



Hochschule für Angewandte Wissenschaften Hamburg  
*Hamburg University of Applied Sciences*

## **Bachelor Thesis**

Michel Wegner

# **Development of a vehicle dynamics control system for a driverless racing vehicle**

*Fakultät Technik und Informatik  
Department Fahrzeugtechnik und Flugzeugbau*

*Faculty of Engineering and Computer Science  
Department of Automotive and  
Aeronautical Engineering*

**Michel Wegner**

**Development of a vehicle dynamics  
control system for a driverless racing  
vehicle**

Bachelor thesis based on the examination and study regulations  
of the Bachelor of Science degree program Mechatronics  
at the Department of Automotive and Aeronautical Engineering  
of the Faculty of Engineering and Computer Science  
of the University of Applied Sciences Hamburg

Supervising examiner: Prof. Dr. Michael Erhard  
Second examiner: Prof. Dr. Wolfgang Fervers

Date of submission: 05.03.2021

**Michel Wegner**

**Title of the paper**

Development of a vehicle dynamics control system for a driverless racing vehicle

**Keywords**

Autonomous Driving, Pure Pursuit Control, PID Control, Formula Student, Vehicle Modeling, Vehicle Simulation

**Abstract**

This thesis describes the development of a vehicle control system for an autonomous Formula Student vehicle. The problem is first described in detail and several control techniques commonly used in autonomous vehicles are examined. The requirements for the vehicle control system are defined and several different concepts for longitudinal and a lateral vehicle control are proposed and compared. A PI-controller for longitudinal control and a Pure Pursuit controller for lateral control are chosen and implemented in a simulation in MATLAB/Simulink. From the simulation results it is concluded that the system achieves the defined requirements. The system is stable and accurate enough to be used at speeds of up to 15 km/h (4.2 m/s) on the racetracks of the Formula Student Germany.

**Thema der Bachelorthesis**

Entwicklung einer Fahrdynamikregelung für einen fahrerlosen Rennwagen

**Stichworte**

Autonomes Fahren, Pure Pursuit Regelung, PID Regelung, Formula Student, Fahrzeugmodellierung, Fahrzeugsimulation

**Kurzzusammenfassung**

Diese Arbeit beschreibt die Entwicklung einer Fahrdynamikregelung eines autonom fahrenden Formula Student Fahrzeugs. Das Problem wird zuerst beschrieben und mehrere Regelungstechniken, welche in autonomen Fahrzeugen Anwendung finden, werden untersucht. Die Anforderungen an die Regelung werden definiert und verschiedene Konzepte für die Längs- und Querregelung des Fahrzeugs vorgeschlagen und verglichen. Ein PI-Regler für die Längs- und ein Pure Pursuit Regler für die Querregelung werden ausgewählt und in einer Simulation in MATLAB/Simulink implementiert. Aus den Simulationsergebnissen wird schlussgefolgert, dass das System die definierten Anforderungen erfüllt. Das System ist stabil und genau genug um bei Geschwindigkeiten von bis zu 15 km/h (4,2 m/s) auf den Rennstrecken der Formula Student Germany verwendet zu werden.

# Contents

List of Figures .....	vii
List of Tables .....	x
List of Symbols .....	xi
List of Abbreviations .....	xiii
List of Appendices .....	xv
Acknowledgement .....	xvi
<b>1 Introduction .....</b>	<b>1</b>
<b>2 Problem Description and Analysis.....</b>	<b>3</b>
2.1 The Formula Student Germany .....	3
2.2 The Vehicle .....	4
2.2.1 Overall Goal and Design .....	4
2.2.2 Chassis .....	5
2.2.3 Autonomous System .....	5
2.2.4 Sensors .....	6
2.2.5 Actuators .....	6
2.2.6 Software .....	8
2.3 The Control Structure.....	8
2.4 Testing and Tuning .....	10
<b>3 Theoretical Background.....</b>	<b>11</b>
3.1 Vehicle Modelling.....	11
3.1.1 Reference Frames .....	11
3.1.2 Kinematic Modeling .....	12
3.1.3 Dynamic Modeling .....	13
3.2 Path Tracking.....	16
3.3 Control Techniques .....	17
3.3.1 PID Control .....	17
3.3.2 Feedforward Control.....	19
3.3.3 Gain-Scheduling Control .....	20
3.3.4 Pure Pursuit Control.....	21

