is than the reference value.
- The steady state offset is the offset the system have after the settling time.

3.1 Requirements on the longitudinal behaviour

Given that the vehicle partly will be used indoors the maximum velocity of the vehicle should be low for safety reasons, hence a low velocity was used for setting the requirements on the longitudinal behaviour. Given a step in the demanded velocity of 1.5 m/s the longitudinal control system should be able to control the longitudinal behaviour of the vehicle such that:

- The rise time of the velocity is at maximum 3 seconds.
- The settling time of the velocity is at maximum 5 seconds.
- The overshoot of the velocity is at maximum 20 %.
- The steady-state offset should not be more than 10 %.

3.2 Requirements on the lateral behaviour

It is unlikely that the motion planing algorithms would output discrete step references for the heading but this is still useful for setting the requirements on the lateral controller. Given a step in the reference heading of 30° while the vehicle have a velocity at 1.5 m/s the lateral controller should be able to control the lateral behaviour of the vehicle such that:

- The rise time of the heading is at maximum 3 seconds.
- The settling time of the heading is at maximum 5 seconds.
- The overshoot of the heading is at maximum 15°.
- The steady-state offset should not be more than 10°.

3.3 Software compatibility

The designed control system should be made compatible with the existing vehicle, as explained in Chapter 2. The higher level motion planning systems are done in ROS, hence the final controller should be compatible with ROS and therefore needs to be written in C++ or Python code.

3.4 Summary of chapter

In this chapter the requirements on the closed loop system was stated with performance variables. If the requirements are reached it will ensure that the driveline control system performs well enough such that it can follow references that will be fed into it. The requirements was set for the longitudinal and the lateral behaviour as well how the software written in this thesis should be compatible with the higher level motion planning systems.

4
Approach

This section provides an overview of how the driveline control system was designed in order to fulfill the requirements on the closed loop system. It describes why the driveline control system was divided into layers and what the purpose of each layer is, as well as each layers inputs and outputs. Finally, the tests that will be used for verifying that the driveline control system is meeting the requirements are described.

4.1 Control structure

The control structure for over-actuated systems is often separated into several sub-controllers, as can be seen in Figure 4.1. According to [11] the main reason for this is to make the system more modular which can make it easier to change or replace each layer, it also becomes easier to tune the performance of the closed loop system. If two layers are used they are often referred to as high and low level controller and correspondingly a third layer becomes the middle layer that is the control allocation. The reason for using the middle layer is because the system is over actuated, as described in Section 6.2.

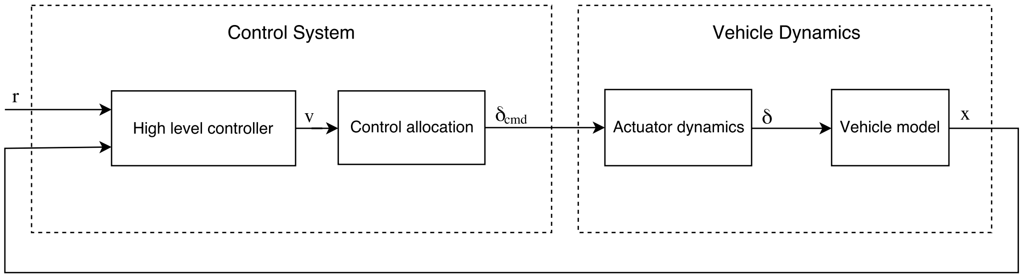


Figure 4.1: *A normal control structure for over-actuated vehicles. The high level controller uses the control signals* $\boldsymbol{r}$ *and the vehicles measured states* $\boldsymbol{x}$ *to calculate the control signals* $\boldsymbol{v}$ *that is fed in the control allocation layer. Here the control signals* $\boldsymbol{\delta}_{cmd}$ *is calculated and directly fed to the actuators. The vehicle´s response is then measured and fed back to the high level controller.*

4.1.1 High level controller

The high level controller is responsible for generating the references for the control allocation, which is the next layer. The inputs are the desired velocity $v_{\mathrm{x,r}}$ and the desired heading θ_r which are put into a vector $\boldsymbol{r}$ and the corresponding measured states $\boldsymbol{x}$

$$\boldsymbol{r} = \begin{bmatrix} v_\mathrm{x,r} & \theta_\mathrm{r} \end{bmatrix}^\mathrm{T} \tag{4.1}$$

$$\boldsymbol{x} = \begin{bmatrix} v_\mathrm{x} & \theta \end{bmatrix}^\mathrm{T} \tag{4.2}$$

The desired states can be generated by a simple remote control or motion planning algorithms, see Figure 1.1. The output vector $\mathbf{v}$ is the virtual command vector and it is the reference for the next layer, it consist of the force in the longitudinal direction as well as the rotational force around the z-axis of the vehicle

$$\mathbf{v} = \begin{bmatrix} F_\mathrm{x} & M_\mathrm{z} \end{bmatrix}^\mathrm{T} \tag{4.3}$$

The reason for converting the velocity and heading errors into forces is that a force is required to change the state of the system according to Newtons first law. It is also conceptually easier to allocate forces instead of variables that can generate forces e.g. current to the motors.

The high level controller was selected to be two Proportional-Integral-Derivative (PID) controllers, one for each reference signal. The reason for this choice is that a PID controller is simple to implement and it is easy to tune the parameters. The performance of a PID controller should be enough for meeting the requirements, this is discussed more in Section 6.1 and evaluated in Section 7.4 and 8.3.

4.1.2 Control allocation

The control allocation is the second layer in the control structure and it uses the forces $\mathbf{v}$, from the high level controller, to calculate the commanded control effort $\boldsymbol{\delta}_\mathrm{cmd}$ for all the actuators such that the desired $\mathbf{v}$ is reached, this is described more in Section 6.2. Here $\boldsymbol{\delta}_\mathrm{cmd}$ is defined as follows

$$\boldsymbol{\delta}_\mathrm{cmd} = \begin{bmatrix} T_\mathrm{fl} & T_\mathrm{fr} & T_\mathrm{rl} & T_\mathrm{rr} & \theta_\mathrm{f} & \theta_\mathrm{r} \end{bmatrix}^\mathrm{T} \tag{4.4}$$

and T_fl, T_fr, T_rl and T_rr is the torque for each wheel and θ_f, θ_r is the front and rear wheel steering angles. One assumption which can be made is to neglect the actuator dynamics

$$\boldsymbol{\delta}_\mathrm{cmd} \approx \boldsymbol{\delta} \tag{4.5}$$

where $\boldsymbol{\delta}$ is the actual control effort of the actuators. The validity of this assumption depends on the dynamics of the actuators, for example electrical motors usually have much faster dynamics than a normal vehicle. If the assumption does not hold the control allocation can include the dynamical behaviour of the actuators, this results in a more complex control allocation algorithm.

The control allocation problem was solved by formulating a quadratic optimization problem (QP), the motivation behind this choice can be seen in Chapter 6 and it is evaluated in Section 7.4 and 8.3.

4.1.3 Low level controller

Instead of including the dynamics of the actuators in the control allocation algorithm, low level controllers can be used to increase the performance such that the assumption in equation 4.5 holds. This yields a control structure which can be seen in Figure 4.2. The low level controller is simply added to the control structure in Figure 4.1 where the type of controller can vary and feedback can be used to enhance the performance.

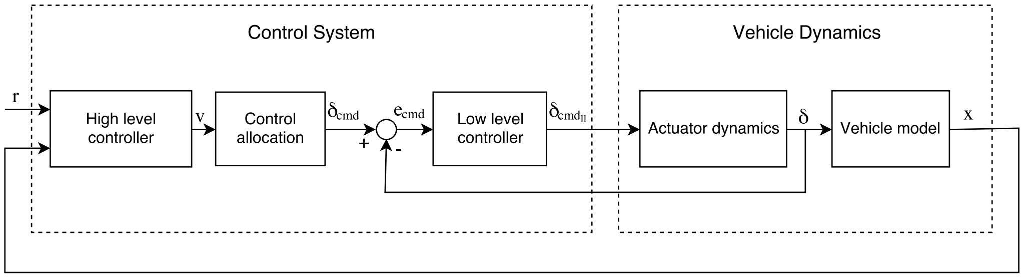


Figure 4.2: *A control structure for an over-actuated vehicle with a low level controller. The high level controller calculates the control signal* $\boldsymbol{v}$ *that is fed in the CA algorithm. This calculates the desired control effort* $\boldsymbol{\delta}$ *which is sent to the low level controller which finally calculates the control signals for the actuators which include the dynamics of the actuators. The vehicle response is then fed back to the high level controller.*

The control allocation layer outputs the desired steering angles for the front and rear axle, hence low level controllers are needed in order to rotate the axles to the desired angles. The controller type were selected to be two PID controllers one for each wheel axle. The main reason for this is that the regulation problem is easy, this is discussed more in Section 6.6 and evaluated in Section 7.4 and 8.3.

4.2 Validation

The final part of this thesis was to test and validate the driveline control system on the vehicle. Hence, this will show if the control structure and the specific controller types are suitable for these types of over-actuated vehicles.

4.2.1 Test scenarios in simulation

The first tests will be made in a simulation environment where the driveline control system is tested on a model of the vehicle. The simulation environment provides an idealized environment where the core behavior of the controller and the vehicle model can be analyzed in an easy way. First two simple step response tests will be made which verifies the functionality of the control allocation algorithm and the low level controller.

1. A step in the longitudinal force F_x will be fed into the control allocation layer. The expected response would be that the control allocation distributes the torques equally over the wheels. If these torques are summed up and divided

by the wheel radius they should match the reference F_{x}. The vehicle model should in turn accelerate the vehicle in the x direction according to Newton's second law.

2. Next the vehicle will travel at a constant velocity and a step in the reference M_{z} will be fed into the control allocation layer. The expected response would be that the control allocation distributes torques more to the outside of the car and to output steering angles different from zero. As described in Section 4.1.3, this reference angles will be reached by using two PID controllers. The vehicle model should in turn drive the vehicle in a smaller and smaller circle as long as the M_{z} does not change sign or return to zero.

After these layers have been verified the high level controller can be included in the tests to verify the closed loop systems performance according to the requirements set in Chapter 3.

1. A step in the reference velocity $v_{\mathrm{x,r}}$ of 1.5 m/s was used to evaluate the system response and to tune the PID parameters in the velocity controller.
2. A step in the reference heading θ_{r} of 30° combined with a constant velocity of 1.5 m/s the vehicles response was evaluated to tune the PID parameters for the heading controller.

With the described tests the stability and the performance of the complete controller can be investigated to see if the closed-loop system meets the requirements.

4.2.2 Test scenarios in the vehicle

The same tests as described in the section above will be carried out in the vehicle. These test will verify the robustness of the controller in a non-idealized environment. In order to evaluate the test scenarios the longitudinal motor torque constant needs to be estimated, the reason for this is that the control allocation outputs the commanded torque for the longitudinal motors while the longitudinal motor controllers receives a commanded current.

1. Torque constant estimation. The motor controllers can be controlled by current references. The current to the motors than can therefor be mapped to the corresponding torque that got out of the wheel. A constant current reference will produce an accelerating force on the vehicle and the relationship can be calculated.

In order to verify that the requirements set for the closed loop system holds, tests of the requirements had to be evaluated in the same way as the tests in the simulation environment as seen above.

4.3 Summary of chapter

In this chapter the approach for the control structure used in this thesis was presented. The control structure was separated into three layers to make the system modular and easy to tune. All layers and the signals between them was further described. Finally the way the validation should be done to show that the requirements holds was described, both for the simulation environment and in the vehicle.

5

Vehicle modelling

This chapter describes how the vehicle was modelled for both the verification of the controller in simulations and for the model based control allocation that is later designed. The chapter starts with a tire model that is a simplified version of the Pajecka's magic formula [12]. Then a bicycle model [13] is used to explain the fundamental behaviour of the vehicle dynamics that was used in the model based control allocation. At last, a four wheel model is derived which was used in the simulations of the driveline control system. First a visualization of how the vehicles axis was defined are shown in Figure 5.1 and the main vehicle parameters are shown in Table 5.1.

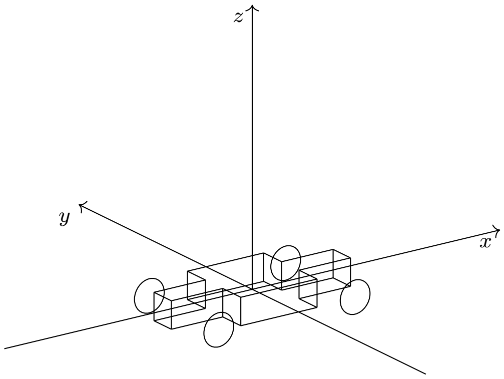


Figure 5.1: *The axis defined for the vehicle, the front is located in the direction of the positive x-axis.*

Table 5.1: *The table shows the main vehicle parameters. All except the inertia have been measured directly by either a measuring tape or a normal body weight scale. COG stands for center of gravity, that is the mass center of an object.*

Description	Symbol	Value	Unit
Mass of vehicle	m	74	kg
Inertia around z-axis	I_z	100	kg/m^2
Wheel radius	r	0.115	m
Track width	t	0.700	m
Distance, front axle to COG	a	0.4975	m
Distance, rear axle to COG	b	0.4975	m

5.1 Tire model

A tire model is essential to explain the complex relationship that exists between the tires and the road. First the most relevant angles are introduced, they are the slip angle, the side-slip angle and the steering wheel angle, these are shown in Figure 5.2.

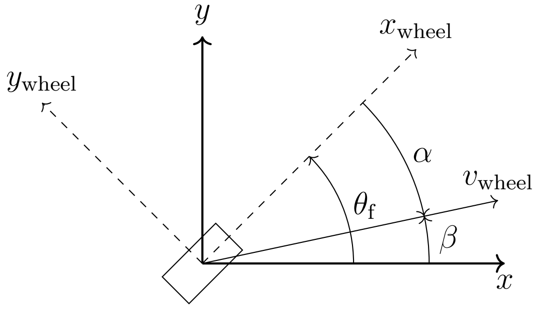


Figure 5.2: *The figure shows how the main angles for a tire was defined. α is the slip angle, β is the side slip angle and θ_{f} is the tires heading angle, in this example approximated to be the steering angle of the front axle. Notice that β and θ_{f} is defined from the vehicles x-axis while α is defined from the tires x-axis.*

The steering angle θ_{f} is the angle between the tires x-axis and the vehicles x-axis, as can be seen in Figure 5.2. The slip angle α is defined as the angle between the tires velocity vector and its x-axis, notice that this angle is usually negative for normal conditions because it is defined from the wheels x-axis. The side-slip angle β is the angle between the vehicles x-axis and the tires velocity vector, hence the following relationship can be established.

$$\alpha = \beta - \theta_{\text{f}} \tag{5.1}$$

The slip angle can also be described in terms by the tires velocity vectors in the x and y the direction

$$\alpha = \arctan\left(\frac{v_{\text{y,wheel}}}{v_{\text{x,wheel}}}\right) \tag{5.2}$$

5.1.1 Pacejka's magic formula

One of the most well known formulas for relating the tire slip angle to the lateral force is the semi-emprical Pajecka's magic formula [12], it exists several versions of it and one of the more simpler one can be seen in equation 5.3. The reason for using a simpler model is that it have fewer parameters than a more advanced models while still giving an accurate estimate. The parameters can often be hard to measure or identify without using special tools or performing experiments which destroys the tires, this was outside the scope the thesis.

$$f_{\text{y}} = D_{\text{y}} \cdot \sin\left(C_{\text{y}} \cdot \arctan\left(B_{\text{y}}\alpha - E_{\text{y}}(B_{\text{y}}\alpha - \arctan(B_{\text{y}}\alpha))\right)\right) \tag{5.3}$$

where B_y, C_y, D_y and E_y can be identified using system identification techniques. This relationship can be simplified further by the small angle approximation

$$\sin(x) = x \quad \text{and} \quad \arctan(x) = x \tag{5.4}$$

this leads to

$$f_\mathrm{y} = D_\mathrm{y} \cdot C_\mathrm{y} \cdot B_\mathrm{y}\alpha - E_\mathrm{y}(B_\mathrm{y}\alpha - B_\mathrm{y}\alpha) = C_\alpha \alpha \tag{5.5}$$

where C_α is the cornering stiffness. This is calculated by the following formula

$$C_\alpha = \mathrm{PKY1} \cdot \mathrm{F}_{\mathrm{z,nominal}} \cdot \sin\left(2 \cdot \tan^{-1}\left(\frac{F_{\mathrm{z}_{ij}}}{\mathrm{PKY2} \cdot \mathrm{F}_{\mathrm{z,nominal}}}\right)\right) \tag{5.6}$$

where the parameters PKY1 and PKY2 in this equation also can be identified by performing system identification techniques. If it is assumed that the side-slip angle β is zero in equation 5.1

$$\alpha = \theta_\mathrm{f} \tag{5.7}$$

equation 5.5 becomes

$$f_\mathrm{y} = C_\alpha \theta_\mathrm{f} \tag{5.8}$$

The small angle approximation and the assumption that the side-slip angle $\beta = 0$ was done in [14]. Figure 5.3 shows the linear approximation described above for the vehicle. The tire parameters was taken from [15] while the mass of the vehicle have been measured.

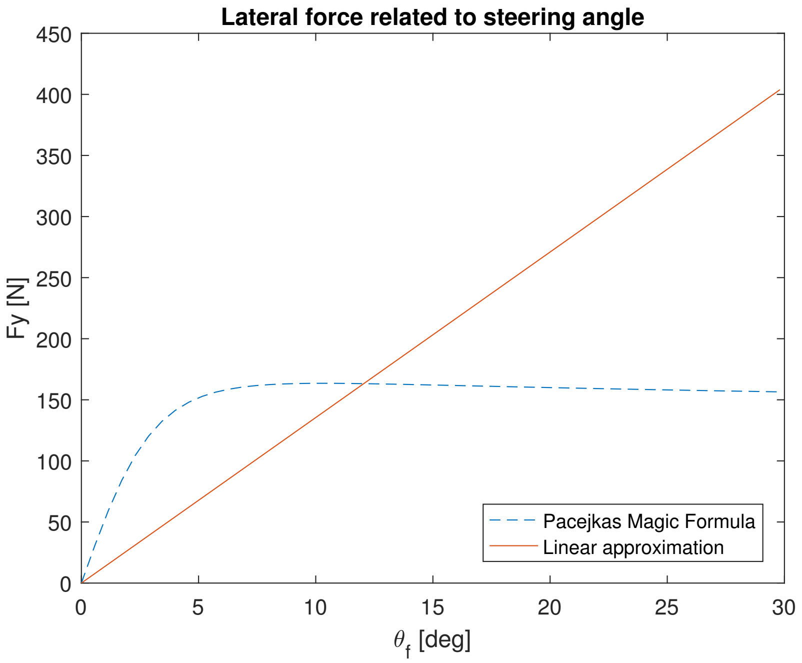


Figure 5.3: *The figure shows the linear function that was used to approximate the magic formula. The only parameter which was from the vehicle was the mass which affect the maximum height of the curve. The other parameters was taken from [15].*

5.2 Vehicle models

A vehicle model can be designed to map all the wheel forces to the total forces of the vehicle, which can be used to calculate the motions of the vehicle. The motions and dynamics can be expressed in various ways. Advanced methods often includes the complex behaviours of the vehicle, for example the traction forces generated by the tires from different slip ratios, weight transfer by the chassis when traveling through corners, spring travel and more.
Simpler models can be designed that uses a 2D-planar motion model. These models uses the basic dynamics of the vehicles but they lack the more complex dynamics. However for control design a simple model which captures the fundamental behavior is often enough, especially when traveling at lower velocities.