3.3.5 Stanley Control .....	24
3.3.6 Model Predictive Control .....	26
<b>4 Requirements Analysis .....</b>	<b>30</b>
4.1 Stakeholder Analysis .....	30
4.2 Use Cases.....	31
4.2.1 Regular Driving .....	31
4.2.2 Brake Test Driving .....	32
4.3 System Requirements .....	32
4.3.1 Functional Requirements .....	33
4.3.2 Non-Functional Requirements.....	34
<b>5 Concept Development.....</b>	<b>35</b>
5.1 Evaluation and Comparison .....	35
5.1.1 Lateral Control Techniques .....	36
5.1.2 Longitudinal Control Techniques .....	37
5.2 Concept Decision.....	38
5.2.1 Control Concept .....	38
5.2.2 Vehicle Model Concept .....	39
<b>6 Implementation .....</b>	<b>40</b>
6.1 Simulation Environment and Setup .....	40
6.2 Trajectory Generation .....	41
6.3 Vehicle Model Design.....	42
6.3.1 Vehicle Body Model .....	42
6.3.2 Tire Model .....	44
6.3.3 Steering system model.....	47
6.3.4 Powertrain model .....	48
6.4 Controller Design.....	49
6.4.1 Data Analyzer .....	50
6.4.2 Lateral controller.....	51
6.4.3 Longitudinal controller.....	51
6.5 Data Visualization.....	52
<b>7 Testing and Results .....</b>	<b>53</b>
7.1 Longitudinal Controller Testing.....	53
7.2 Lateral Controller Testing.....	56
<b>8 Evaluation.....</b>	<b>61</b>
8.1 Interpretation of the Test Results .....	61

Contents

---

8.2 Review of the Requirements.....	62
<b>9 Conclusion and Outlook .....</b>	<b>65</b>
9.1 Conclusion .....	65
9.2 Outlook.....	65
<b>Bibliography .....</b>	<b>67</b>
<b>Appendix .....</b>	<b>71</b>
<b>Declaration .....</b>	<b>103</b>

# List of Figures

Figure 1: H14 at the Formula Student Germany in 2018.....	2
Figure 2: CAD of H14DV, the vehicle that is to be controlled by the vehicle control system.....	4
Figure 3: The wheelbase L and the track width W of H14DV .....	5
Figure 4: Simplified diagram of the Autonomous System [7] .....	6
Figure 5: The rear side of H14DV showing the electric drive, including the electric motor (blue), the chain drive (red), and the differential (yellow) in CAD.....	7
Figure 6: The autonomous steering actuator (orange) in CAD .....	7
Figure 7: The block diagram illustrates the control structure by showing the feedback loops for longitudinal and lateral motion control.....	9
Figure 8: Block diagram showing the feedback loop controlling the torque of the drive motor.....	10
Figure 9: Block diagram showing the cascaded feedback loop controlling the position of the steering motor .....	10
Figure 10: The orientation of the inertial frame (red) and the body frame (yellow) .....	11
Figure 11: Kinematic Bicycle Model Depiction.....	12
Figure 12: Longitudinal Vehicle Dynamic Model .....	14
Figure 13: Lateral Vehicle Dynamic Model .....	15
Figure 14: Illustration of the crosstrack error $\epsilon$ (blue) and the heading error $\theta$ (green) in the bicycle model with the reference point in the center of the front axle ...	17
Figure 15: PID Control block diagram [25] .....	18
Figure 16: Feedforward control structure block diagram [30] .....	19
Figure 17: Feedforward control block diagram [26] .....	20
Figure 18: The underlying geometry of the Pure Pursuit control law .....	22
Figure 19: Effect of look-ahead distance for small and large values [15] .....	24
Figure 20: The underlying geometry of the Stanley control law .....	25
Figure 21: Illustration of the MPC parameters: Sample time, prediction horizon, and control horizon [20].....	27
Figure 22: Model Predictive Control block diagram [19] .....	28
Figure 23: The image shows the driverless vehicle from KA-RaceIng during testing on an Autocross/Trackdrive racetrack [42] .....	31
Figure 24: The driverless vehicle from the TU Delft driverless team performing the autonomous break test at the FSG in 2019 [45] .....	32
Figure 25: Diagram comparing the performance and the development time of different types of control approaches based on the experience from AMZ. [31] .....	35

Figure 26: The longitudinal acceleration in relationship to the propulsion command of the electric Formula Student vehicle from AMZ [31].....	39
Figure 27: The highest-level of the simulation of H14DV created in Simulink .....	40
Figure 28: The created Acceleration trajectory .....	41
Figure 29: The created Skidpad trajectory .....	41
Figure 30: The created Autocross/Trackdrive trajectory .....	41
Figure 31: Trajectory data of the FSG’s 2019 Autocross/Trackdrive racetrack from AMZ ....	41
Figure 32: The subsystem models of the vehicle model.....	42
Figure 33: The air flow simulation of H14 with ANSYS .....	43
Figure 34: The curve of the magic formula (in blue) is fitted to empirical tire data (in red) by adjusting the tire parameters <b><i>Bx, Cx, Dx</i></b> and <b><i>Ex</i></b> [46] .....	45
Figure 35: The Magic Formula (blue) is fitted to the longitudinal tire data of the Continental 205/510R13 tire (red) .....	46
Figure 36: A linear function (blue) is fitted to the lateral tire data of the Continental 205/510R13 tire (red) for small slip angles .....	47
Figure 37: Simulink model of the autonomous steering system .....	48
Figure 38: Static characteristic curve of the steering system relating the steering command with the actual steering position .....	48
Figure 39: Dynamic characteristic curve of the steering system showing the progress of the steering position over time .....	48
Figure 40: Simulink model of the electric powertrain .....	49
Figure 41: The vehicle control block in Simulink.....	50
Figure 42: The data analyzer block in Simulink.....	51
Figure 43: The lateral controller block in Simulink .....	51
Figure 44: The longitudinal controller block in Simulink .....	52
Figure 45: The Data Visualization Section.....	52
Figure 46: The vehicle’s reference and actual speed are plotted over time on the Acceleration racetrack.....	53
Figure 47: The vehicle’s reference and actual speed are plotted over time on the Acceleration racetrack.....	54
Figure 48: The vehicle’s reference and actual speed are plotted over time on the Acceleration racetrack.....	54
Figure 49: The vehicle’s reference and actual speed are plotted over time on the Skidpad racetrack .....	55
Figure 50: The vehicle’s reference and actual speed are plotted over time on the Autocross/Trackdrive racetrack at varying reference speeds, with $K_p = 1.5$ , $K_i = 1.5$ , lap time = 102 s.....	55
Figure 51: Simulation of the autonomous brake test at $v_{ref} = 40.0$ km/h (11.1 m/s), with $K_p = 0.5$ , $K_i = 0.1$ .....	56
Figure 52: The crosstrack error is plotted over the distance of the Autocross/Trackdrive track.....	57



Figure 53: The crosstrack error is plotted over the distance of the Autocross/Trackdrive track.....	57
Figure 54: The crosstrack error is plotted over the distance of the Autocross/Trackdrive track.....	58
Figure 55: The crosstrack error is plotted over the distance of the Autocross/Trackdrive track.....	58
Figure 56: The crosstrack error is plotted over the distance of the Skidpad racetrack.....	59
Figure 57: The crosstrack error is plotted over the distance of the Acceleration track .....	60
Figure 58: Simulation of the autonomous brake test at $v_{ref} = 40.0$ km/h (11.1 m/s), with $K_{ld} = 1.1$ .....	60