5.2.1 Bicycle model

One of the simpler vehicle models is the bicycle model, see Figure 5.4. A bicycle model maps the wheels forces and dynamics to a single track model. One advantage of using a bicycle models is that is computationally lighter compared to more complex models while it still captures the basic dynamics of the vehicle. A well designed controller can also be robust enough to handle any differences between the model and reality. One of the downsides with a bicycle model is that a controller that is based on a bicycle model cannot directly control all actuators uniquely. For example in a in a four wheel drive (4WD) vehicle the left and right motors on the same axle would be given the same demanded torque.

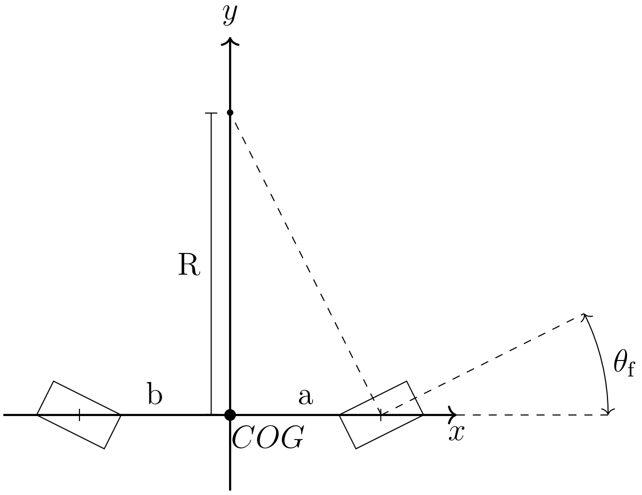


Figure 5.4: *A bicycle model of the vehicle with steering on both the front and rear axle. For a bicycle model it is assumed that the front steering angle θ_f is the average of the individual front wheel angles.*

The longitudinal force for the two wheels in the wheel´s local coordinate frame can be calculated by

$$f_{\mathrm{x}_i} = \frac{T_i}{r}, \quad i \in [f, r] \tag{5.9}$$

The longitudinal forces generated by the wheel torques T_i and the lateral forces, see equations 5.5 and 5.9 respectively, can be mapped from the wheels coordinate systems to the total force acting on the vehicle in the longitudinal and lateral direction, see equations 5.10-5.11.

$$\begin{aligned} F_{\mathrm{x}} = & f_{\mathrm{xf}} \cos(\theta_{\mathrm{f}}) + f_{\mathrm{x_r}} \cos(\theta_{\mathrm{r}}) \\ & - f_{\mathrm{y_f}} \sin(\theta_{\mathrm{f}}) - f_{\mathrm{y_r}} \sin(\theta_{\mathrm{r}}) \end{aligned} \tag{5.10}$$

$$\begin{aligned} F_{\mathrm{y}} = & f_{\mathrm{xf}} \sin(\theta_{\mathrm{f}}) + f_{\mathrm{x_r}} \sin(\theta_{\mathrm{r}}) \\ & + f_{\mathrm{y_f}} \cos(\theta_{\mathrm{f}}) + f_{\mathrm{y_r}} \cos(\theta_{\mathrm{r}}) \end{aligned} \tag{5.11}$$

Similar, the resulting rotational force acting on the vehicle generated by the wheel forces can be expressed as in equation 5.12 using the vehicles geometry where a and b are the levers.

$$\begin{aligned} M_{\mathrm{z}} = & \left(f_{y_{\mathrm{f}}} \cos(\theta_{\mathrm{f}}) + f_{x_{\mathrm{f}}} \sin(\theta_{\mathrm{f}})\right) a \\ & - \left(f_{y_{\mathrm{r}}} \cos(\theta_{\mathrm{r}}) + f_{x_{\mathrm{r}}} \sin(\theta_{\mathrm{r}})\right) b \end{aligned} \tag{5.12}$$

From the global linear and rotational forces, the equation of motion in the x-axis can be calculated as

$$a_{\mathrm{x}} = \frac{F_{\mathrm{x}}}{m} - c_{\mathrm{wind}} v_x^2 - c_{\mathrm{roll}} v_{\mathrm{x}} \tag{5.13}$$

where the constants c_{wind} and c_{roll} can be experimentally identified. The lateral and rotational behaviour of the vehicle and how these affect each other can be seen in equation 5.14, this was derived in [13].

$$\begin{bmatrix} a_{\mathrm{y}} \\ \ddot{\theta} \end{bmatrix} = \begin{bmatrix} -\frac{C_\alpha + C_\alpha}{m v_{\mathrm{x}}} & \frac{b C_\alpha - a C_\alpha}{m v_{\mathrm{x}}} - v_{\mathrm{x}} \\ \frac{b C_\alpha - a C_\alpha}{I_{\mathrm{z}} v_{\mathrm{x}}} & -\frac{b^2 C_\alpha + a^2 C_\alpha}{I_{\mathrm{z}} v_{\mathrm{x}}} \end{bmatrix} \begin{bmatrix} v_{\mathrm{y}} \\ \dot{\theta} \end{bmatrix} + \begin{bmatrix} \frac{1}{m} & 0 \\ 0 & \frac{1}{I_{\mathrm{z}}} \end{bmatrix} \begin{bmatrix} F_{\mathrm{y}} \\ M_{\mathrm{z}} \end{bmatrix} \tag{5.14}$$

5.2.2 Four wheel model

For complex vehicle systems to be used for verification in simulations there is often a need to model the dynamics of all the actuators, which makes bicycle models unsuited. Ackermann steering [7] which was used in the vehicle works by turning the wheels around a common point. This implies that the wheels have different distances to this point. Hence, the wheels need to be turned by different amounts on the left and the right side of the vehicle, see Figure 5.5. In the figure, R is the distance from the Center of Gravity (COG) to the turning point and θ_{fl} and θ_{fr} are the turning angles of the front wheels. In the test vehicle the Ackermann steering is implemented by a physical linkage connecting the wheels on the same axis, as can be seen in Figure 5.6. The steering angle θ_{f} defined as an imaginary wheel on the corresponds to the mean angle of the two front wheels. It is the same as the angle θ_{f} in the bicycle model, see Figure 5.4.

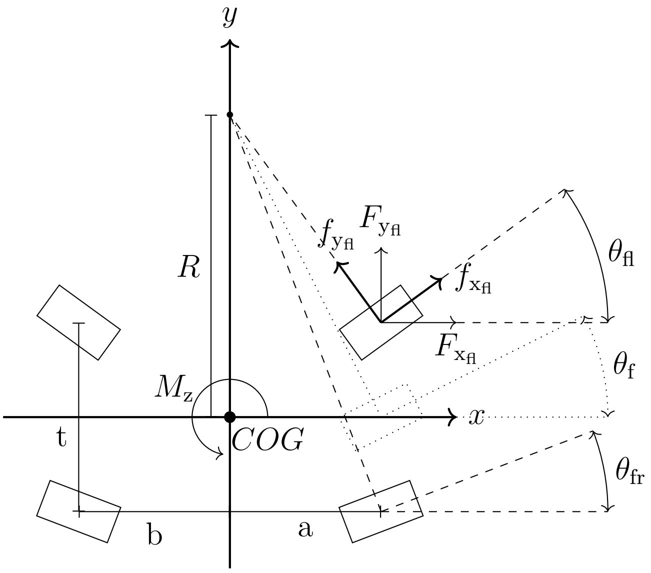


Figure 5.5: *Model over vehicle, notice the different wheel angles θ_{fl} and θ_{fr}. The dotted wheel have the mean angle of the two front wheels and it corresponds to the two front wheels in a bicycle model.*

Therefore it is possible to only calculate one steering reference angle $\theta_{\mathrm{f,r}}$ for the imaginary wheel on the front axle. From this angle the resulting steering angles for the left and right wheels can be calculated by the equations 5.15 and 5.16a-5.16d, where the radius R is assumed to be the same for the front and the rear axle.

$$R = a\cot(\theta_{\mathrm{f,r}}) \tag{5.15}$$

$$\theta_{\mathrm{fl}} = \tan^{-1}\left(\frac{a}{R-\frac{1}{2}t}\right) \tag{5.16a}$$

$$\theta_{\mathrm{fr}} = \tan^{-1}\left(\frac{a}{R+\frac{1}{2}t}\right) \tag{5.16b}$$

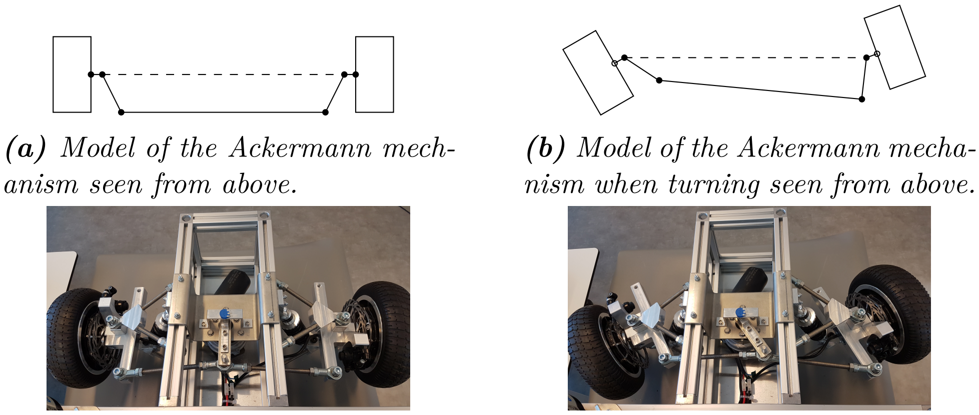
$$\theta_{\mathrm{rl}} = -\tan^{-1}\left(\frac{b}{R-\frac{1}{2}t}\right) \tag{5.16c}$$

$$\theta_{\mathrm{rr}} = -\tan^{-1}\left(\frac{b}{R+\frac{1}{2}t}\right) \tag{5.16d}$$

The longitudinal force for each wheel in the wheel´s local coordinate frame can be calculated by

$$f_{\mathrm{x}_{ij}} = \frac{T_{ij}}{r}, \quad i \in [f,r],\ j \in [l,r] \tag{5.17}$$

The longitudinal forces generated by the wheel torques T_{ij} and the lateral forces, see equations 5.17 and 5.5 respectively, can be mapped from the wheels coordinate systems to the total force acting on the vehicle in the longitudinal and lateral direction, see equations 5.18-5.19. This is an extension of the longitudinal forces from



(a) *Model of the Ackermann mechanism seen from above.*

(b) *Model of the Ackermann mechanism when turning seen from above.*

(c) *Picture of the Ackermann mechanism on the vehicle seen from above.*

(d) *Picture of the Ackermann mechanism on the vehicle when turning seen from above.*

Figure 5.6: *Model of the Ackermann steering mechanism and picture of the same system implemented in the vehicle, comparison is made both when there the steering angle is* 0 *and when there is a positive one.*

the bicycle model, described in equations 5.10 - 5.11, to include the acting forces from all the four wheels.

$$\begin{aligned}F_{\mathrm{x}} =& f_{\mathrm{xfl}}\cos(\theta_{\mathrm{fl}}) + f_{\mathrm{xfr}}\cos(\theta_{\mathrm{fr}}) + f_{\mathrm{xrl}}\cos(\theta_{\mathrm{rl}}) + f_{\mathrm{xrr}}\cos(\theta_{\mathrm{rr}}) \\ &- f_{\mathrm{yfl}}\sin(\theta_{\mathrm{fl}}) - f_{\mathrm{yfr}}\sin(\theta_{\mathrm{fr}}) - f_{\mathrm{yrl}}\sin(\theta_{\mathrm{rl}}) - f_{\mathrm{yrr}}\sin(\theta_{\mathrm{rr}})\end{aligned} \tag{5.18}$$

$$\begin{aligned}F_{\mathrm{y}} =& f_{\mathrm{xfl}}\sin(\theta_{\mathrm{fl}}) + f_{\mathrm{xfr}}\sin(\theta_{\mathrm{fr}}) + f_{\mathrm{xrl}}\sin(\theta_{\mathrm{rl}}) + f_{\mathrm{xrr}}\sin(\theta_{\mathrm{rr}}) \\ &+ f_{\mathrm{yfl}}\cos(\theta_{\mathrm{fl}}) + f_{\mathrm{yfr}}\cos(\theta_{\mathrm{fr}}) + f_{\mathrm{yrl}}\cos(\theta_{\mathrm{rl}}) + f_{\mathrm{yrr}}\cos(\theta_{rr})\end{aligned} \tag{5.19}$$

Similar to equation 5.12, the resulting rotational force acting on the vehicle generated by the individual wheel forces can be expressed as in equation 5.20 using the vehicles geometry where $t/2$, a and b are the levers.

$$\begin{aligned}M_{\mathrm{z}} =& \frac{\left(-f_{x_{\mathrm{fl}}}\cos(\theta_{\mathrm{fl}}) + f_{y_{\mathrm{fl}}}\sin(\theta_{\mathrm{fl}})\right)t}{2} + \left(f_{y_{\mathrm{fl}}}\cos(\theta_{\mathrm{fl}}) + f_{x_{\mathrm{fl}}}\sin(\theta_{\mathrm{fl}})\right)a + \\ & \frac{\left(f_{x_{\mathrm{fr}}}\cos(\theta_{\mathrm{fr}}) - f_{y_{\mathrm{fr}}}\sin(\theta_{\mathrm{fr}})\right)t}{2} + \left(f_{y_{\mathrm{fr}}}\cos(\theta_{\mathrm{fr}}) + f_{x_{\mathrm{fr}}}\sin(\theta_{\mathrm{fr}})\right)a + \\ & \frac{\left(-f_{x_{\mathrm{rl}}}\cos(\theta_{\mathrm{rl}}) + f_{y_{\mathrm{rl}}}\sin(\theta_{\mathrm{rl}})\right)t}{2} - \left(f_{y_{\mathrm{rl}}}\cos(\theta_{\mathrm{rl}}) + f_{x_{\mathrm{rl}}}\sin(\theta_{\mathrm{rl}})\right)b + \\ & \frac{\left(f_{x_{\mathrm{rr}}}\cos(\theta_{\mathrm{rr}}) - f_{y_{\mathrm{rr}}}\sin(\theta_{\mathrm{rr}})\right)t}{2} - \left(f_{y_{\mathrm{rr}}}\cos(\theta_{\mathrm{rr}}) + f_{x_{\mathrm{rr}}}\sin(\theta_{\mathrm{rr}})\right)b\end{aligned} \tag{5.20}$$

From the global linear and rotational forces, the equation of motion in the x-axis can be calculated as

$$a_{\mathrm{x}} = \frac{F_{\mathrm{x}}}{m} - c_{\mathrm{wind}}v_{\mathrm{x}}^2 - c_{\mathrm{roll}}v_{\mathrm{x}} \tag{5.21}$$

where the constants c_{wind} and c_{roll} can be experimentally identified. The lateral and rotational behaviour of the vehicle and how these affect each other can be seen in the equation below, this was derived in [13].

$$\begin{bmatrix} a_{\mathrm{y}} \\ \ddot{\theta} \end{bmatrix} = \begin{bmatrix} -\frac{C_\alpha + C_\alpha}{m v_{\mathrm{x}}} & \frac{bC_\alpha - aC_\alpha}{m v_{\mathrm{x}}} - v_{\mathrm{x}} \\ \frac{bC_\alpha - aC_\alpha}{I_{\mathrm{z}} v_{\mathrm{x}}} & -\frac{b^2 C_\alpha + a^2 C_\alpha}{I_{\mathrm{z}} v_{\mathrm{x}}} \end{bmatrix} \begin{bmatrix} v_{\mathrm{y}} \\ \dot{\theta} \end{bmatrix} + \begin{bmatrix} \frac{1}{m} & 0 \\ 0 & \frac{1}{I_{\mathrm{z}}} \end{bmatrix} \begin{bmatrix} F_{\mathrm{y}} \\ M_{\mathrm{z}} \end{bmatrix} \tag{5.22}$$

5.3 Summary of chapter

In this chapter the vehicle modelling was made, first the use of tire models was described and a simplified version of Pacejka's magic formula was derived. Then a bicycle model was derived to easily describe a vehicles motion in a 2D environment, this is later useful in the control allocation algorithm. Similarly a four wheel model that uses the Ackermann steering systems was derived that should be used for validation of the control allocation in the simulation environment.

6
Control design

Given the knowledge on how each actuator affects the vehicle, it is possible to determine how much each actuator should contribute with in order to reach the desired states. As described in Chapter 4 the driveline control system was separated into several layers mainly to make it more modular and easy to tune. Hence, this chapter explains how each layer was designed with a focus on the control allocation layer. The chapter starts with a description and design motivation of the high level controller before a background in control allocation is presented. Then the core concept of several control allocation algorithms are presented and the final choice of algorithm is motivated. The variables for the chosen algorithm are then defined to suite the vehicle. Finally the low level controllers are presented and the types are motivated.

6.1 High level controller

As described in Section 4.1.1, the high level controller generates the virtual command $\boldsymbol{v}$ that the control allocation uses to determine the demanded control effort. There are several different controller types that can generate this command, one of the simplest controller is a proportional controller which calculates a control signal proportional to the error between the reference and the measured values. However, this can result in a steady-state offset, this can be removed by adding a integral part, which integrates the error between the reference and the measured values. The integral part grows as long as the error exists which results in a higher and higher control signal until the error is removed. The derivative part of a PID controller can improve the settling time of a system, decrease or completely remove large overshoots and it can make the system more stable. However, the derivative gain is sensitive to noise and this should be taken into account while determining the size of it. The PID controller is formulated as shown in equation 6.3.

$$u(t) = K_{\mathrm{p}}e(t) + K_{\mathrm{i}}\int_0^t e(\tau)\mathrm{d}\tau + K_{\mathrm{d}}\frac{\mathrm{d}}{\mathrm{dt}}e(t) \tag{6.1}$$

where $e(t)$ is the difference between the reference signal $r(t)$ and the measured value $y(t)$. The K_{p}, K_{i} and K_{d} is called the proportional, integral and derivative gain respectively. These will be tuned by first only using the proportional gain until the desired response time is achieved if it is possible and also to get as low steady-state offset as possible without yielding a big overshoot. After this the integral gain can be tuned to remove the steady-state offset while keeping it low enough to avoid oscillations. If a large overshoot exist the derivative gain can be used to decrease it and also to decrease or completely remove oscillations.

A PID controller can only handle one minimization problem, hence given that the driveline control system is should follow both a velocity and a heading reference two PID controllers are needed. This result in the following two equations for calculating the virtual command

$$F_\mathrm{x}(t) = K_\mathrm{p}e(t) + K_\mathrm{i}\int_0^t e(\tau)\mathrm{d}\tau + K_\mathrm{d}\frac{\mathrm{d}}{\mathrm{dt}}e(t) \quad \text{where} \quad e(t) = v_\mathrm{x,r} - v_\mathrm{x} \tag{6.2}$$

$$M_\mathrm{z}(t) = K_\mathrm{p}e(t) + K_\mathrm{i}\int_0^t e(\tau)\mathrm{d}\tau + K_\mathrm{d}\frac{\mathrm{d}}{\mathrm{dt}}e(t) \quad \text{where} \quad e(t) = \theta_\mathrm{r} - \hat{\theta} \tag{6.3}$$

6.2 Background in control allocation

An over-actuated system is a system in which there are more actuators than there are degrees of freedom, hence the distribution of the efforts among the actuators can become non-trivial. One simple example of this is two people sliding a box over a surface in one direction, they must allocate how much force they contribute with in order to achieve a desired behavior of the box, for example the velocity of the box. There are infinitely many solutions to this problem, as they can distribute the forces in infinitely many ways.

Control allocation was discovered at the same time as the first over-actuated systems were introduced. The first control allocation method often consisted of locking together different actuators, hence the actuators effort was divided into fix ratios. Another early method was to use the actuators in a specific sequence. Combination of these two methods were also often used. Today's car industry often use rule based control allocation [16], it implies that the allocation is done by a specific set of rules. These kind of algorithms are called static control allocation by some authors [17] or direct control allocation by others [18] in which case static refers to something else. In this thesis they have been called simple control allocation algorithms. These requires little computational power but often requires long tuning periods of the thresholds in order to attain good performance and they cannot adapt the ratios, sequences or rules online which can limit the performance and robustness.