# List of Tables

Table 1: The effects of increasing the independent PID control gains on the system response [24].....	19
Table 2: List of stakeholders interested in the vehicle control system.....	30
Table 3: Functional Requirements for the vehicle control system .....	33
Table 4: Non-functional Requirements for the vehicle control system.....	34
Table 5: The summarized characteristics and results of the evaluation of control techniques for lateral control.....	37
Table 6: The summarized results of the evaluation of control techniques for longitudinal control .....	38
Table 7: Vehicle Parameters .....	43
Table 8: The determined longitudinal and lateral tire model parameters [43].....	47
Table 9: Estimation of the steering system parameters .....	48
Table 10: Estimation of the drive motor parameters .....	49

# List of Symbols

Symbol	Unit	Description
$A, B$	-	State space matrices
$a$	m	Horizontal distance from the vehicle COG to the front wheel axle
$b$	m	Horizontal distance from the vehicle COG to the rear wheel axle
$B_x, C_x, D_x, E_x$	-	Empirical tire parameters
$C_{set}$	-	Reference point on the vehicle
$C_f, C_r$	N·rad <sup>-1</sup>	Front and rear tire's cornering stiffness factor
$F_{aero}$	N	Aerodynamic forces
$F_{gx}$	N	Gravitational force in longitudinal direction
$F_{xf}, F_{xr}$	N	Longitudinal front and rear tire forces
$F_{yf}, F_{yr}$	N	Lateral front and rear tire forces
$F_{zf}, F_{zr}$	N	Front and rear tire normal force
$G_{steer}$	-	Transfer function of the Steering system actuator
$I_{zz}$	kg·m <sup>2</sup>	Moment of inertia about z-axis
$i$	-	Discrete steering angle index
$K_S$	-	Proportional gain of the Steering system actuator
$K_d$	-	Derivative gain
$K_{ld}$	-	Lookahead scaling constant
$K_i$	-	Integral gain
$K_p$	-	Proportional gain
$k_{S,p}$	-	Proportional gain of the Stanley controller
$k_{d,steer}$	-	Steering rate gain
$k_{d,yaw}$	-	Yaw rate gain
$k_{soft}$	-	Softening constant
$k$	-	Current MPC time step
$L$	m	Wheelbase
$l_d$	m	Look-ahead distance
$l_{dmin}, l_{dmax}$	m	Minimum and maximum look-ahead distance
$l_f$	m	Distance between vehicle reference point and front axle

$l_r$	m	Distance between vehicle reference point and rear axle
$m$	kg	Total mass of the vehicle
$Q, R$	-	Weight matrices
$R$	m	Distance from vehicle's COG to point of ICR
$R_f$	m	Distance from vehicle's center of front axle to point of ICR
$R_r$	m	Distance from vehicle's center of rear axle to point of ICR
$R_{xf}, R_{xr}$	N	Front and rear rolling resistance forces
$r_{tf}, r_{tr}$	m	Front and rear tire radius
$T_S$	-	Lag time constant of the Steering system actuator
$U$	-	Set of control input vectors
$u_t$	-	State space input vector
$v$	$m \cdot s^{-1}$	Velocity vector of vehicle
$v_x$	$m \cdot s^{-1}$	X-component of velocity vector of vehicle
$v_y$	$m \cdot s^{-1}$	Y-component of velocity vector of vehicle
$X, Y, Z$	m	Cartesian coordinates in inertial reference frame
$X_{tp}, Y_{tp}$	m	Coordinates of the target point in the inertial frame
$X_{vh}, Y_{vh}$	m	Coordinates of the vehicle's current position in the inertial frame
$x, y, z$	m	Cartesian coordinates in body reference frame
$x_t$	-	State space state vector
$x_{t+1}$	-	State space future state vector
$\alpha_f, \alpha_r$	rad	Front and rear tire slip angles
$\beta$	rad	Slip angle
$\delta$	rad	Steering angle
$\dot{\delta}$	$rad \cdot s^{-1}$	Steering rate
$\delta_{meas}$	rad	Measured steering angle
$\delta_{min}, \delta_{max}$	rad	Minimum and maximum steering angle
$\delta_{traj}$	rad	Steering angle trajectory
$\varepsilon$	m	Crosstrack error
$\theta$	rad	Heading error angle