The increase in computational power have open up the possibility to use more advanced control allocation methods. This started out in the aerospace industry, where over-actuated aircraft could have redundancy in the case of actuator failure. Advanced methods can also provide the possibility to choose the best combination of actuators with respect to some performance index, for example energy efficiency [19], [20] or system response time. The algorithms which uses some kind of performance index are based on optimization techniques. They require more computational power than simple control allocation methods but can often results in better performance.

6.3 Control allocation algorithms

The goal of all control allocation algorithms are to determine the commanded control effort $\boldsymbol{\delta}_\mathrm{cmd} \in \mathbb{R}^{m\times 1}$ such that the combination of all actuators effort reaches the desired effort $\boldsymbol{v} \in \mathbb{R}^{k\times 1}$. How each actuator effect the system is described in the effectiveness matrix $\boldsymbol{B} \in \mathbb{R}^{k\times m}$. Hence, this can be formulated as

$$\mathbf{B}\boldsymbol{\delta}_{\text{cmd}} = \mathbf{v}. \tag{6.4}$$

which should be solved for $\boldsymbol{\delta}_{\text{cmd}}$. For some algorithms the solution can be subjected to constraints on the actuators

$$\underline{\boldsymbol{\delta}} \leq \boldsymbol{\delta}_{\text{cmd}} \leq \overline{\boldsymbol{\delta}}. \tag{6.5}$$

The most simple case is when $\mathbf{B}$ is squared $k = m$ and non-singular. It is then possible to multiply both sides of equation 6.4 by the inverse of $\mathbf{B}$ to yield

$$\boldsymbol{\delta}_{\text{cmd}} = \mathbf{B}^{-1}\mathbf{v}. \tag{6.6}$$

For $\mathbf{B}$ to be squared there must be the same number of actuators as there are references and this means that the system is not over-actuated.

6.3.1 Explicit ganging

Explicit ganging is a simple method to decrease the number of unique commanded control efforts that shall be calculated. The method works by locking together the actuators that contribute to the same effort. Hence, this decreases the number of rows in the vector $\boldsymbol{\delta}_{\text{cmd}}$ and the number of columns in the effectiveness matrix $\boldsymbol{B}$. If enough actuators are coupled like this, $\boldsymbol{B}$ becomes square and equation 6.4 can be solved directly as shown above. For a vehicle with similar configuration as the test vehicle, described in Section 2, all longitudinal motors could for example be coupled to give the same control effort. In a similar way the front and the rear steering wheel axle could have the same angle but with opposite signs. This would result in

$$\mathbf{B}\begin{bmatrix} T_{\text{wheels}} \\ \theta_{\text{steering}} \end{bmatrix} = \begin{bmatrix} F_{\text{x}} \\ \theta_{\text{r}} \end{bmatrix} \tag{6.7}$$

which can be solved directly as the section above describes. After T_{wheels} and θ_{steering} have been determined the individual torques $T_{ij,r}$ and steering angles can be calculated directly, for example

$$T_{\text{fl,r}} = T_{\text{fr,r}} = T_{\text{rl,r}} = T_{\text{rr,l}} = \frac{1}{4}T_{\text{wheels}} \qquad \theta_{\text{f,r}} = -\theta_{\text{r,r}} = \frac{1}{2}\theta_{\text{steering}}. \tag{6.8}$$

6.3.2 Daisy chaining

Another simple algorithm is daisy chaining, this is an hierarchical method where the actuators are used in a predetermined order, see Figure 6.1. First the demanded control effort for the first actuator $\delta_{\text{cmd},1}$ is calculated, if this is below the saturation limit for the actuator no more actuators needs to be used. However, if it is above the actuator´s saturation limit there will be a remaining error. This effort error e_1 is sent to the next controller, where in a similar way it calculates the demanded control effort $\delta_{\text{cmd},2}$ and if the actuator does not saturate no more effort is needed,

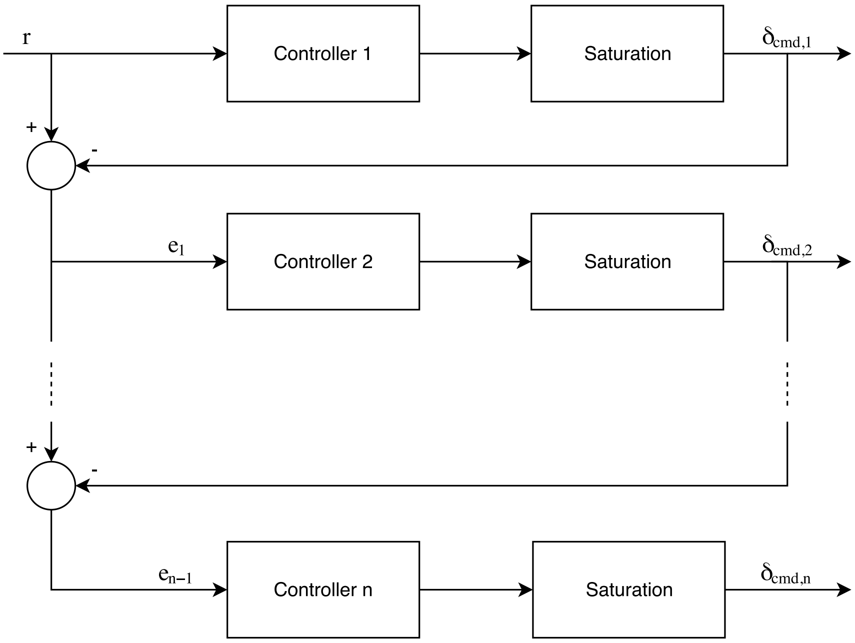


Figure 6.1: *The general structure of a daisy chaining control allocation algorithm. Notice that the controllers have a fixed order, which is determined beforehand. Hence, the order of which actuator is used is also determined beforehand.*

otherwise the remaining error is sent to the next controller. This process is repeated until the desired effort have been reached or all actuators have been saturated.
An example for a 4WD vehicle is that first are the rear motors effort is calculated and if they saturates the effort for the front wheels are calculated e.g.

$$F_{\mathrm{x,r}} = 1.5 \quad \text{and} \quad \boldsymbol{\delta}_{\mathrm{cmd}} \leq \overline{\boldsymbol{\delta}}_{\mathrm{cmd}} = 1 \tag{6.9}$$

hence the commanded control effort will be

$$\boldsymbol{\delta}_{\mathrm{cmd,\ rear\ wheels}} = 1 \quad \text{and} \quad \boldsymbol{\delta}_{\mathrm{cmd,\ front\ wheels}} = 0.5. \tag{6.10}$$

6.3.3 Pseudo Inverse

When $\boldsymbol{B}$ is non-square i.e. $k \neq m$ $\boldsymbol{B}$ cannot be inverted but one can calculate the pseudo inverse, which is defined as

$$\boldsymbol{B}^{\dagger} = \boldsymbol{B}^{T}(\boldsymbol{B}\boldsymbol{B}^{T})^{-1}. \tag{6.11}$$

This can be used to get $\boldsymbol{\delta}_{\mathrm{cmd}}$

$$\boldsymbol{\delta}_{\mathrm{cmd}} = \boldsymbol{B}^{T}(\boldsymbol{B}\boldsymbol{B}^{T})^{-1}\boldsymbol{v} = \boldsymbol{B}^{\dagger}\boldsymbol{v} \tag{6.12}$$

As shown in [21], this is also the solution to the following optimization problem

$$\min_{\boldsymbol{\delta}_{\mathrm{cmd}}} \frac{1}{2}\boldsymbol{\delta}_{\mathrm{cmd}}^{T}\boldsymbol{\delta}_{\mathrm{cmd}} \tag{6.13}$$

subject to the constraint

$$\boldsymbol{B}\boldsymbol{\delta}_{\mathrm{cmd}} = \boldsymbol{v} \tag{6.14}$$

It was solved by formulating the Hamiltonian but this is out of this thesis scope. This method also supports to weight the effort from the actuators by using the weighted pseudo inverse $\boldsymbol{B}^{\#}$

$$\boldsymbol{B}^{\#} = \boldsymbol{W}^{-1}\boldsymbol{B}^{T}(\boldsymbol{B}\boldsymbol{W}^{-1}\boldsymbol{B}^{T})^{-1} \tag{6.15}$$

where $W \in \mathbb{R}^{m \times m}$ is filled with weights in the diagonal for penalizing the use of the actuators. Hence, some actuators effort can be better to use and therefore given a smaller weight. This leads to the following expression for determining $\boldsymbol{\delta}_{\text{cmd}}$

$$\boldsymbol{\delta}_{\text{cmd}} = \boldsymbol{W}^{-1}\boldsymbol{B}^{T}(\boldsymbol{B}\boldsymbol{W}^{-1}\boldsymbol{B}^{T})^{-1}\boldsymbol{v} = \boldsymbol{B}^{\#}\boldsymbol{v} \tag{6.16}$$

This is the solution to an extension of the optimization problem in equation 6.13

$$\min_{\boldsymbol{\delta}_{\text{cmd}}} \frac{1}{2}\boldsymbol{\delta}_{\text{cmd}}^{T}\boldsymbol{W}\boldsymbol{\delta}_{\text{cmd}} \tag{6.17}$$

The pseudo inverse does not directly support constraints on the actuators but it can be extended to include this, which can be seen in [21].

6.3.4 Linear Programming

A more direct algorithm to handle constraints on the actuators are linear programming (LP). Here the basic problem is formulated as

$$\min_{\boldsymbol{\delta}_{\text{cmd}}} ||\boldsymbol{B}\boldsymbol{\delta}_{\text{cmd}} - \boldsymbol{v}||_1 \tag{6.18}$$

subject to

$$\underline{\boldsymbol{\delta}} \leq \boldsymbol{\delta}_{\text{cmd}} \leq \overline{\boldsymbol{\delta}}. \tag{6.19}$$

This is solved by reformulating it to a standard LP problem formulation and using a standard LP solver. If no actuators are saturated the equation $\boldsymbol{B}\boldsymbol{\delta}_{cmd} = \boldsymbol{v}$ holds and infinitely many solutions exists. Therefore, this problem formulation is often extended to also minimize the difference between the commanded and the desired control effort $\boldsymbol{\delta}_{\text{d}} \in R^{m \times 1}$. Hence the problem formulation becomes

$$\min_{\boldsymbol{\delta}_{\text{cmd}}} ||\boldsymbol{\delta}_{\text{cmd}} - \boldsymbol{\delta}_{\text{d}}||_1 + \gamma||\mathrm{B}\boldsymbol{\delta}_{\text{cmd}} - \boldsymbol{v}||_1 \tag{6.20}$$

subject to

$$\underline{\boldsymbol{\delta}} \leq \boldsymbol{\delta}_{\text{cmd}} \leq \overline{\boldsymbol{\delta}}. \tag{6.21}$$

Here γ is a weighting factor which enables the ability to choose the relative importance of the two terms. This method can also be extended to include weights on both terms in equation 6.20, this leads to

$$\min_{\boldsymbol{\delta}_{\text{cmd}}} ||\boldsymbol{\delta}_{\text{cmd}} - \boldsymbol{\delta}_{\text{d}}||_{1,\boldsymbol{W}_{\boldsymbol{\delta}}} + \gamma||\mathrm{B}\boldsymbol{\delta}_{\text{cmd}} - \boldsymbol{v}||_{1,\boldsymbol{W}_{\text{v}}} \tag{6.22}$$

subject to

$$\underline{\boldsymbol{\delta}} \leq \boldsymbol{\delta}_{\text{cmd}} \leq \overline{\boldsymbol{\delta}}. \tag{6.23}$$

Here, the weighting matrix W_{δ} is a diagonal matrix were a weight or cost is specified for the actuators effort. The weighting matrix W_{v} have one weight for each entry in the virtual command vector $\boldsymbol{v}$, hence there relative importance can be specified.

6.3.5 Quadratic Programming

In a similar way as in LP the QP approach uses almost the same formulation for the optimization problem but it uses the 2-norm instead of the 1-norm, hence

$$\min_{\boldsymbol{\delta}_{\mathrm{cmd}}} ||\boldsymbol{\delta}_{\mathrm{cmd}} - \boldsymbol{\delta}_{\mathrm{d}}||^2_{2,\boldsymbol{W}_{\boldsymbol{\delta}}} + \gamma||\mathbf{B}\boldsymbol{\delta}_{\mathrm{cmd}} - \boldsymbol{v}||^2_{2,\boldsymbol{W}_{\mathrm{v}}} \tag{6.24}$$

subject to

$$\underline{\boldsymbol{\delta}} \leq \boldsymbol{\delta}_{\mathrm{cmd}} \leq \overline{\boldsymbol{\delta}}. \tag{6.25}$$

When no actuator is saturated this gives exactly the same result as the pseudo inverse method which is shown in [21]. There is a solver [22], which solves the problem stated in equations 6.24 and 6.25 without the need to reformulate it on standard QP form.

6.3.6 Model Predictive Control Allocation

The model predictive control (MPC) scheme can be used to solve the control allocation problem, this is essentially an extension of the quadratic programming approach where the behaviour of the system in the future is predicted. This is called model predictive control allocation (MPCA). Hence, the algorithm can predict that the system will reach the constraints on a certain actuator and change the demanded control effort. It is formulated as the following

$$\min_{\boldsymbol{\delta}_{\mathrm{cmd}}} \left(\sum_{n=1}^{N} ||\boldsymbol{\delta}_{\mathrm{cmd}}(n) - \boldsymbol{\delta}_{\mathrm{d}}||^2_{2,\boldsymbol{W}_{\boldsymbol{\delta}}} + \gamma \sum_{n=1}^{N} ||\mathbf{B}\boldsymbol{\delta}_{\mathrm{cmd}}(n) - \boldsymbol{v}||^2_{2,\boldsymbol{W}_{\mathrm{v}}} \right) \tag{6.26}$$

subject to the dynamics and the constraints of the actuators. N is the prediction horizon and n the time instant. Only the control effort for the first time instant are applied to the system before everything is recomputed at the next time instant.

6.4 Selection of control allocation algorithm

Given the available computing power more advanced algorithms could be used than explicit ganging or daisy chaining, this would give better performance. Linear programming and quadratic programming are more flexible than pseudo inverse control when it comes to deciding the balance between minimizing the error or the control effort. The difference in computation time between LP and QP solvers are not too big with modern solvers, and QP is a second order approximation of nonlinearities while LP is only a first order approximation. The increase in cost for the errors increases quadratically instead of linearly therefore should the actuators reach there limits later compared to LQ.

An investigation, [14], was made were an MPCA algorithm were implemented on a full-size truck and tested on split-μ surfaces. The MPCA algorithm gave better result than a corresponding QP solution when the system was pushed close to it limits but required a lot of computational power. The advantages of using MPCA are also deemed to be small given that vehicle will be driving indoors which means that the road friction will be high, at low velocities and the accelerations should be kept low. Hence, a QP control allocation scheme will investigated in this thesis.

6.5 Control allocation with quadratic programming

This section goes through how the quadratic programming approach was formulated to solve the control allocation problem by using an existing CA solver, made by Ola Härkegård [22]. This solver was the Control Allocation Toolbox (QCAT) and it was based on QP control allocation methods. It calculates the commanded control effort and the virtual command $\boldsymbol{v}$ can be seen below

$$\boldsymbol{v} = \begin{bmatrix} F_{\mathrm{x}} \\ M_{\mathrm{z}} \end{bmatrix} \tag{6.27}$$

First, the actuators effort was put into the demanded control effort vector

$$\boldsymbol{\delta}_{\mathrm{cmd}} = \begin{bmatrix} T_{\mathrm{fl}} & T_{\mathrm{fr}} & T_{\mathrm{rl}} & T_{\mathrm{rr}} & \theta_{\mathrm{f}} & \theta_{\mathrm{r}} \end{bmatrix}^{\mathrm{T}} \tag{6.28}$$

In the effectiveness matrix $\boldsymbol{B}$ it was modelled how each actuator in the control effort vector affected the system. The torque contributions to the longitudinal reference force F_{x} could be modelled as

$$F_{\mathrm{x}} = \sum \frac{1}{r} T_{i,j} \quad i \in [f, r], \quad j \in [l, r] \tag{6.29}$$

where r is the wheels radius. Similarly, the wheel torque effects on the M_{z} can be formulated as follows

$$M_{\mathrm{z},1} = \sum \pm \frac{t}{2r} T_{i,j} \quad i \in [f, r], \quad j \in [l, r] \tag{6.30}$$

where t is the width of the vehicle and therefor $t/2$ is used as a lever, the minus sign was used for the wheels on the left side of the vehicle. The lateral force that comes from the front steering angle contributes to the M_{z} in the following way

$$M_{\mathrm{z},2} = 2C_{\alpha} a \theta_{\mathrm{f}} \tag{6.31}$$

where the number two is because of that there are two wheels on each axle. The rear steering angle effects on M_{z} can be formulated in a similar way

$$M_{\mathrm{z},3} = -2C_{\alpha} b \theta_{\mathrm{r}} \tag{6.32}$$

Hence, the total M_{z} can be calculated as

$$M_{\mathrm{z}} = M_{\mathrm{z},1} + M_{\mathrm{z},2} + M_{\mathrm{z},3} \tag{6.33}$$

This leads to an effectiveness matrix $\boldsymbol{B}$ that has the following form

$$\boldsymbol{B} = \begin{bmatrix} \frac{1}{r} & \frac{1}{r} & \frac{1}{r} & \frac{1}{r} & 0 & 0 \\ \frac{-t}{2r} & \frac{t}{2r} & \frac{-t}{2r} & \frac{t}{2r} & 2C_{\alpha} a & -2C_{\alpha} b \end{bmatrix} \tag{6.34}$$

where the matrix with numerical values can be seen in Appendix A. The effects from the steering angles will only be valid while the vehicle is moving because according to the $\boldsymbol{B}$ matrix the vehicle can have an rotational force in M_{z} if only the wheels have an steering angle. However, this is deemed to not be a problem because it is not a type of reference the vehicle would get during normal driving. The $\boldsymbol{\delta}_{\text{cmd}}$, $\boldsymbol{v}$ and $\boldsymbol{B}$ was used in the quadratic problem formulation, repeated below

$$\min_{\boldsymbol{\delta}_{\text{cmd}}} ||\boldsymbol{\delta}_{\text{cmd}} - \boldsymbol{\delta}_{\text{d}}||^2_{2,W_{\boldsymbol{\delta}}} + \gamma||\mathbf{B}\boldsymbol{\delta}_{\text{cmd}} - \boldsymbol{v}||^2_{2,W_{\text{v}}} \tag{6.35}$$

subjected to

$$\underline{\boldsymbol{\delta}} \leq \boldsymbol{\delta}_{\text{cmd}} \leq \overline{\boldsymbol{\delta}}. \tag{6.36}$$

The desired control effort $\boldsymbol{\delta}_{\text{d}}$ was set to zeroes, see equation 6.37. By setting it to zeroes, the QP distributes the forces more evenly and as low as possible.

$$\boldsymbol{\delta}_{\text{d}} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}^T \tag{6.37}$$

The limits $\underline{\boldsymbol{\delta}}$ and $\overline{\boldsymbol{\delta}}$ for the actuators was derived from what they could handle. The limiting factor for the motors was the current they could draw from the batteries. From the maximum current of approximately ±15 A for each wheel the corresponding maximum torque could be calculated to ±5 Nm by using the torque constant K_{T}, which is derived in section 8.2.2.4. The maximum steering angles for the Ackermann steering systems are ±45°, because this limit never should be reached a limit of ±35° was used, which corresponds to ±0.61 rad. The values for the constraints are shown in equation 6.38.

$$\underbrace{\begin{bmatrix} -5 \\ -5 \\ -5 \\ -5 \\ -0.61 \\ -0.61 \end{bmatrix}}_{\underline{\boldsymbol{\delta}}} \leq \underbrace{\begin{bmatrix} T_{\text{fl}} \\ T_{\text{fr}} \\ T_{\text{rl}} \\ T_{\text{rr}} \\ \theta_{\text{f}} \\ \theta_{\text{r}} \end{bmatrix}}_{\boldsymbol{\delta}_{\text{cmd}}} \leq \underbrace{\begin{bmatrix} 5 \\ 5 \\ 5 \\ 5 \\ 0.61 \\ 0.61 \end{bmatrix}}_{\overline{\boldsymbol{\delta}}} \tag{6.38}$$

6.6 Low level controller

Given the fast dynamics of the longitudinal motors the assumption that the demanded and actual control effort was deemed to hold. Hence, the reference torque outputted from the control allocation layer could directly be used by the motors. However, given that the control allocation layer also outputs the demanded steering angles a controller was needed in order to make the actual steering angle follow the demanded.