# List of Abbreviations

<b>Abbreviation</b>	<b>Definition</b>
2D	Two-dimensional
3D	Three-dimensional
AMZ	Akademischer Motorsportverein Zürich
ANSYS	Analysis System
AS	Autonomous System
ASCU	Autonomous System Control Unit
Back EMF	Back Electromotive Force
BLDC	Brushless Direct Current electric motor
C, C++	Programming language
CAD	Computer-aided design
COG	Center of Gravity
CV	Internal Combustion Engine Vehicle
CPU	Central Processing Unit
DARPA	Defense Advanced Research Projects Agency
DCU	Driverless Compute Unit
DV	Driverless Vehicle
EBS	Emergency Brake System
EV	Electric Vehicle
FSG	Formula Student Germany
GNSS	Global Navigation Satellite System
ICR	Instantaneous Center of Rotation
ID	Identification
IMU	Inertial Measurement Unit
LiDAR	Light Detection and Ranging Sensor
LQR	Linear Quadratic Regulator

MATLAB	Matrix Laboratory
MIMO	Multi-Input Multi-Output
MPC	Model Predictive Control
PID	Proportional–Integral–Derivative control
PMSM	Permanent Magnet Synchronous Motor
$PT_1$	First order lag element
RAM	Random-Access Memory
RL	Reinforcement Learning
ROS	Robot Operating System
SACU	Steering Actuator Control Unit
SB	Service Brake system
TSMC	Tractive System Motor Controller
TSMU	Tractive System Management Unit
TU	Technical University

# List of Appendices

A.1	Intel NUC Specifications [5, P. 4] .....	71
A.2	Diagram of the Autonomous System [7] .....	72
A.3	Racetrack Acceleration Discipline [3, P.13].....	73
A.4	Racetrack Skidpad Discipline [3, P.15] .....	73
A.5	Racetrack Autocross and Trackdrive Disciplines [3, P.14].....	74
A.6	Electric Drive Motor Datasheet [8] .....	75
A.7	Electric Drive Motor Controller Datasheet [35].....	77
A.8	Electric Steering Motor Datasheet [34] .....	78
A.9	Electric Steering Motor Controller Datasheet [34].....	79
A.10	Total Mass and Moment of Inertia Calculation Table H14DV.....	80
A.11	Simulink “Vehicle-Body-3DOF-Block” Documentation [38].....	82
A.12	Simulink “Longitudinal-Wheel-Block” Documentation [40] .....	91
A.13	Longitudinal tire data [40] .....	96
A.14	Lateral tire data [40] .....	96
A.15	Source code of the function blocks used in the data analyzer.....	97
A.16	Simulation MATLAB Setup Script .....	98
A.17	MATLAB code used for fitting the longitudinal tire data to the magic formula.....	99
A.18	MATLAB code used for fitting the lateral tire data to the linear function .	100
A.19	Simulation Setup.....	101
A.20	Compact disc containing Bachelor thesis as PDF-Format.....	102

# Acknowledgement

I would like to thank Prof. Dr. Michael Erhard and Prof. Dr. Wolfgang Fervers for the guidance and advice they gave me throughout the process of writing this thesis. I greatly appreciate your expertise and your commitment towards me and this project.