The low level controller used for the steering, see Figure 6.2, can in a similar fashion as for the high level controller be motivated to use a PID type controller. However, the force needed to change the steering angle on the wheel axle is highly dependent on the speed in which the vehicle is traveling at. A proportional controller was chosen because it simplicity and for the reason that the high level controller can still remove steady state offset, that could come from the offset from the steering system.

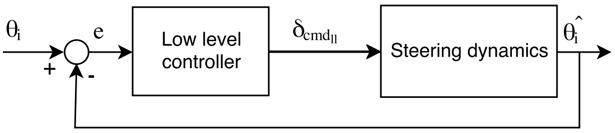


Figure 6.2: *The figure shows the structure for the P regulators in the steering systems.*

6.7 Summary of chapter

In this chapter the controllers used in the control structure was presented. First the use of PID controllers in the high level controllers was described. Several control allocation methods that could be used in the control allocation layer was presented and one used in the thesis was motivated and designed. Finally this chapter presented the use of P controllers for the steering systems.

7

Simulation

In the last chapter the complete driveline control system was designed to suit the vehicle. Together with the vehicle model, made in Chapter 5, the control system can be implemented and evaluated in simulations. This chapter will go through how a simulation environment was created in order to test the control system and especially the control allocation layer. It starts by going through how each layer in the control structure was implemented before it is describes how the vehicle model was implemented. Finally, the two test scenarios, described in Section 4.2.1 are analyzed and the control parameters are tuned.

7.1 Simulation environment

The simulation environment was created in MATLAB® Simulink® and it was used for testing and verifying the control system for the vehicle. It was structured to make it easy to test and verify the functionality of all parts that builds the control system, by using the same structure as the block diagram in Figure 4.1. The main layout of the simulation environment is shown in Figure 7.1, note that the reference velocity and heading was generated inside the High Level Controller block.

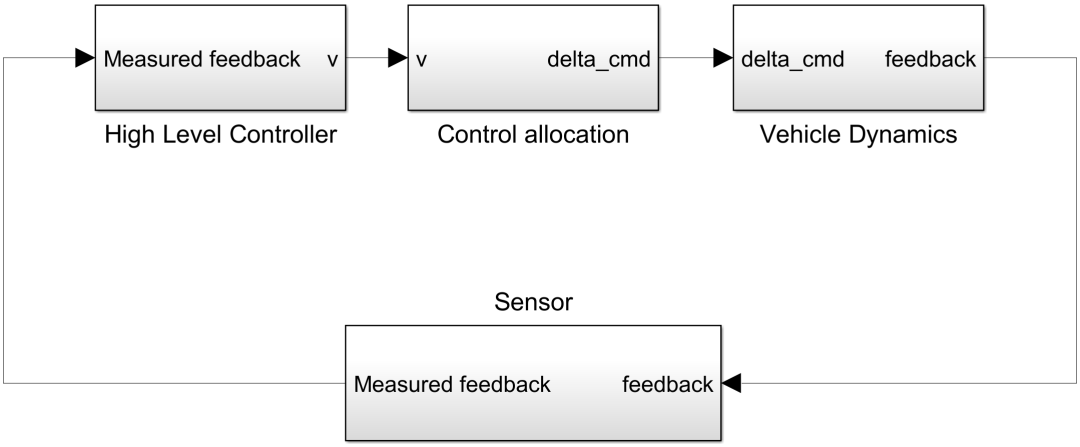


Figure 7.1: *The layout of the simulation environment created in Simulink®. As can be seen it has the same structure as the control system in Figure 4.1.*

7.2 Controller implementation

This section describes how the controller implementation was made in the simulation environment. It shortly describes what each control layer did and how the parameters in each layer were tuned. Naturally, the high level controller is first described and then the control allocation layer. For the controller implementation it was deemed that the assumption in equation 4.5, repeated below

$$\boldsymbol{\delta}_{\mathrm{cmd}} = \boldsymbol{\delta} \tag{7.1}$$

was true because the main focus was to test the two upper most layers. Hence, this resulted in that no low level controller was necessary and therefore not implemented in the simulations.

7.2.1 High level controller

The purpose of the high level controller was to generate the virtual command vector $\boldsymbol{v}$ to the control allocation layer in order to remove the difference between the actual and desired states. In the simulation environment the high level controller consisted of two PID controllers, one for each state that was controlled. The PID controllers tried to minimize the error in the longitudinal velocity and the vehicles heading angle. The controller parameters was tuned in order to achieve the necessary performance needed to meet the requirements stated in Chapter 3. First the proportional gain K_{p} was tuned in order to yield a fast response and then the integral gain K_{i} was added in order to remove the steady-state offset. No derivative gain K_{d} was necessary to meet the requirements and therefore it was not used. The final values can be found in Table 7.1 and the inside of the high level controller block can be seen in Figure B.1.

Table 7.1: *The PID controllers parameters used for the high level controllers in the simulations.*

(a) *F_{x} parameters*

Parameter	Value
K_{p}	100
K_{i}	20
K_{d}	0

(b) *M_z parameters*

Parameter	Value
K_{p}	600
K_{i}	70
K_{d}	0

7.2.2 Control allocation

The functionality of the control allocation, as described in Section 6.5, was implemented in the simulation environment. The code was implemented as MATLAB® functions in the Control allocation Algorithm block in Figure B.2.

The weighting matrices $\boldsymbol{W}_{\delta}$ were chosen to achieve the desired M_{z} by first changing the angles of the wheel axles and then if necessary applying different amounts of torques on the left and right side of the vehicle. Hence, the weights of the steering angles was set to be much lower than the weights of the effort from the longitudinal motors. As described in Section 3, the vehicle would only travel at low velocities therefore was the weighting factors of the longitudinal motor set to the same values because the vehicles weight transfer was small. If there was large weight transfers during normal operation it could for example during an acceleration phase have been better to have a lower cost for the longitudinal motors in the rear such that the wheels with the most grip outputted the most torque. The desired states of the vehicle i.e. the heading and the velocity was deemed to have the same priority, therefore all the weights in $\boldsymbol{W}_{\mathrm{v}}$ was set to 1. The weighting factor γ was set high in order to prioritize the desired virtual command $\boldsymbol{v}$ over the desired control effort $\boldsymbol{\delta}_{\mathrm{d}}$. All the weights can be seen in Table 7.2.

Table 7.2: *The weights used in the the simulation tests for the control allocation layer. The $\boldsymbol{W}_{\delta}$ told the CA solver how expensive it was to use an actuator relatively to the other ones. The $\boldsymbol{W}_{\mathrm{v}}$ specified the relative importance of the two forces in the virtual command vector $\boldsymbol{v}$. Finally, the γ was used to specify for the CA solver if the virtual command $\boldsymbol{v}$ was more important than the desired control effort $\boldsymbol{\delta}_{\mathrm{cmd}}$.*

(a) *W_{δ}*

Parameter	Value
$w_{\mathrm{wheel_{fl}}}$	1000
$w_{\mathrm{wheel_{fr}}}$	1000
$w_{\mathrm{wheel_{rl}}}$	1000
$w_{\mathrm{wheel_{rr}}}$	1000
$w_{\mathrm{steering_{f}}}$	1
$w_{\mathrm{steering_{r}}}$	1

(b) *W_{v}*

Parameter	Value
$w_{\mathrm{F_x}}$	1
$w_{\mathrm{M_z}}$	1

(c) *γ*

Parameter	Value
γ	1e6

7.3 Vehicle model

As described in Chapter 5 the chosen vehicle model was a 2D-planar motion model. This section goes through how this model was implemented in the simulation environment. The parameters used was either measured or estimated from the test vehicle, see Table 5.1.

7.3.1 Modelling of the actuators

In a perfect system, in terms of complexity, actuators would have an instant response time and the output would be exactly what was desired by a controller. This is never possible, therefore all the actuators was modelled as first order low pass filters, to have a more realistic behaviour. The longitudinal motors reacted faster to changes than the turning of the front and rear axle, therefor a low pass filter with a smaller time constant was used for the motors compared to those used for the steering, see equation 7.2 and 7.3.

$$G_{\text{Motors}}(s) = \frac{1}{1 + s/100} \tag{7.2}$$

$$G_{\text{Steering}}(s) = \frac{1}{1 + s/10} \tag{7.3}$$

7.4 Simulation results

To verify that the two step responses meets the requirements set in Chapter 3, the two test scenarios was carried out in Simulink®. The first step response test was in the vehicle´s longitudinally velocity and the second one was in the vehicles heading whiles traveling at a constant velocity.

7.4.1 Longitudinal step response

The reference velocity in this test was set to 1.5 m/s, according to the requirements set in Section 3.1, and vehicles response was recorded, see Figure 7.2. As can be seen the control system manages to reach the desired velocity with only a small overshoot. The computed reference force F_x from the high level controller, see Figure 7.3, behaved as expected for this test because it was large in the beginning of the acceleration face in order to reach the desired velocity. Then it became small before it grew slightly and settled to a constant small force to counteract the frictions while it kept the desired velocity.

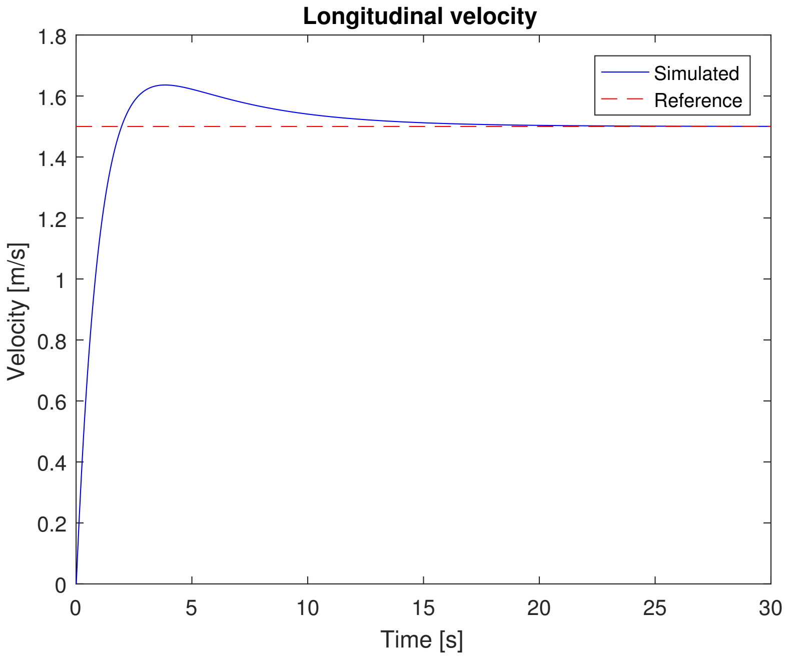


Figure 7.2: *The figure shows a step response on the longitudinal velocity of the vehicle in the simulation environment. As can be seen the driveline control system managed to reach the desired velocity with a little overshoot. The steady state error was removed by the integral part in the high level controller.*

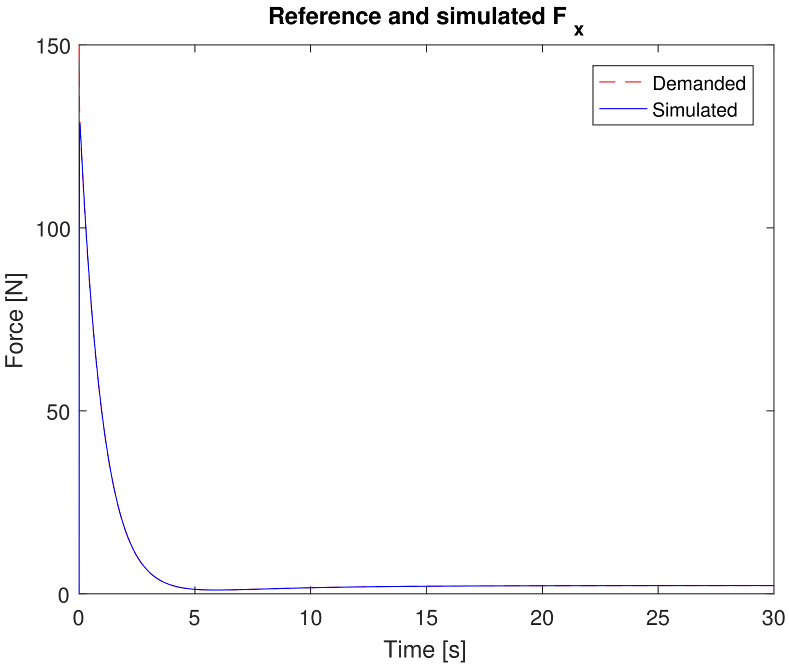


Figure 7.3: *The figure shows the demanded force generated by the high level controller and the simulated force generated by the vehicle in the 1.5 m/s longitudinal step response. Note that both the demanded and the simulated are on the same line.*

This calculated force was then fed into the control allocation algorithm which calculated the desired control efforts. Because the vehicles heading reference was kept at a constant level, the demanded steering angles was never changed from zero. Thus, the only actuators used was the longitudinal motors, see Figure 7.4. Because they were given the same weight they always contributed with the same effort and followed the reference force curve.
The results from the test scenario for the longitudinal velocity can be seen in Table 7.3. As can be seen, the simulated data shows that the system passes all the requirements for the test.

Table 7.3: *The table shows the calculated values and the requirements for the performance variables described in Section 3.1. As can be seen all requirements for the longitudinal behaviour passed.*

Variable	Value	Requirement	Pass/Fail
Rise time	1.35 [s]	3 [s]	Pass
Max overshoot	9.07 %	20 %	Pass
Steady-state offset	0 %	± 10 %	Pass
Settling time	1.45 [s]	5 [s]	Pass

7.4.2 Lateral step response

During this test the velocity of the vehicle was kept at a constant level of 1.5 m/s. After 5 seconds a step of 30° in the reference heading was given, according to the

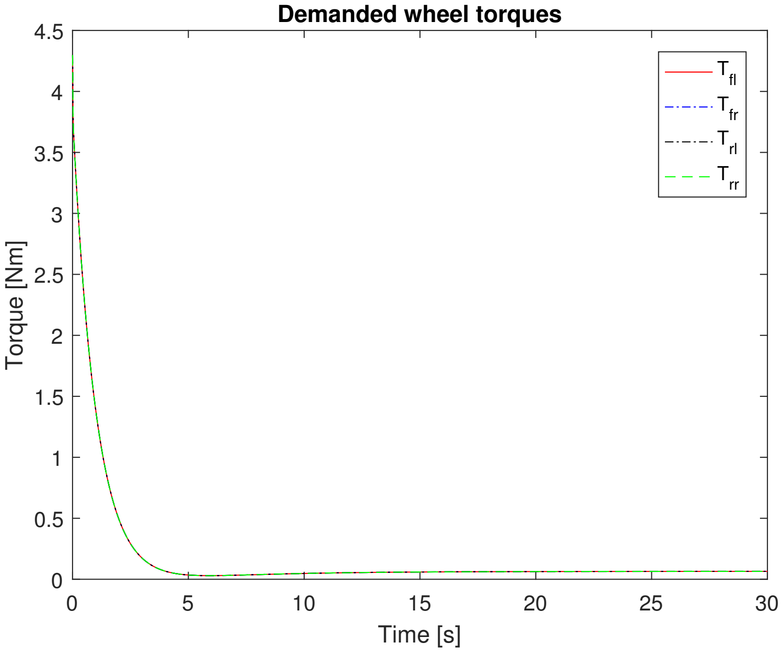


Figure 7.4: *The demanded wheel torque for each wheel calculated by the control allocation layer given the demanded force from the 1.5 m/s longitudinal step response. Because the vehicle was only going straight all the wheel torques was always calculated to equally large.*

requirements set in Section 3.2, the vehicles heading can be seen in Figure 7.5. As can be seen the driveline control system successfully manages to track the given reference heading and all the requirements on the lateral behaviour was fulfilled, which can be seen in Table 7.4. The reference torque M_z, that was calculated by the high level controller, can be seen in Figure 7.6. As explained in Section 7.2.2 the weight of the steering effort was much lower than the weight for the longitudinal motors. Therefore the control allocation prioritized the steering axles to reach the desired M_z and the rotational force that could be generated by distributing torque differently on the sides of the vehicle was small enough to be neglected. Because the cost of using the front and rear steering motor was the same, the computed steering angles always became equally large but with opposite signs, as seen in Figure 7.7.

Table 7.4: *The table shows the calculated values and the requirements for the performance variables described in Section 3.2. As can be seen all requirements for the lateral behaviour passed.*

Variable	Value	Requirement	Pass/Fail
Rise time	0.59 [s]	3 [s]	Pass
Max overshoot	3.83°	15°	Pass
Steady-state offset	0°	± 10°	Pass
Settling time	1.75 [s]	5 [s]	Pass

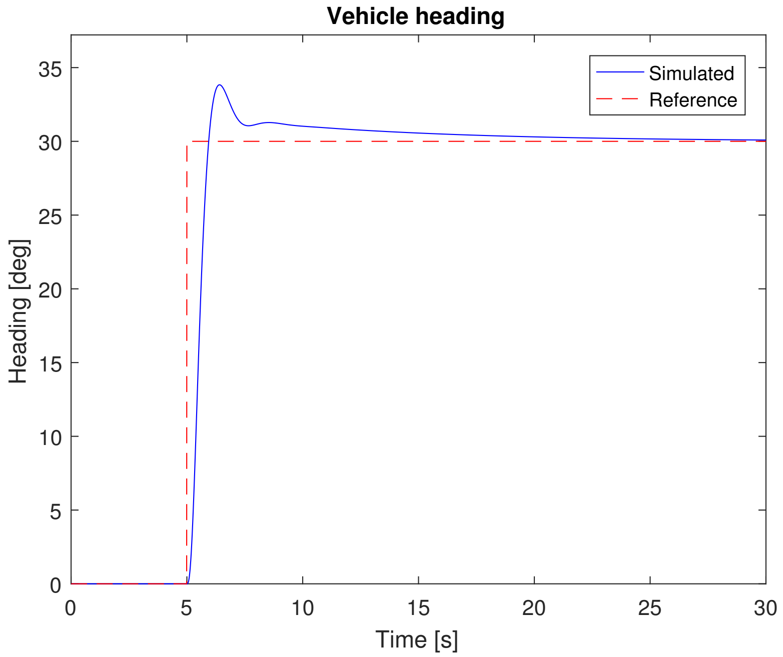


Figure 7.5: *The figure shows the heading step response of the vehicle in the simulation environment. The vehicle was traveling at a constant velocity of 1.5 m/s when a step in the reference heading of* 30° *was given. As can be seen the control system managed to reach the desired heading and the steady state offset was removed.*

7.5 Summary of chapter

This chapter described how the control structure was designed in the simulation environment for testing and verifying it according to the tests presented in Section 4.2.1. It presented how the vehicle and actuators was modelled and implemented in the environment. Finally, the simulations showed promising results for the longitudinal and the lateral behaviour.

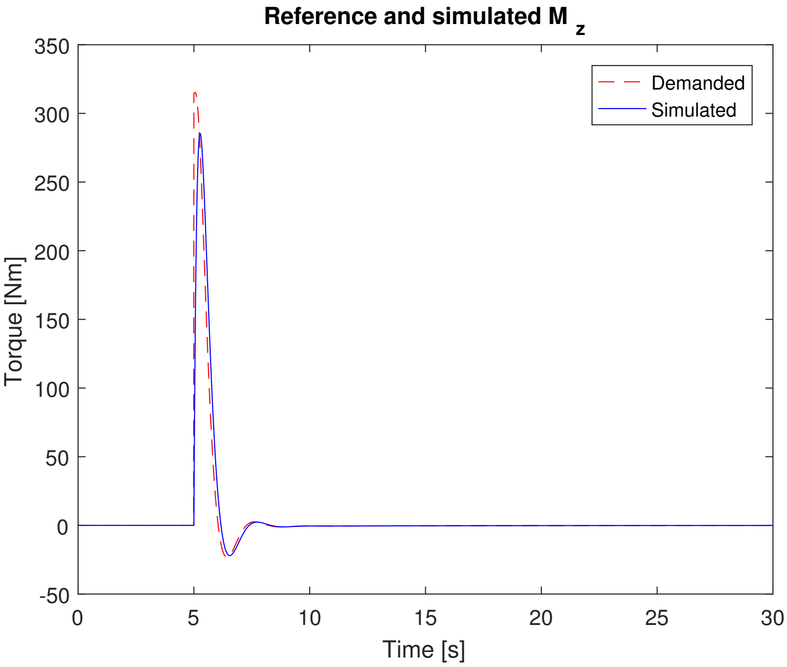


Figure 7.6: *The figure shows the demanded rotational force generated by the high level controller and the simulated rotational force generated by the vehicle, in the* 30° *heading step response.*

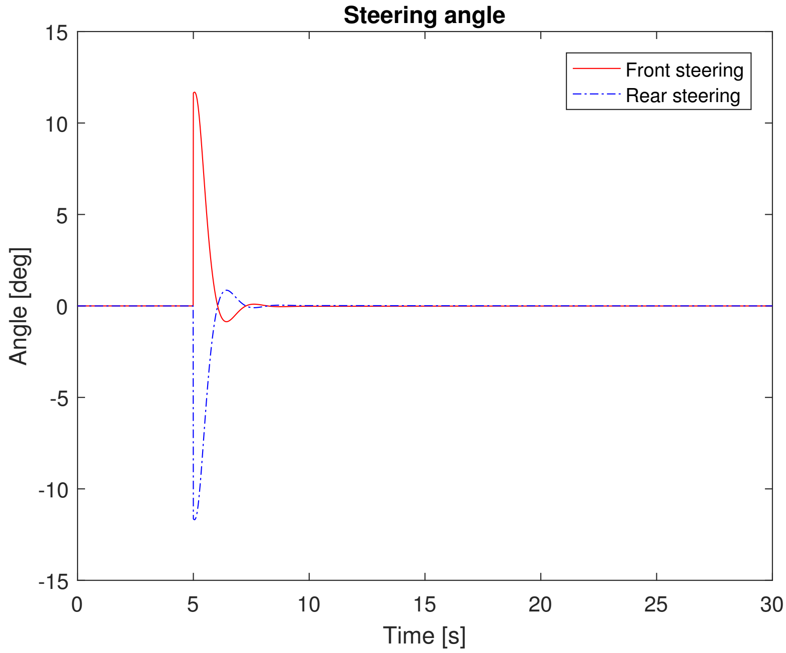


Figure 7.7: *The demanded steering angles for each wheel axle calculated by the control allocation layer, given the demanded rotational force, from the* 30° *lateral step response.*

8
Implementation

In the last chapter the designed control system were tested and tuned in simulations. It was also important to verify that the driveline control system could work outside simulations, hence it should be tested on the vehicle described in Chapter 2. This chapter describes how it was implemented in the vehicle on a Beaglebone black. First it is described how the code was structured in ROS, then how the necessary state was estimated to give the necessary feedback for the controller. Finally, the tests are described that were carried out to verify the requirements stated in Chapter 3.

8.1 Implementation in ROS

The designed control structure, see Chapter 6, was a mix of the block diagrams shown in Figure 4.1 and 4.2 depending on which actuators that needed an additional low level controller, were implemented in simulations. The code used in these simulations was rewritten into C++ and python depending on what existing code libraries that were available. Hence exactly the same structure could not be used, in addition the structure in ROS included the functionality needed to read sensor values and to send commands to all the actuators. The structure of which nodes that communicated to each other can be seen in Figure 8.1 and below is a list of all the nodes with a description of their functionality.

1. **Motion planning.** The motion planning node generated the references for the test scenarios as explained in Section 4.2.2. This node only had the basic functionality of generating steps in the reference vector r. It can be replaced with more advanced motion planning algorithms as long as they sends the references on the same topic.
2. **Longitudinal controller.** In the ROS structure the high level controller was divided into two nodes. The longitudinal controller was the first one, it received the desired velocity $v_{\mathrm{x,r}}$ from the motion planning node and also the current velocity v_{x} from the velocity estimation node. It was a PID controller used to calculate the output F_{x} that is the first element in the virtual command vector $\mathbf{v}$.
3. **Heading controller.** The second node of the high level controller was the heading controller. It received the desired heading θ_{r} from the motion planning node and the current heading angle estimate $\hat{\theta}$ of the vehicle from the compass node. Similar to the longitudinal controller the heading controller used a PI controller to calculate the output M_{z}. Together with the F_{x} from the

longitudinal controller it built the virtual command vector $\boldsymbol{v}$ to be used in the control allocation.

4. **Control allocation.** The control allocation node only consisted of the solver for the QP problem. It received the virtual command $\boldsymbol{v}$ and delivered the demanded control effort δ_{cmd} for each actuator. The QP solver was generated to C++ code from the MATLAB® code used in the simulations by the tool MATLAB® Coder™[23].
5. **Motor controller.** The motor control node mapped the demanded torques to demanded currents that was used as the control signals to the motors. The node used the demanded current calculations which is described in Section 8.2.2.4. The control signals was sent to the motors by serial communication to one of the four longitudinal motor controllers. This motor controller then delivered the control signals for the other three motor controllers by CAN communication.
6. **Steering controller.** The steering control node was a low level controller for controlling the steering angles of the front and rear axle. It used the information from the steering potentiometers to estimate the steering angle at both the front and the rear axle. From the differences between the reference and the estimated steering angles it calculated two control signals which was converted into a PWM signals and sends them out to the steering motor controller.
7. **Velocity estimation.** The velocity estimation node read a motor speed and maps it to the velocity of the vehicle. This information was used as feedback in the longitudinal controller.
8. **Compass.** The compass node read the vehicles heading and sent the information to the heading controller. The code was and based on an Adafruit library but it was converted into a ROS node.

8.2 Implementation in the test vehicle

The ROS structure was implemented on the BBB in the test vehicle. The sections below consists both of how the nodes received data and how it sent commands to the motor controllers as well as how the nodes used the measured data to estimate the vehicles states.

8.2.1 Selection of sampling time

The selection of sampling time is an important design choice for a control systems, this section describes how the sampling time was selected. According to [24] a rule of thumb can be used to determine the required sampling time. It states that the sampling time should at least be a tenth of a systems process time. The process time is defined as the time it takes from the system to go from 0 % to 63 % of the maximum desired amplitude of a step response. The process time was identified for the vehicle by doing a step response on both the velocity and heading of the vehicle. The step response in the vehicles velocity is shown in Figure 8.2 together with the

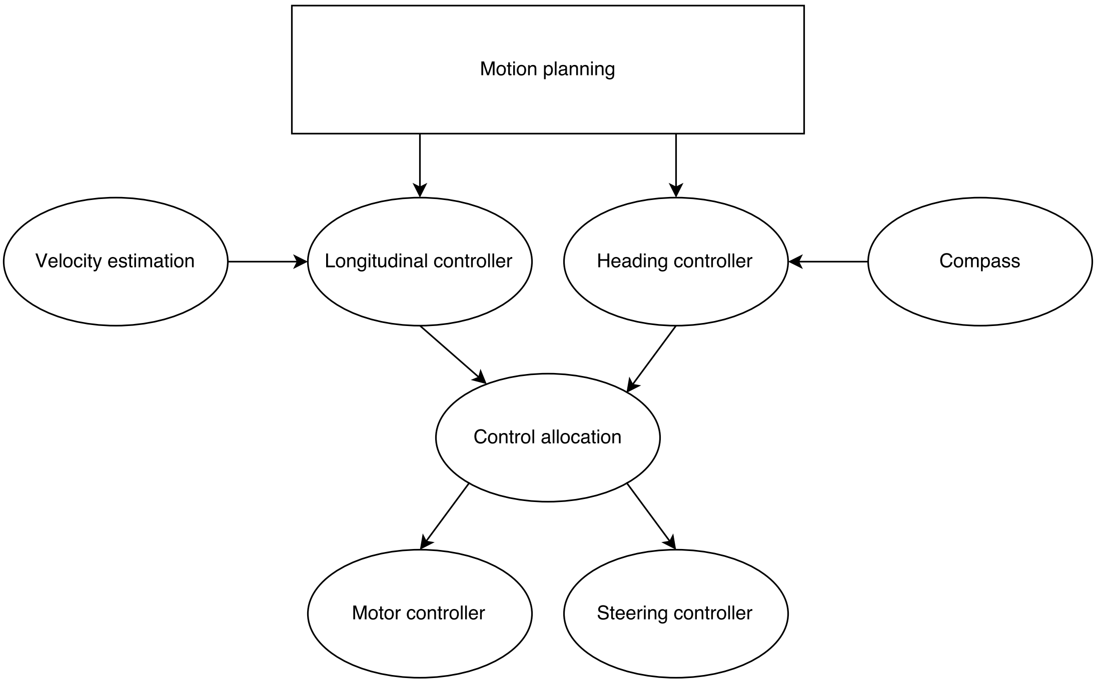


Figure 8.1: *Structure of the ROS nodes used in the implementation of the driveline control system for the vehicle. It shows how they communicate with each other. The motion planning node sent the references to the high level controllers. The high level controllers calculated the error in velocity and heading with the information from the Compass and the Velocity estimation nodes. The reference forces are then fed into the control allocation node which calculated the individual wheel torques and steering angles. The steering control node controlled the steering angles and the Motor control node mapped the torques requested by the control allocation node to the currents which was sent to the motors. This is a similar structure as in Figure 4.1 and 4.2*

process time. The step response for the heading of the vehicle can be seen in Figure 8.3.

The identified process times are shown in Table 8.1 together with the calculated sampling times and the sampling frequencies that was deemed to be necessary according to the previously stated rule of thumb. Naturally the highest sampling time was selected and therefore should the control system operate at 13 Hz, this could be specified for each node in ROS.

Table 8.1: *The table shows the process times for both the lateral and longitudinal behaviour, from this was the recommended sampling frequencies calculated using the thumb rule. According to this data the control system should operate at approximately 13 Hz.*

Behaviour	Process time [s]	Sampling time [s]	Sampling frequency [Hz]
Lateral	0.93	0.093	10.75
Longitudinal	0.75	0.075	13.32

However, ROS is not a real time operating system and the nodes does not execute in a predetermined order this could result in a much lower sampling time for the

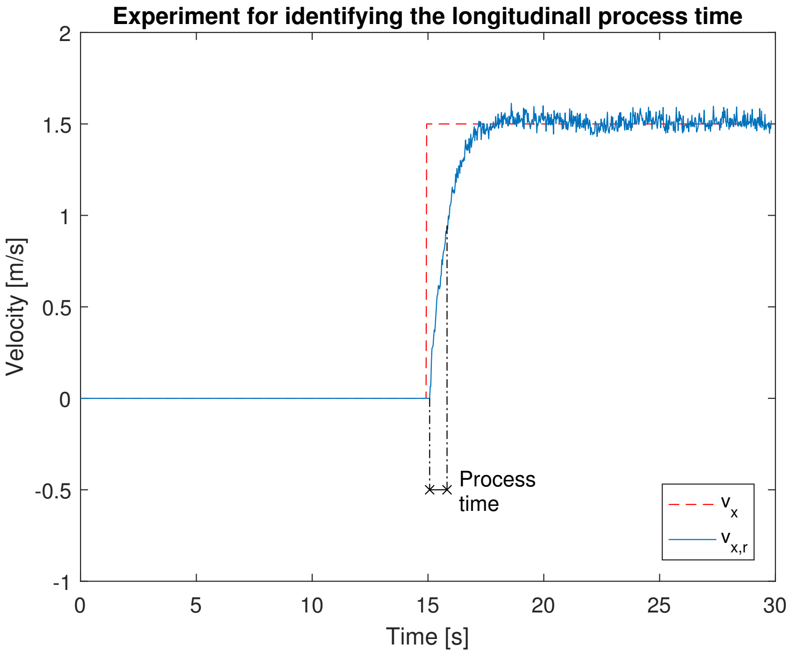


Figure 8.2: *The figure shows the recorded data from an experiment where the longitudinal process time was identified. The process time is the time it takes for the system to go from 0 % to 63 % of the maximum amplitude of a given step reference. In this experiment the desired velocity was set to be 1.5 m/s and the identified process time was 0.75 s.*

whole control system than what was specified for each node. This was confirmed by calculating how much time it takes for a given signal to travel through all necessary nodes. Two experiments was made where all nodes operated at a specified frequency and the time it took from that the high level controller received a step until the low level controllers outputted a current command to the longitudinal motor was calculated, see Table 8.2.

Table 8.2: *The tables shows data from an experiment where the nodes running in ROS was set to execute at two different sampling frequencies. The second column was the time it took from that a reference was received in the high level controller to the time the low level controller outputted a demanded torque for the longitudinal motors. Hence, this shows what the actual sampling frequency of the total driveline control system was.*

Sampling frequency [Hz]	Time delay [s]	Actual sampling frequency [Hz]
10	0.24	4.12
40	0.08	12.50

From this result it was clear that a specified sampling frequency of at least 40 Hz was required in order to make the total control system operate close to the required sampling frequencies in Table 8.1. A sampling frequency of more than 40 Hz was not possible because the state estimation node needed to wait approximately 20 ms after a command was sent to the motor controllers to receive the velocity before it arrived. This resulted in an upper limit of 50 Hz and given that the CA needs time

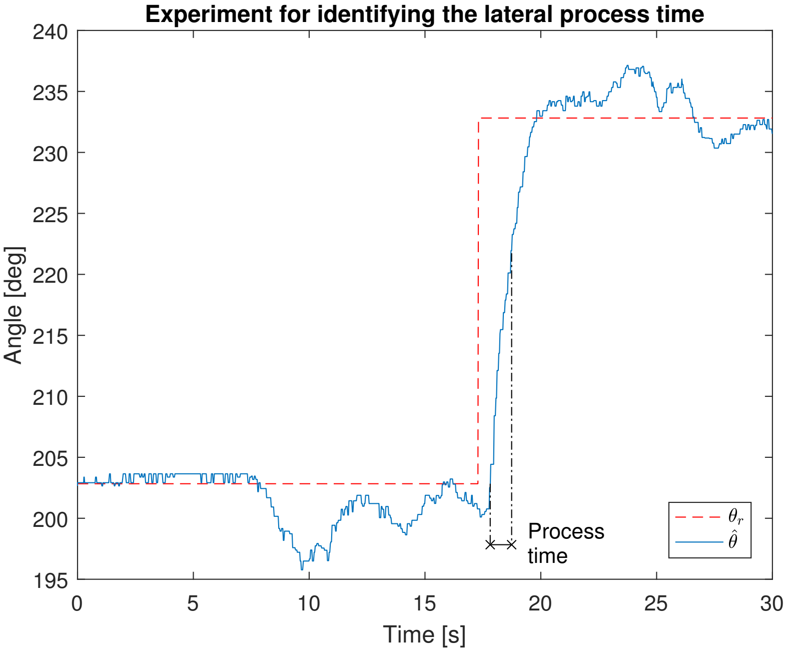


Figure 8.3: *The figure shows data from an experiment where the vehicle´s lateral process time were identified. The process time is the time it takes for the system to go from* 0 % *to* 63 % *of the maximum amplitude for a given step reference. The vehicle was traveling at a velocity of 1.5 m/s when a step in the reference heading of* 30° *was inputted. The process time was identified conservatively which means that the delay of the system was ignored, this resulted in a faster process time. The process time was identified to be 0.93 s.*

to find a solution 40 Hz was selected as the sampling frequency for all the nodes.

8.2.2 State estimation

The states of the vehicle was estimated and measured in various ways to give feedback to the control system. All state estimations started by measuring the data to be used with the sensors built in the vehicle. The data in its raw format is not always useful as it is and therefore additional mapping, sanity checks and sometimes filtration is key for a proper state estimation. This section describes how the necessary states was estimated.

8.2.2.1 Velocity estimation

The velocity of the vehicle needed to be estimated in order to provide feedback for the high level controller, see Section 4.1. The way the velocity was estimated was by using the motors hall sensors, these was used by the VESC motor controllers which measured the electrical revolutions per minute (ERPM) for each wheel. The ERPM is the motors mechanical revolutions per minute (RPM) multiplied by the number of pole pairs. ERPM could therefor be converted to RPM, which could give an estimate of the vehicles velocity.
A hall sensor measures the voltage that arises when a magnetic field passes nearby and a BLDC motor have magnets inside them. Hence, the hall sensor will give

out an periodic signal if the wheel rotates at a constant velocity. The number of pole pairs in the motor was identified by measuring the period and the number of rotations. The result from this was

$$N_{\text{Poles}} = 5 \tag{8.1}$$

which means that the signal would go high 5 times for each revolution. This was the ERPM that the motor controller could measure. The motors are also geared and the gear ratio was

$$R_{\text{Gear}} = \frac{Teeth_{\text{Wheel}}}{Teeth_{\text{Motor}}} = \frac{62}{10} = 6.2 \tag{8.2}$$

The ERPM together with the two constants could be used to estimate the velocity. First the ERPM was converted to the RPM of the wheel

$$RPM_{\text{Wheel}} = \frac{1}{R_{\text{Gear}} \cdot N_{\text{Poles}}} \cdot ERPM \tag{8.3}$$

With this the velocity in the longitudinal direction could be calculated by the following relationship

$$v_{\text{x}} = \frac{1}{R_{\text{Gear}} \cdot N_{\text{Poles}}} \cdot \frac{2 \cdot r \cdot \pi}{60} \cdot ERPM = C_{\text{erpm}} \cdot ERPM \tag{8.4}$$

where r is the wheel radius, hence

$$C_{\text{erpm}} \approx 0.0003885 \tag{8.5}$$

8.2.2.2 Heading estimation

The available method of estimating the heading of the vehicle was to use a compass in a form of a magnetometer. The compass had already made code which measured the amplitude of the magnetic field in x and y direction, from these measurements the heading was estimated and also filtered. The code was put into a ROS node in the ROS structure. A magnetometer can be a good way to estimate a heading or direction if the environment is noise free. But when the environment is not ideal the signal can be disturbed as seen in Figure 8.4. This magnetometer test was logged while pushing the vehicle on a straight line and as can be seen in the figure the signal was catching disturbances.