I would like to acknowledge my colleagues from the HAWKS Racing team. In particular, I would like to thank Philip Lohde and Helen Smaleit, who helped me to set up an air flow simulation of our vehicle. Also, I would like to thank Moritz Höwer and Leon Schwarz for their constructive feedback regarding my work.

Finally, I would also like to thank my parents, Anja Wegner and Markus Mohr, for their long-time support throughout my studies.



# 1 Introduction

Self-driving cars have become an increasingly prominent topic in the recent decade. The technology promises to improve safety, efficiency, access, and convenience of automotive vehicles. Also, it has the potential to lower the overall cost of transportation significantly. According to a report by Statista, one in ten vehicles globally will be fully self-driving by the year 2030. Robo-taxis are expected to become the top use case for driverless cars and the market volume of fully automated cars is predicted to grow to 13.7 billion U.S. dollars by 2030. Big technology companies and automakers like Waymo (Google), Argo AI (Volkswagen, Ford), and Tesla invest heavily into the research of this technology. The overall automotive startup funding has increased ten-fold between 2013 and 2018, reaching a record-breaking 27.5 billion U.S. dollars in 2018 alone. These companies also support student competitions, like the Formula Student, to excite the next generation of engineers and computer scientists about autonomous driving and aim to increase the pool of skilled engineers. [1][33]

The Formula Student Germany (FSG) is an international engineering design competition for university students. The goal of the competition is to design, manufacture and assemble a formula-style racing vehicle. The teams then compete with their self-build vehicles in a weeklong event at the Hockenheimring in August annually. The teams can participate in the Formula Student Germany with internal combustion engine vehicles (CV) and electric vehicles (EV). Both types of drive systems have their own class. Since 2017, a third class exists, the driverless vehicle (DV) class. In this class, instead of the human driver, the vehicle must drive fully autonomously with the use of sensors, actuators, and computers. Both, CV and EV vehicles, are allowed to participate in the DV class and compete against each other. Although the vehicles must drive fully autonomously in the competition, a driver must still be able to operate the vehicle manually from the inside of the vehicle. Therefore, the steering wheel, seat, and pedal box cannot be removed from the vehicle. [3][4]

The Formula Student team of the University of Applied Science in Hamburg, HAWKS Racing, defined the goal to build their first electric and autonomous vehicle to participate in the Formula Student Events 2021. To build their first electric and autonomous vehicle, the team decided to use the monocoque and the suspension of the vehicle from the 2018 season and slightly modify them. Not having to design these parts entirely new allows to shift the focus to the autonomous and electronic components. The 2018 vehicle is called H14 and can be seen in Figure 1. The new driverless vehicle is subsequently called “H14DV” and is equipped with several sensors, actuators for autonomous braking and steering, a compute unit and multiple control units. The “brain” of the vehicle is the high-level software, which is split into three modules. These modules are state estimation, motion planning, and vehicle control.

## 1 Introduction

---

This thesis deals with the development of the vehicle control module which controls the motion of the vehicle and ensures that the vehicle tracks a given trajectory. It is the highest level of control and provides the reference commands to the low-level controllers. The objective of this thesis is to develop a foundation for the vehicle control system used in the autonomous racing vehicle H14DV and to lay the groundwork for future autonomous racing vehicles developed by HAWKS Racing. To achieve this objective, first the problem will be discussed and analyzed in detail in chapter 2. Then, the fundamentals of vehicle modelling and the state of the art of vehicle control will be described chapter 3, by surveying multiple control techniques used in autonomous vehicles. Chapter 4 will identify the stakeholders and define the requirements for the system. Chapter 5 will evaluate the presented techniques and will develop several concepts for longitudinal and lateral control of the vehicle. The best suited concepts for the vehicle control system in H14DV will be chosen and implemented MATLAB/Simulink in Chapter 6. The implemented vehicle control system will be tested in simulation and the results will be presented in chapter 7. Chapter 8 will evaluate the results and compare the outcome to the requirements. Finally, chapter 9 will summarize the significant results and will provide an outlook on how the project is planned to be continued.