8.2.2.3 Steering angle estimation

As described in the Section 2.3, the steering angle was estimated by using one potentiometer on each the steering axle. The potentiometers received a constant supply voltage from the BBB and outputted a voltage between 0 V and the reference voltage, 1.8 V, depending on how much the wheels was turned, this voltage was fed back to the BBB. The value read by the analogue pins for the front and rear steering was mapped to the corresponding steering angle by the linear formula

$$y = k \cdot x + m \tag{8.6}$$

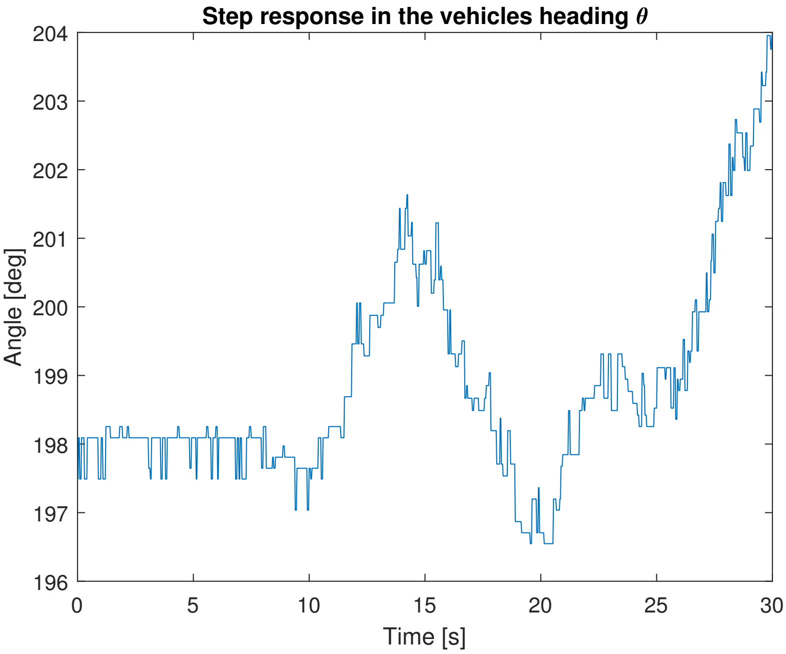


Figure 8.4: *This figure shows the heading estimate while pushing the vehicle on a straight parking line. As can be seen the measured heading was sensitive to disturbances.*

where the output angle y was either θ_{f} or θ_{r} and x was the input value. The constants k and x was estimated by performing a test where the steering angle was set to several positions and the corresponding feedback voltages was recorded. The root mean square was used in MATLAB®. The original data can be seen in Figure 8.5 and the identified parameters are shown in Table 8.3.

Table 8.3: *The table shows the constants used in equation 8.6 for estimating the steering angles. The constants was estimated by recording both the angle and the voltage for one axle and then fitting a linear curve to this data.*

Parameter	Value
k	4.189
m	-2.067

To decrease the noise level in the measured signals for the front and rear steering angles they where filtered through a infinite impulse response (IIR) filter of the first order and it was done as following

$$\hat{\theta}_{\mathrm{f}}[k] = \alpha_{\mathrm{f}} \cdot \theta_{\mathrm{f}}[k] + (1 - \alpha_{\mathrm{f}}) \cdot \hat{\theta}_{\mathrm{f}}[k-1] \tag{8.7}$$

$$\hat{\theta}_{\mathrm{r}}[k] = \alpha_{\mathrm{r}} \cdot \theta_{\mathrm{r}}[k] + (1 - \alpha_{\mathrm{r}}) \cdot \hat{\theta}_{\mathrm{r}}[k-1] \tag{8.8}$$

where the value for α was selected during the testing of the control system in the vehicle, the value can be seen below

$$\alpha_{\mathrm{f}} = \alpha_{\mathrm{r}} = 0.3 \tag{8.9}$$

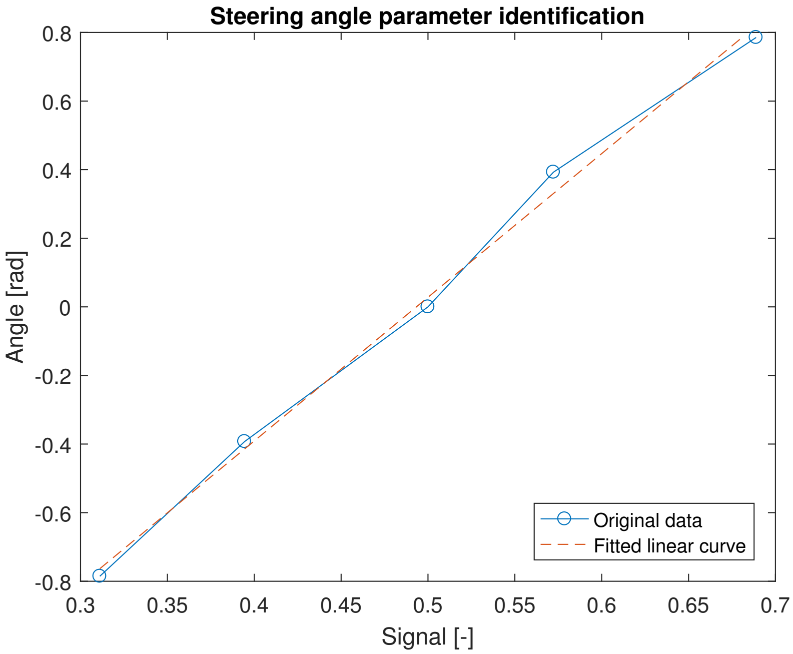


Figure 8.5: *The figure shows the linear relationship between the voltage from the potentiometer and the steering angle. The voltage was read by the BBB and converted into values between 0 and 1 while the steering angle was estimated by visual inspection. As can be seen, the fitted linear curve approximates the function well.*

A bode plot of the filter can be seen in Figure 8.6, as it shows all frequencies above the zero frequency gets a lower magnitude.

8.2.2.4 Calculation of demanded currents

The control signals for the longitudinal BLDC motors was demanded current while the output from the control allocation was demanded torque. Hence, the demanded torques needed to be converted into demanded currents. For BLDC motors there exists a linear relationship between the motor current i and the motor torque τ i.e.

$$\tau = K_{\mathrm{T}} \cdot i \quad \Rightarrow \quad i = \frac{\tau}{K_{\mathrm{T}}} \tag{8.10}$$

where K_{T} is the motor constant. This could be estimated by rewriting equation 8.10 to

$$K_{\mathrm{T}} = \frac{\tau_{\mathrm{i,j}}}{i_{\mathrm{i,j}}} \tag{8.11}$$

and measuring the motor current $i_{\mathrm{i,j}}$ and the output torque $\tau_{\mathrm{i,j}}$. However, the motor output torque $\tau_{\mathrm{i,j}}$ could not be measured directly but the wheel torque could be estimated by using the following relationship

$$\tau_{\mathrm{i,j}} = f_{\mathrm{i,j}} \cdot r = \frac{F_{\mathrm{x}}}{4} \cdot r = \frac{a_{\mathrm{x}} m}{4} \cdot r = \frac{m \cdot r}{4} \cdot \frac{\mathrm{d}}{\mathrm{dt}} v_{\mathrm{x}} \tag{8.12}$$

where v_{x} can be measured directly. Using this for calculating the torque constant directly means that it also includes the gear ratio. However this makes it easier to

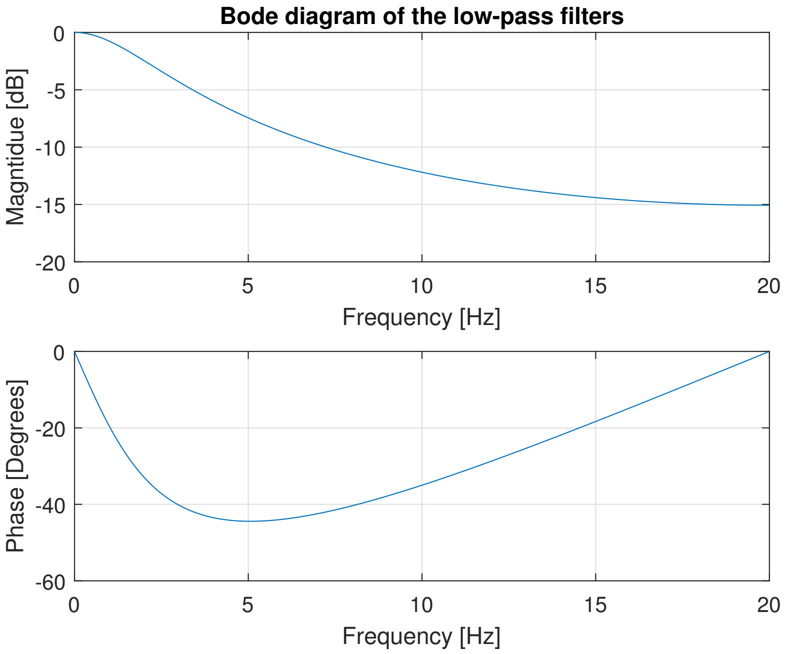


Figure 8.6: *The figure shows the bode diagram of the low-pass filters used to filter both the front and rear steering angles. The filter was a first order IIR low-pass filter.*

calculate the current needed to yield the demanded wheel torque, which is calculated by the control allocation layer, see Section 4.1.2. Hence, equation 8.11 becomes

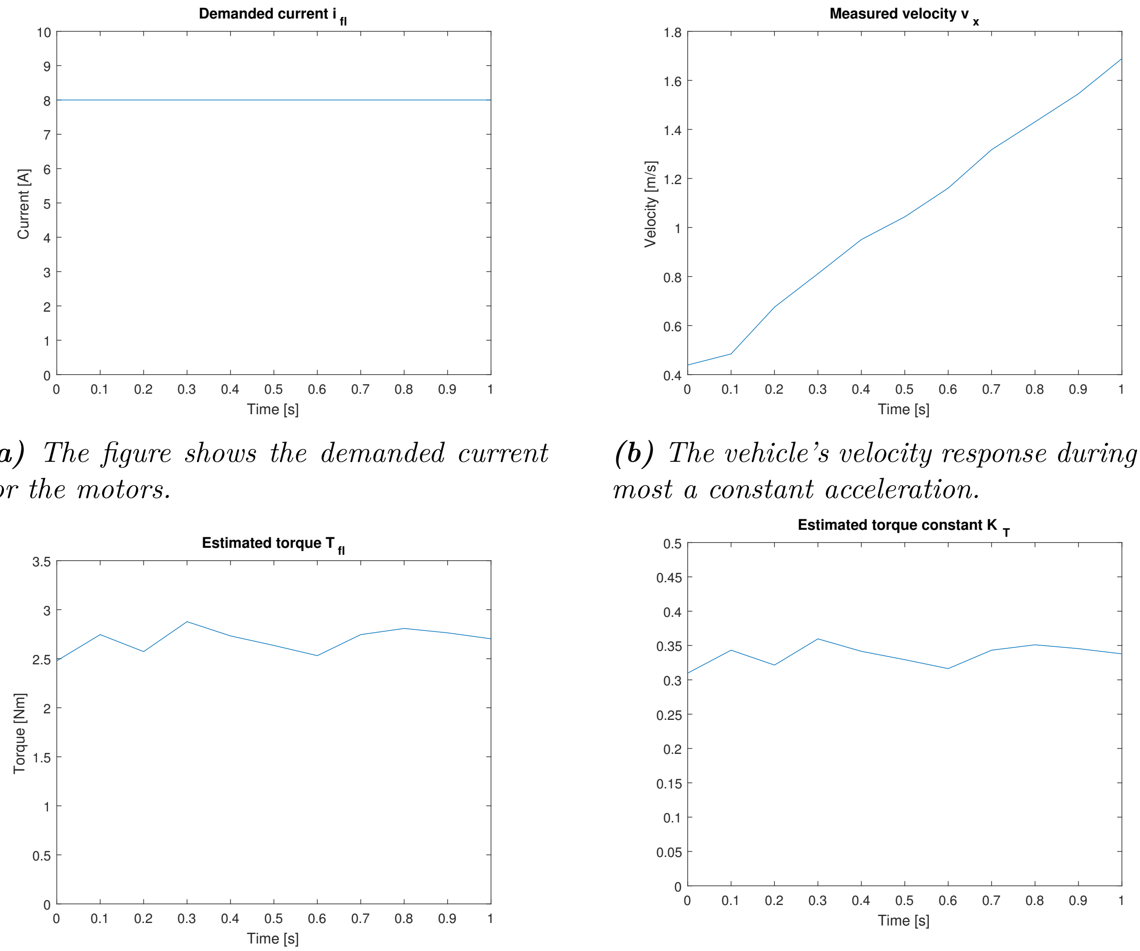
$$K_{\mathrm{T}} = \frac{m \cdot r}{4i} \cdot \frac{\mathrm{d}}{\mathrm{dt}} v_{\mathrm{x}} \tag{8.13}$$

This was done in an experiment were the demanded motor current i where held at a constant level for 1 second, which resulted in an almost constant acceleration, see Figure 8.7. From the estimated values of the torque constant in Figure 8.7d the mean value was calculated to be

$$K_{\mathrm{T}} = 0.34 \tag{8.14}$$

8.3 Test results from the vehicle

This section goes over the main results from the test carried out to verify the implementation and control design in the vehicle. It is discussed how the control parameters were tuned to yield the desired performance necessary for meeting the requirements in Chapter 3. First the longitudinal behaviour is tuned and investigated by analyzing a step response test in the vehicles velocity. After this is the same procedure made for a step response test in the vehicles heading. Finally, a last test is analyzed where the vehicle was turned in an more advanced way.



(a) *The figure shows the demanded current for the motors.*

(b) *The vehicle's velocity response during almost a constant acceleration.*

(c) *The torque response for one wheel derived from the velocity, mass and wheel radius.*

(d) *The estimated torque constant K_T. This was calculated by dividing the torque in 8.7c by the current in 8.7a.*

Figure 8.7: *The figure shows data from an experiment where the motor torque constant were identified by demanding a constant motor current for all four motors.*

8.3.1 Longitudinal step response

The longitudinal step response was the first test scenario, described in 4.2.2, that were evaluated in order to verify that the requirements for the closed loop system was reached. The motion planning node was programmed to generate a step of 1.5 m/s after a certain time while the program logged all necessary data. First the K_p gain was tuned to yield a fast rise time without oscillations, after that a K_i and K_d gain was added to remove the steady-state offset and to improve the performance. The step response for the final tuning can be seen in Figure 8.8 and the parameters can be seen in Table 8.4.

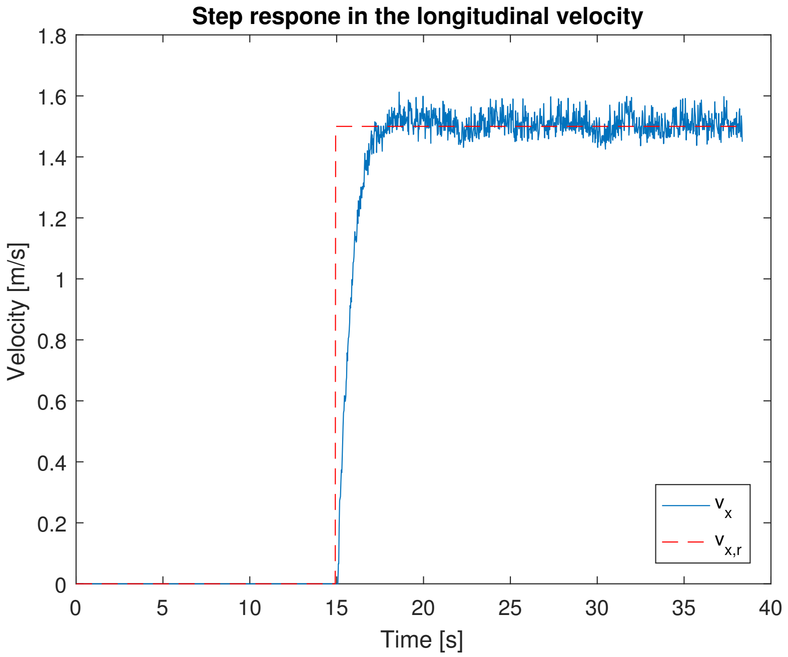


Figure 8.8: *Plot of the vehicles velocity response when a step of 1.5 m/s was applied. As can be seen the desired velocity was reached and zero offset was achieved.*

Table 8.4: *The parameters from the final tuning of the high level longitudinal controller. First the K_p was selected to yield a fast rise time, then the K_i and K_d was added in order to remove the steady-state offset and improve the performance.*

Parameter	Value
K_p	70
K_i	20
K_d	1

From the logged data of the velocity response the performance variables could be calculated and compared to the requirements. Figure 8.9 shows how the response compared to the requirements and the specific values of the performance variables are listed in Table 8.5.

Table 8.5: *The table shows the calculated values and the requirements for the performance variables described in Section 3.1. The reason that the max overshoot and steady-state offset is not applicable is that neither overshoot or steady-state offset occurred. Hence, all requirements for the longitudinal behaviour have passed.*

Variable	Value	Requirement	Pass/Fail
Rise time	1.58 [s]	3 [s]	Pass
Max overshoot	Not applicable	20 %	Pass
Steady-state offset	Not applicable	± 10 %	Pass
Settling time	1.85 [s]	5 [s]	Pass

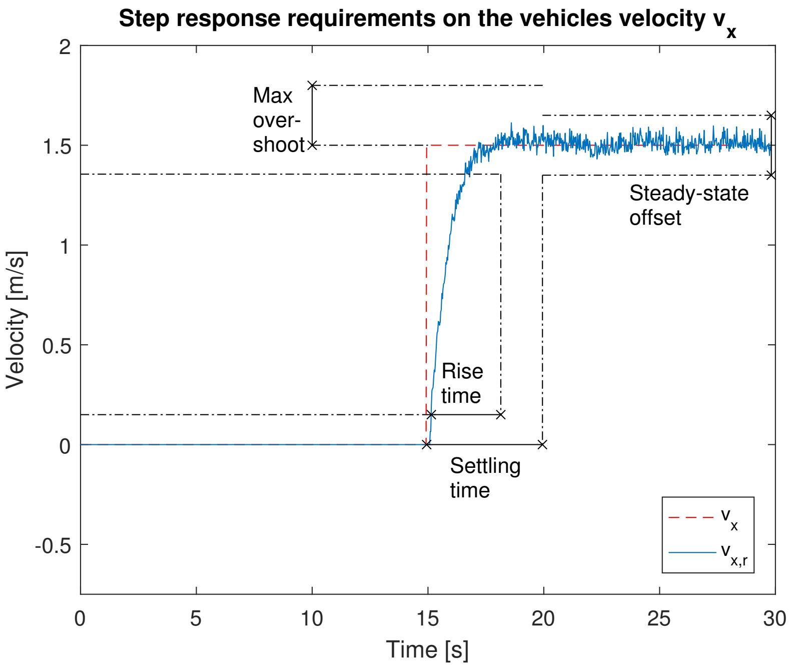


Figure 8.9: *The figure shows the system response compared to the requirements when a step in the reference velocity $v_{x,r}$ of 1.5 m/s is used as input. As can be seen the control system managed to control the vehicle such that the requirements described in Section 3.1 are reached.*

From the step response the force needed to accelerate and to keep the vehicle in the desired velocity can be evaluated, this is shown in Figure 8.10. The contributions from the gains in the PID controller are shown in 8.11.
The total demanded force F_x was fed into the control allocation node that calculates the torques that needs to be applied to the wheels in order to accelerate the vehicle with the given force, the torques can be seen in Figure 8.12. The CA distributes the torques equally over the four wheels because the costs for using them is set to the same value. The torques is then sent to the motor control node that calculates the current references that needs to be sent to the motors and it is done according to the motor torque constant calculated in Section 8.2.2.4. The currents calculated can be seen in Figure 8.13

8.3.2 Lateral step response

As described in Section 4.2.2 a test scenario were made to verify that the driveline control system fulfilled the requirements on the lateral behaviour. Here, the motion planing node generated a reference velocity of 1.5 m/s and when the velocity was reached it outputted a step of 30° on the reference heading θ_r. The driveline control system should thus try to steer the vehicle such that the new reference heading is reached. The tuning of the lateral controller was carried out in a similar way as the longitudinal counterpart, first a K_p was chosen such that a fast rise time was achieved without creating any oscillations. Then just a K_i was added to remove steady-state offset, however no K_d was added because the performance was already

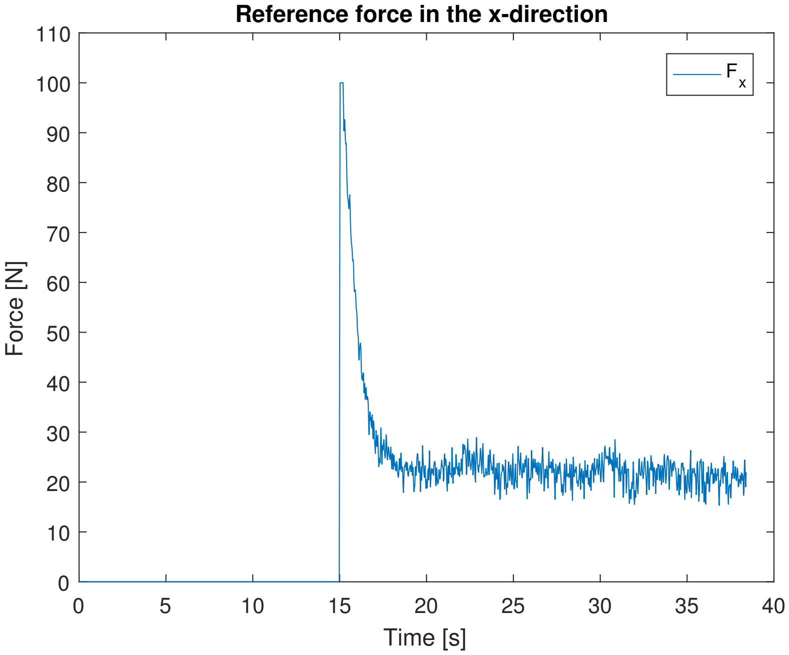


Figure 8.10: *The reference force generated by the longitudinal PID controller. The result was reasonable because first a large force was needed to accelerate the vehicle to the desired velocity and then a small force was needed to keep the desired velocity mainly because of friction in the system and between the tires and the ground.*

good enough. As can be seen in Figure 8.14 it successfully managed to follow the reference signal and it also successfully passed all the requirements which can be seen in Figure 8.15 and Table 8.6.
The small ripple that occurred was deemed to be caused mainly because of two reasons. First, as described in Section 8.2.2.2, the heading was only estimated by a magnetic compass which was sensitive to disturbances in the magnetic field, see Figure 8.4. Secondly, the measurements of the front steering angle was very sensitive to noise as can be seen in Figure 8.18 even though it is filtered by a low pass filter, as described in Section 8.2.2.3. This decreased the performance of the low level controller for the front steering angle.

Table 8.6: *The table shows the calculated values and the requirements for the performance variables described in Section 3.2 for the lateral behaviour. The reason that the max overshoot and steady-state offset is not applicable is that neither overshoot or steady-state offset occurred. Hence, all requirements for the lateral behaviour have passed.*

Variable	Value	Requirement	Pass/Fail
Rise time	1.47 [s]	3 [s]	Pass
Max overshoot	Not applicable	15°	Pass
Steady-state offset	Not applicable	±10°	Pass
Settling time	1.28 [s]	5 [s]	Pass

In Figure 8.16 the rotational force that the high level lateral controller generated is

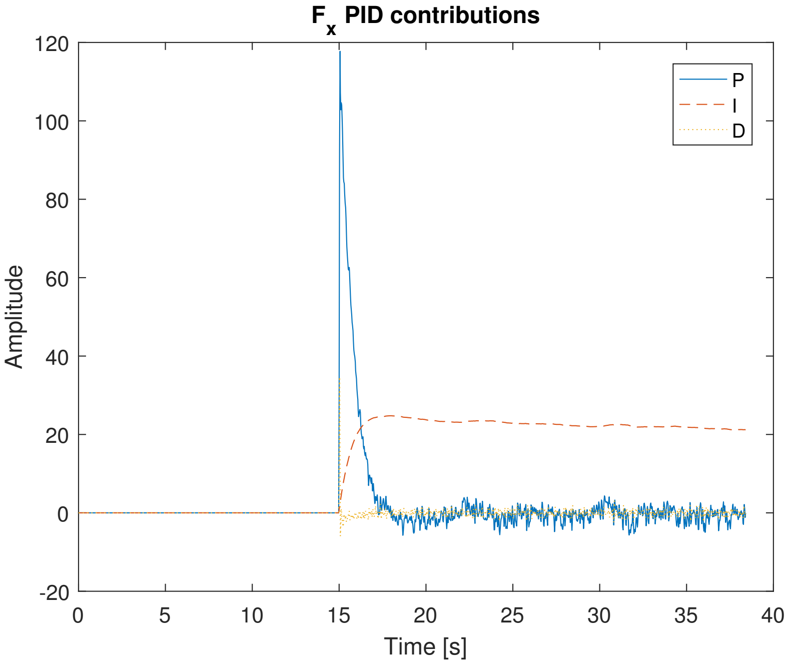


Figure 8.11: *The contributions from the proportional, integral and derivative parts from the longitudinal PID controller can be seen in this figure. The main contribution at the beginning of the step was from the proportional part while the integral part removed the steady-state offset. The derivative part had the lowest contribution, however this was not tuned further because to achieve a perfect tuning was out of this thesis scope.*

shown. How much the proportional and integrative gains contributed to the total M_z can be seen in Figure 8.17 and the final gains for them can be seen in Table 8.7.

Table 8.7: *The value of the parameters used in the final tuning of the lateral high level controller.*

Parameter	Value
K_p	-800
K_i	-50

The rotational force M_z was fed into the control allocation layer which calculated the desired control effort for all the actuators. As described in the 6.5 the steering angle for the front and rear axle were given smaller weights compared to the weights of the torques for the longitudinal motors. This makes the CA to first steer the vehicle by turning the front and rear axle and if needed by applying different amount of torques on the left and right side of the vehicle. Because the steering angles did not become saturated in carried out tests i.e. reaching the limits of ± 35°, before reaching the demanded M_z the torque on the left and right side was always the same, see Figures 8.18 and 8.19.

It can be concluded that the control structure for the lateral behaviour, that was designed in Section 4.1, successfully met all the requirements.

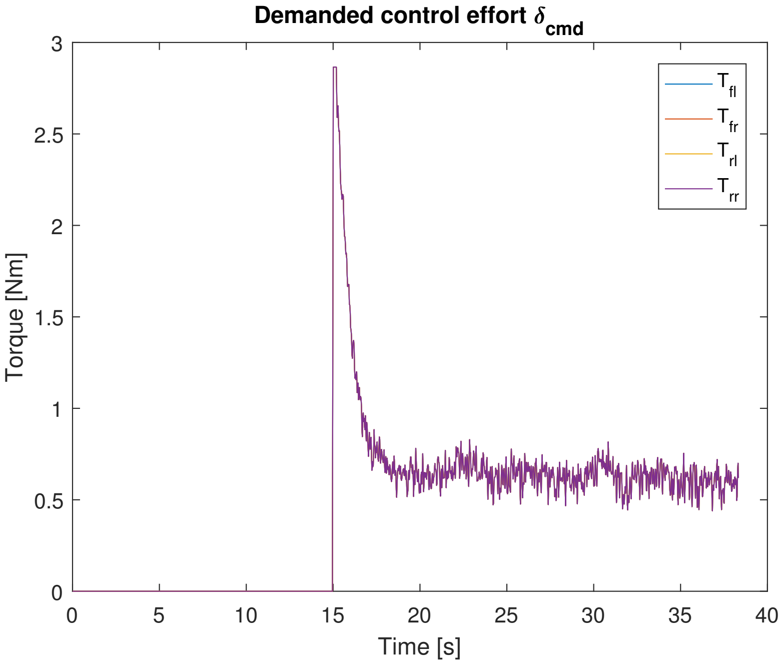


Figure 8.12: *The commanded torques generated for each wheel by the control allocation given the reference force, see Figure 8.10. The costs for using the motors was set to the same value, therefor was the contributions from the actuators equal. Note that four all lines are on exactly the same place same in this figure.*

8.4 Summary of chapter

This chapter presented how the software structure was built according to the control structure designed in Chapter 6. It also describes how the sampling time was set for the system, how sensor data was mapped and used to estimate the vehicles states, and how the torque constant was identified for the longitudinal motors. Finally this chapters shows that the control structure passes all requirements on the test scenarios that is described in Section 4.2.2.

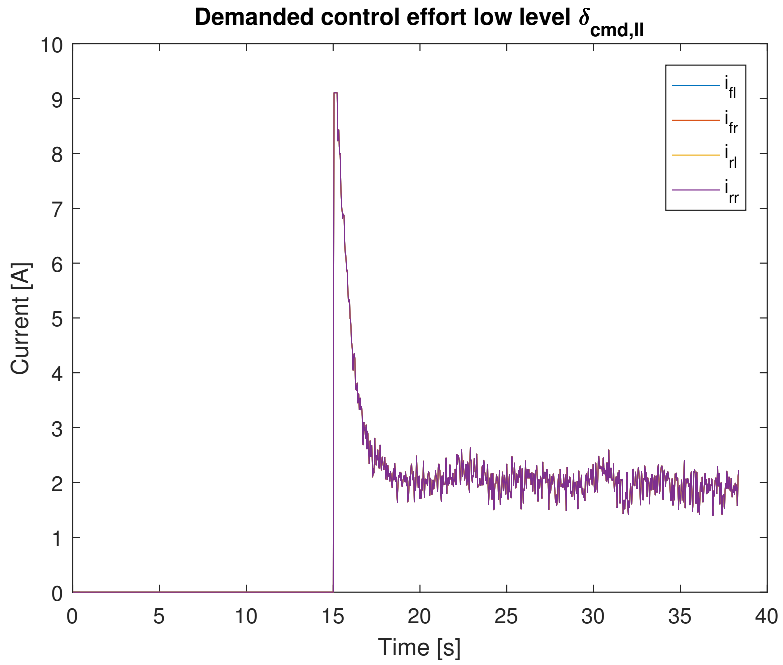


Figure 8.13: *The currents set to the motors by the motor control node given the torques set by the control allocation node in Figure 8.12. The currents here are on the same line because the torques was too*

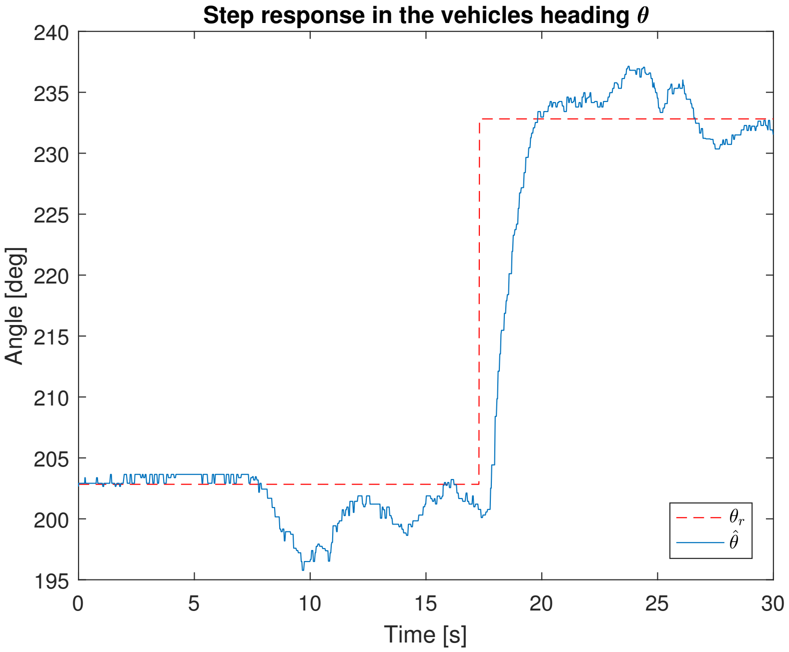


Figure 8.14: *The figure shows a response in the vehicles estimated heading $\hat{\theta}$ after a step had been generated in the vehicles reference heading θ_r. The lateral part of the high level controller is activated after approximately 7 to 8 seconds which is at the same time as the vehicle starts to accelerate. As can be seen the driveline control system successfully manages to track the reference heading despite the noise sensitive heading estimate.*

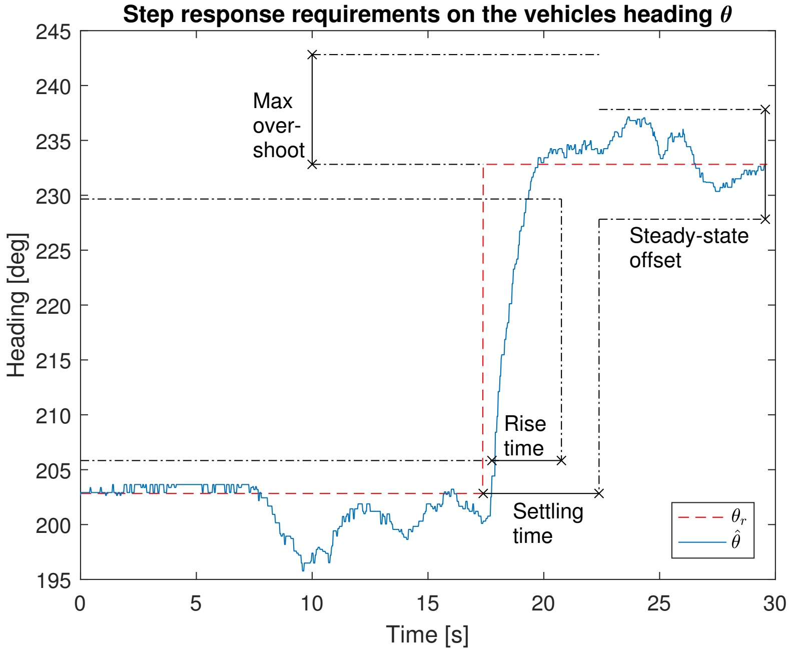


Figure 8.15: *This figure shows how the step response in the vehicles heading compares to the requirements on the performance variables. It is clear from the figure that all requirements are met and both the rise time and settling time have a big margin. The response did not get an overshoot in the normal sense and therefore was not calculated.*

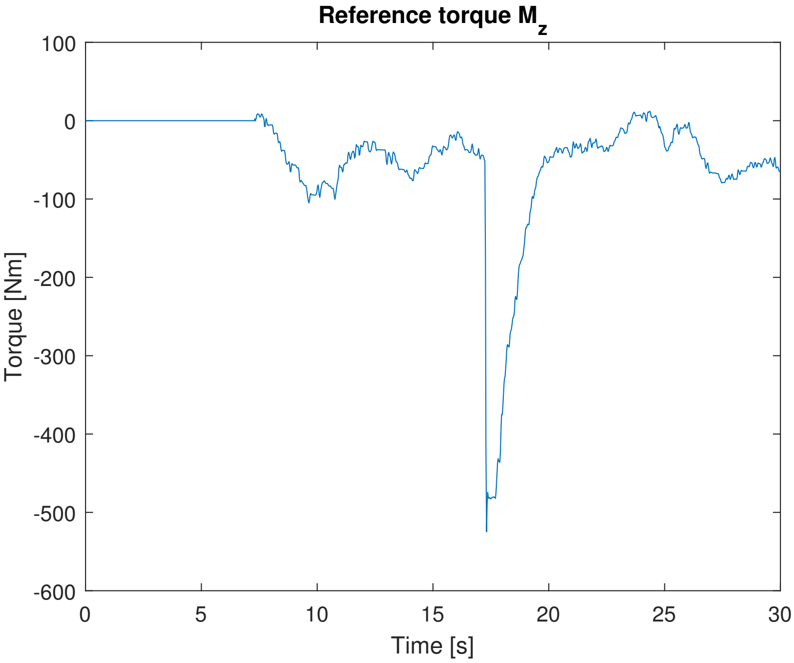


Figure 8.16: *The figure shows the reference torque M_z that was generated by the lateral high level controller. The controller was activated after approximately 7 seconds, which was at the same time as the vehicle started to accelerate. The spike that can be seen around 17 seconds is when the step in the heading occurred.*

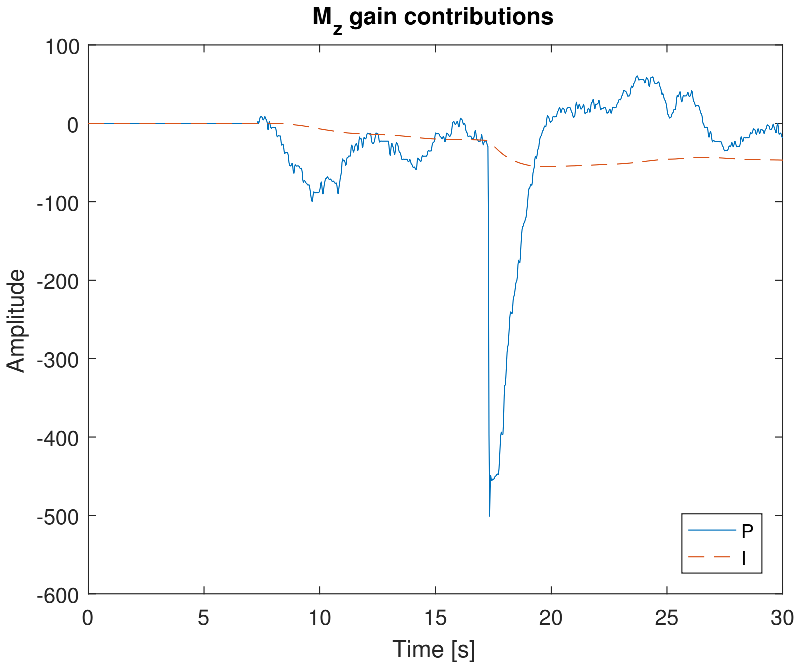


Figure 8.17: *The contributions from the proportional and integral parts from the lateral high level controller can be seen in this figure. Most of the total reference torque comes from the proportional part, which is expected as the purpose for the integral part is to remove the steady state offset.*

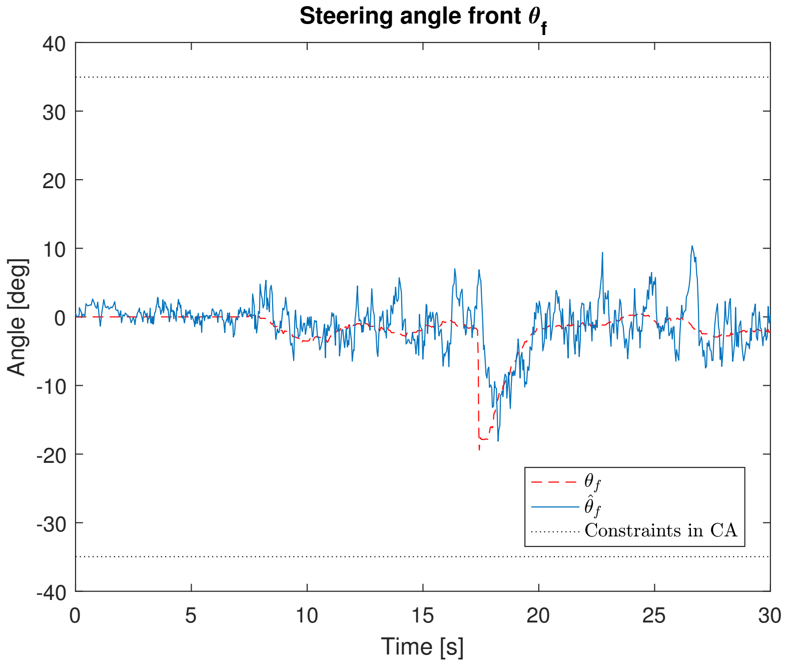


Figure 8.18: *The figure shows the demanded front steering angle θ_f from the control allocation compared to the estimated steering angle $\hat{\theta}_f$. The vehicle started to accelerate at approximately 7 seconds and it can be seen that the noise on the measurements increased when that happened. The reason for this as that the sensor that was measuring the angle was connected to the Beaglebone Black in the rear of the vehicle via an analog interface and the other electronics causes disturbances in the cable and thus readings. It is also clearly seen that the upper and lower constraints was not reached.*

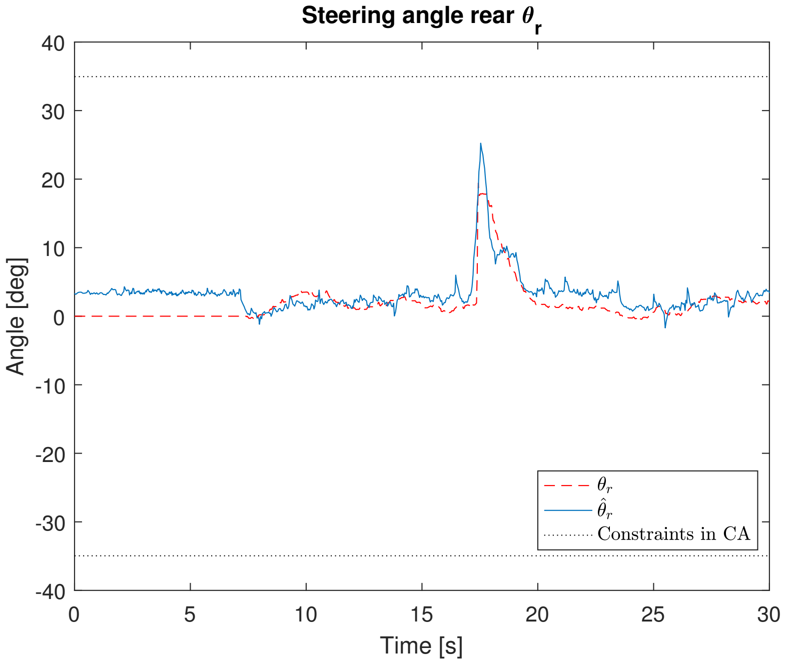


Figure 8.19: *The demanded rear steering angle θ_r from the control allocation layer compared to the estimated rear steering angle $\hat{\theta}_r$. The reason for the first offset is that the proportional controller was tuned to give good performance when the vehicle was moving and hence it cannot manage to remove the error in steering angle when the vehicle was at standstill. By comparing the noise levels in this figure with the noise levels in Figure 8.18 it is clear that the noise was far lower on the rear axle because the BBB was located close to the rear potentiometer.*

9
Discussion and conclusion

In the last chapter the driveline control system was implemented in the vehicle and the performance was evaluated by performing two different tests scenarios. This chapter starts with a short summary of what have been done in this thesis, then the different design choices are discussed. Future recommendations on how the performance of the driveline control system can be evaluated is discussed and also how the functionality and performance can be improved. Finally, it is discussed what the sustainability aspects of this work is and also how the development of autonomous vehicles can affect the society.