Figure 1: H14 at the Formula Student Germany in 2018

## 2 Problem Description and Analysis

This chapter will describe and analyze the control problem in detail. It will first discuss the framework conditions of the Formula Student competition in section 2.1. Then, it will take a closer look at the vehicle that is to be controlled in section 2.2. Afterwards, it will introduce the general structure of the control system that is to be developed in section 2.3 and will briefly discuss the issue of testing and tuning the control system in section 2.4.

### 2.1 The Formula Student Germany

The Formula Student Germany event will be held in Hockenheim, Germany in August 2021. At the beginning of the event, each vehicle must pass technical inspection, before it is allowed to participate in any discipline. Each vehicle must satisfy the rules set by the official rule book, which is published by the FSG. During the technical inspections, official scrutineers check that the vehicle is rules compliant and perform several tests to ensure functionality and safety. The most critical test for the vehicle control system is the autonomous brake test. A driverless vehicle can participate four disciplines. These are Acceleration, Skidpad, Autocross, and Trackdrive. The racetracks are marked by colored cones in all four disciplines. Yellow cones mark the right and blue cones mark the left side of the track. Furthermore, small and big orange cones mark the start and stop areas. The Acceleration discipline, illustrated in A.3, is a simple acceleration race over 75 m on a straight-line racetrack. The vehicle must come to a stop within a marked area after crossing the finish line. In the Skidpad discipline, the vehicle must complete four laps on an “eight”-shaped path, depicted in A.4. It must complete two laps in the right circle, followed by two laps in the left circle, before coming to a full stop in the marked area. The Autocross and Trackdrive disciplines consist of the same closed loop track, shown in A.5. In the Autocross discipline, the vehicle must complete one full lap on a prior unknown track. In the Trackdrive discipline, the vehicle must complete ten full laps on the same track. In both disciplines, the vehicle must come to a complete stop after crossing the finish line. Also in the Trackdrive discipline, the vehicle must reach an average minimum speed of 12.6 km/h (3.5 m/s) in each lap after completing the third lap. In all four disciplines, only a single vehicle is allowed to be on the racetrack at a time. Each team has at least two attempts for each discipline. Furthermore, the driverless vehicle must be able to operate on dry and wet road conditions and must be capable of handling weather conditions, such as rain and wind. The road surface of the racetracks will

be asphalt and will be nearly flat with no serious road bumps. During the race, there will be no people or obstacles on the racetrack, besides the cones used to mark the sides of the racetrack. However, cones which are relocated by contact with the vehicle during the race can become possible obstacles. [3]

### 2.2 The Vehicle

The plant of the control system is the vehicle H14DV itself, shown in computer-aided design (CAD) in Figure 2. This section will discuss the most important features of the vehicle in order to better understand the system that is to be controlled.