9.1 Outcome

The purpose of this thesis was to design, implement and evaluate a driveline control system for an electric vehicle. This vehicle was over-actuated because it had a motor for each of the four wheels and steering on both the front and rear axle. Hence, it existed many ways to determine how much effort each actuator should contribute with in order to reach the total desired effort. To make the control system modular it was divided into three layers; a high level controller that generated a longitudinal and a rotational reference force such that a reference in both the velocity and heading could be tracked. A second layer was a control allocation layer where all actuators effort was calculated by solving an optimization problem. The total combination of all actuators effort should, if possible, match the reference forces that was generated by the high level controller. A third and last layer consisted of two proportional controllers that made sure that the actual control effort matched the demanded control effort.

The main result from this thesis is that the designed control structure works well for over-actuated vehicles, this have been verified both in simulations, see Chapter 7 and by performing tests on a vehicle, Chapter 8. The performance of the closed loop system met all the requirements that were set for it, see Chapter 3.

The choice of dividing the control structure into several layers was not a necessity but it made the control system more modular. This in turn made it easy to tune all the parameters in each layer and the layers can easily be replaced, to example improve the performance or have controller schemes that can guarantee that no constraints are violated.

Given that the vehicle mostly will be used indoors it will be driving on good road conditions therefore could a simpler control allocation algorithm be used without losing any performance. For example the explicit ganging algorithm where the ratios of how much each actuator should contribute is determined beforehand would provide almost the same solution as the used QP algorithm when the vehicle´s performance is not pushed. This type of algorithm also require far less computational power compared to the QP algorithm. However, the chosen algorithm have much more potential in terms of handling rougher road conditions because it can dynamically calculate the optimal allocation of the actuator effort subjected to the weights and the limits on the actuators. This makes it easy to implement different types of traction control systems, see Section 9.2.2.

The high lever controllers used in the control structure was of PID types. Given that the desired reference signals could successfully be tracked, the steady-state offset removed and all requirements on closed loop system was fulfilled, see Chapter 7 and 8, the choice of high level controllers are deemed to be good. These types of controllers are easy to implement and require little computational power. In the literature often more advanced controller types are used for the lateral behaviour of the vehicle, often some kind of robust or predictive scheme are used.

The low level controller was used to control the front and rear steering angle in order to make sure that the demanded steering angle from the control allocation algorithm was reached. The choice of only using two proportional controller was motivated by the fact that the high level controllers would remove steady-state offsets that the vehicle could get. They also performed well enough for the closed loop system to fulfill all the requirements.

The stability of the designed control system was never investigated in an analytic manner but the result showed that the system were stable for the tested scenarios. Hence, for low speed behaviour was the longitudinal part of the control system stable and there is no reason to believe that the lateral part of the control system would become unstable for smaller steps in the reference heading than it was tested for. As it was stated earlier it is unlikely that any motion planing algorithm would give very large steps in the reference heading instead of a continuous reference, this makes it unnecessary to test and verify the stability of the system.

The vehicle was made to make it easier and faster to test new algorithms in a non idealized environments i.e. indoors and therefore would the vehicle only travel at low velocities. This naturally lead to that the control system was tuned for low speed behaviour and how the vehicle´s longitudinal and lateral behaviour was not investigated at higher velocities. Also the environment in which the vehicle has been tested have been very predictable, that is on a flat surface and the vehicles behaviour on rough terrain with unpredictable traction conditions is also not investigated. Some interesting ideas for further work that deals with this problems and extra functionality are described in Section 9.2.

9.2 Further work

This section provides ideas on how the driveline control system can be improved, some are suited for the existing vehicle but given the almost perfect operating conditions it will have during indoor use most of them can be better suited for vehicles with similar driveline systems that operates in tough conditions, for example the Volvo HX1. One interesting test to do with the driveline control system that was developed in this thesis would be to test it with a motion planning system that uses readings from the environment.

9.2.1 Slip angle estimator

In thesis it was assumed that the slip angle was equal to the steering angle, this is quite a crude approximation that affects the estimation of the lateral force that is produced by it. Having a good estimate for the lateral force can enable the control system to drive the vehicle inside specified limitations on the lateral force, this can improve the stability of the vehicle and also the performance. For this to be true a slip angle estimator would be needed and implementing one would require a more reliable way of knowing the yaw rate of the car than what the compass used could provide. However for small angles the estimation is still useful for the controller in simple scenarios that does not require the steering angles to be large.

9.2.2 Anti-slip

The formulated QP problem in the code does take into account the limits for the actuators. In the case for this thesis the limits are constant, but they does not have to be. By monitoring the individual wheel speeds at any time it is trivial to notice if a wheel starts spinning, this could especially happen in rough or slippery terrain. By storing some of the latest torques set to the motors there is possible to find out approximately the torque that is possible to put to the ground at the given time for the spinning wheel. Therefore the limits of the actuators can be set differently for all the wheels at any time and an anti-slip function can be implemented. This without loosing the performance of the vehicle as it will find a optimal solution for the new limits set for it, this will not be the case if an anti-slip function would be added to the low level controller as it does not take into account that some actuators may not be able to handle the reference torques. However implementing an anti-slip function would require faster updating frequencies of the wheel speeds than it was used in this thesis.

9.2.3 Extended virtual command

In two dimensions there is possible to not only affect the system with the forces F_{x} and M_{z}, but also sideways F_{y}. Vehicles where the actuators allows them to move sideways, as the Volvo prototype HX01 and the vehicle used in this thesis, the reachable states at any time increases. Which makes the vehicles even more flexible and could be used in a bigger variety of areas. By adding the force F_{y} to the virtual command vector $\mathbf{v}$ the control allocation could utilize the two Ackermann steering systems to move sideways. An example of how the formulation could look can be seen in Appendix C.

9.2.4 Improved effectiveness matrix

The effectiveness matrix used in this thesis was static. By using a static model of how the actuators affects the vehicle the model will differ from the reality depending on the steering angles. For example while turning the wheels the effect the wheel torques have the longitudinal force F_x will be off as long as there are a steering angle. This will not be a problem while the vehicle is tuned for similar scenarios and the high level controller can handle this errors. But when the system needs to be tuned for a wider variety of speeds and turning angles it could be a good idea to include the effect the actual steering angles have on how the torques affects the vehicle. The B matrix would therefor need to be updated continuously with the scaling from sine, cosine and the mappings in equations 5.16a-5.16d for all the wheels. By updating this, the tuning for the high level controllers will be easier for varying driving scenarios.

9.3 Social aspects

The main gain from this project has from a sustainability viewpoint been that it will make the time needed for testing and verifying different algorithms for autonomous vehicles quicker. This can therefore result in that autonomous vehicles are replacing human drivers earlier. As a result, the usage of vehicles will be more time efficient as well as energy efficient, which is an advantage for the environment.

Testing algorithms on test vehicles can be more efficient than testing them in full size trucks. The size difference of the vehicle and a truck does not restrict testing to large scale areas, which also reduces costs, and the need for nearby testing facilities for companies within the business not located near the testing areas.

When autonomous mining vehicles are put into use they can, from an economy standpoint, make it more profitable to run mines and keep them open where margins are small and possibilities for new ones to be created. This will be good for the local mining municipalities because the people that works at the mines will further increase the turnover of money.

Even though today's mining environment is good compared to a few hundred years ago it is still not an optimal environment for people to be staying in. When the drivers in mines are replaced by the autonomous vehicles, the risks related with the bad environments are removed. While some jobs will disappear, others will rise to further development, installation and maintenance.

Bibliography

[1] *RJ Mainiero, ML Harris, JH Rowland.* Vol. 1. `https://www.cdc.gov/niosh/mining/UserFiles/works/pdfs/dotff.pdf`. International Society of Explosives Engineers. Jan. 2007.

[2] Brian S. Fisher and Sabine Schnittger. *Autonomous and Remote Operation Technologies in the Mining Industry: Benefits and Costs.* BAE Report 12.1, Feb. 2012. ISBN: 978-0-9870894-1-0.

[3] Juliana Parreira. "An interactive simulation model to compare an autonomous haulage truck system with a manually-operated system". PhD thesis. University of British Columbia, 2013. DOI: `http://dx.doi.org/10.14288/1.0074111`. URL: `https://open.library.ubc.ca/cIRcle/collections/24/items/1.0074111`.

[4] Federal Motor Carrier Safety Organisation. *Large truck crash causation study analysis brief.* 2007. URL: `https://www.fmcsa.dot.gov/safety/research-and-analysis/large-truck-crash-causation-study-analysis-brief` (visited on 01/29/2016).

[5] Mathworks. *Model-Based Design.* URL: `https://se.mathworks.com/help/simulink/gs/model-based-design.html` (visited on 07/19/2017).

[6] Volvo CE. *QCAT Control Allocation Toolbox.* 2016. URL: `https://www.volvoce.com/global/en/this-is-volvo-ce/what-we-believe-in/innovation/lx1-and-hx1/` (visited on 03/14/2017).

[7] Staniforth A. Scott I. Mitchell W. "Analysis of Ackermann Steering Geometry". In: *SAE Technical Paper* 2006-01-3638 (2006), pp. 1–11. DOI: `10.4271/2006-01-3638`.

[8] Benjamin Vedder. *VESC – Open Source ESC.* 2015. URL: `http://vedder.se/2015/01/vesc-open-source-esc/` (visited on 04/26/2017).

[9] Dimension Engineering LLC. *Sabertooth dual 25A motor driver.* URL: `https://www.dimensionengineering.com/products/sabertooth2x25` (visited on 04/26/2017).

[10] Beagleboard. *BeagleBone Black.* 2017. URL: `https://beagleboard.org/black` (visited on 05/09/2017).

[11] S. Torkel Glad Ola Härkegård. "Resolving actuator redundancy - optimal control vs. control allocation". In: *Automatica* 41.1 (2005), pp. 137–144.

[12] Hans B. Pacejka and Egbert Bakker. "The magic tyre formula". In: *Vehicle System Dynamics* 21.sup001 (1992), pp. 1–18.

[13] Amir Khajepour, M. Saber Fallah, and Avesta Goodarzi. *Electric and Hybrid Vehicles: Technologies, Modeling and Control: A Mechatronic Approach.* John Wiley Sons, 2014. Chap. 9. ISBN: 9781118341513.

[14] Paolo Falcone et al. "Coordination of Motion Actuators in Heavy Vehicles using Model Predictive Control Allocation". In: *IEEE Intelligent Vehicles Symposium* 4 (2016), pp. 590–596.

[15] Mathworks. *Tire-Road Interaction (Magic Formula).* URL: `https://se.mathworks.com/help/physmod/sdl/ref/tireroadinteractionmagicformula.html` (visited on 05/29/2017).

[16] Brad Schofield. "Model-Based Vehicle Dynamics Control for Active Safety". PhD thesis. Lund University, 2008.

[17] E.H. van den Berg. "Optimal Control Allocation on Over-Actuated Vehicles". diploma thesis. Delft University of Technology, 2016.

[18] Ola Härkegård. "Dynamic Control Allocation using Constrained Quadratic Programming". In: *Journal of Guidance, Control, and Dynamics* 27.6 (2004), pp. 1028–1034.

[19] Yan Chen. "Energy-Efficient Control Allocation for Over-Actuated Systems with Applications to Electric Ground Vehicles". PhD thesis. The Ohio State University, 2013.

[20] Jonathan Brembeck and Peter Ritzer. "Energy optimal control of an over actuated Robotic Electric Vehicle using enhanced control allocation approaches". In: *Intelligent Vehicles Symposium.* Vol. 4. IEEE, 2012, pp. 322–327. DOI: `10.1109/IVS.2012.6232147`.

[21] William S. Levine. *Control System Applications.* 2nd ed. CRC Press, 2010. Chap. 8. ISBN: 978-1-4200-7360-7.

[22] Ola Härkegård. *QCAT Control Allocation Toolbox.* 2008. URL: `http://research.harkegard.se/` (visited on 03/09/2017).

[23] Mathworks. *Matlab Coder.* URL: `https://se.mathworks.com/products/matlab-coder.html` (visited on 06/24/2017).

[24] Control Guru. *Sample Time is a Fundemental Design and Tuning Specification.* URL: `http://controlguru.com/sample-time-is-a-fundamental-design-and-tuning-specification/` (visited on 06/20/2017).

A
Parameters used in the QP problem

$$\boldsymbol{B} = \begin{bmatrix} 8.70 & 8.70 & 8.70 & 8.70 & 0 & 0 \\ -3.04 & 3.04 & -3.04 & 3.04 & 773.12 & -773.12 \end{bmatrix} \tag{A.1}$$

$$\boldsymbol{\delta}_{\mathrm{d}} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}^T \tag{A.2}$$

$$\underline{\boldsymbol{\delta}} = \begin{bmatrix} -5 \\ -5 \\ -5 \\ -5 \\ -0.61 \\ -0.61 \end{bmatrix} \tag{A.3}$$

$$\overline{\boldsymbol{\delta}} = \begin{bmatrix} 5 \\ 5 \\ 5 \\ 5 \\ 0.61 \\ 0.61 \end{bmatrix} \tag{A.4}$$

$$W_\delta = \begin{bmatrix} 1000 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1000 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1000 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1000 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \tag{A.5}$$

$$W_{\mathrm{v}} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \tag{A.6}$$

$$\gamma = 1e6 \tag{A.7}$$

B

Simulink structure

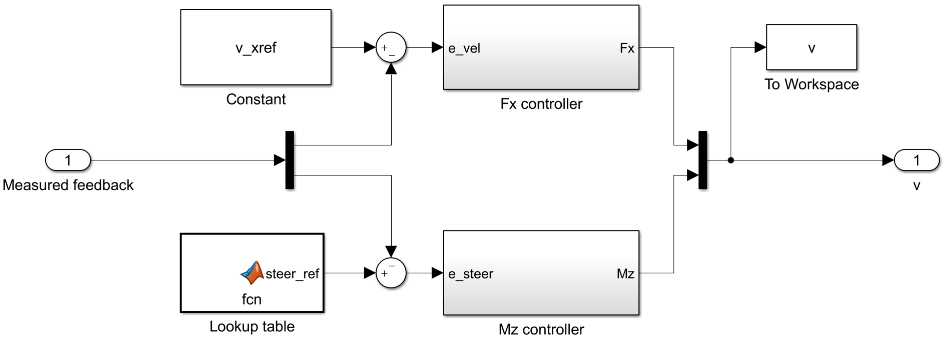


Figure B.1: *The high level controller in the Simulink structure.*

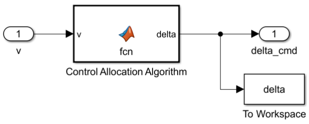


Figure B.2: *The control allocation block in the Simulink structure.*

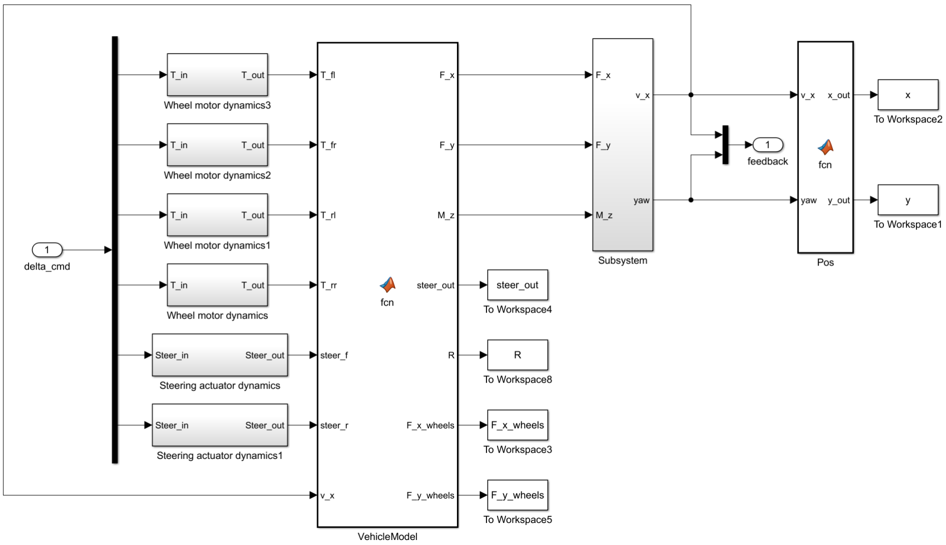


Figure B.3: *The vehicle model in the Simulink structure.*

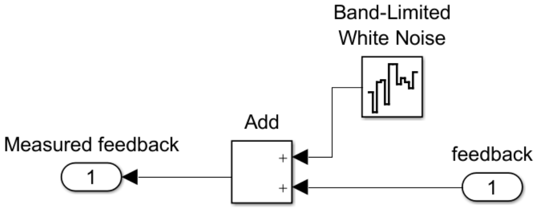


Figure B.4: *Sensor model in Simulink.*

C

Improved virtual command vector

Virtual command vectur including the force in the z direction

$$\mathbf{v} = \begin{bmatrix} F_{\mathrm{x}} & F_{\mathrm{y}} & M_{\mathrm{z}} \end{bmatrix}^{\mathbf{T}} \tag{C.1}$$

The $\boldsymbol{B}$ matrix corresponding to the new state added in the virtual command vector. The dynamics includes the sideways force generated by the steering angles.

$$\boldsymbol{B} = \begin{bmatrix} \frac{1}{r} & \frac{1}{r} & \frac{1}{r} & \frac{1}{r} & 0 & 0 \\ 0 & 0 & 0 & 0 & 2C_\alpha a & 2C_\alpha b \\ \frac{-t}{2r} & \frac{t}{2r} & \frac{-t}{2r} & \frac{t}{2r} & 2C_\alpha a & -2C_\alpha b \end{bmatrix} \tag{C.2}$$