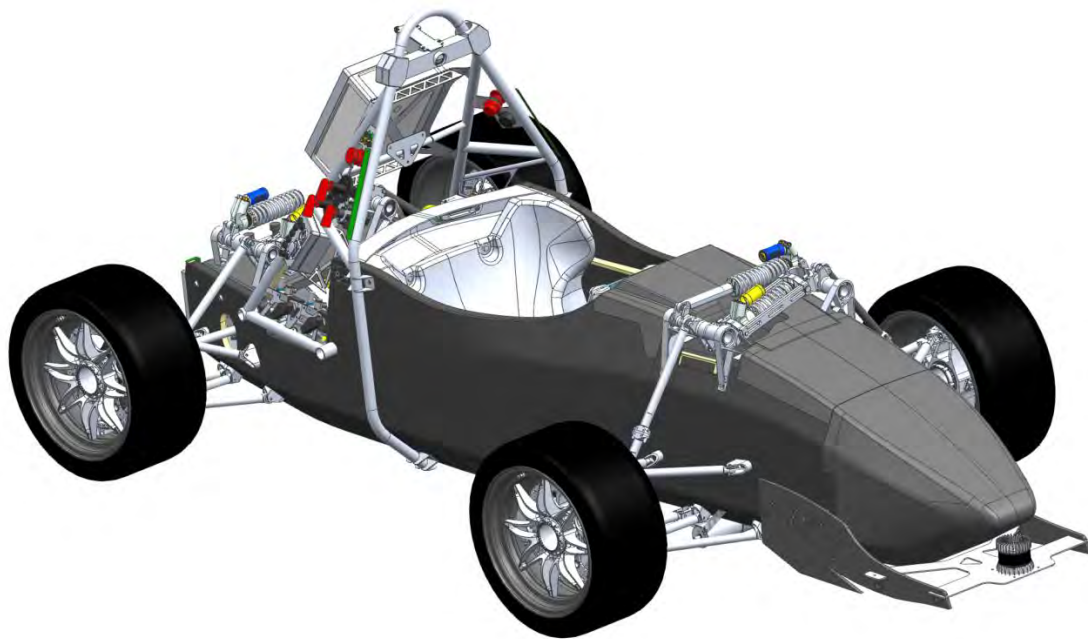


Figure 2: CAD of H14DV, the vehicle that is to be controlled by the vehicle control system

#### 2.2.1 Overall Goal and Design

The HAWKS Racing team decided that the overall goal of H14DV is not to be a competitive racing car, but to serve as a practice vehicle, in order to develop the knowledge and skills needed to build competitive electric and autonomous racecars in the future. This practice vehicle shall pass the rigorous scrutineering at the beginning of the Formula Student event and complete all four disciplines, described in section 2.1. The time in which the vehicle finishes the disciplines is not of importance. Because building an electric and autonomous racecar for the first time is not an easy task, the team decided to approach this challenge with the mindset of “keep it simple, make it work, make it better”.

The vehicle is designed to be powered by a 48-Volt accumulator. The low voltage power system is chosen over a high voltage power system due to safety reasons and fewer required resources. The aerodynamic components are also removed from the vehicle in order to fit the newly added autonomous components. H14DV is designed to reach a minimum top speed of 45 km/h (12,5 m/s). However, because the time in which the vehicle completes the disciplines is not of importance, the vehicle will travel slower in the actual disciplines.

### 2.2.2 Chassis

The main part of the chassis is the monocoque made of a composite of carbon fiber and aluminum honeycombs. This combination of materials achieves in a very high level of stiffness and strength while being light in weight. The monocoque is spring-suspended using a roll-heave-decoupled suspension system. The vehicle uses 205/470R13-tires from Continental. The wheelbase  $L$  and the track width  $W$  of the vehicle are shown in Figure 3.

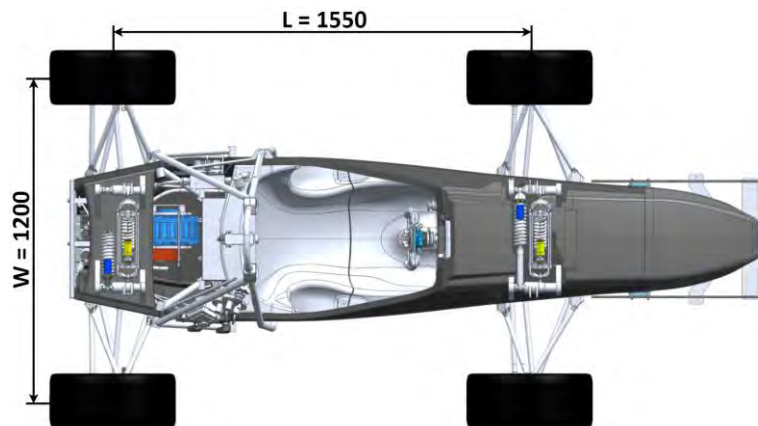


Figure 3: The wheelbase  $L$  and the track width  $W$  of H14DV

### 2.2.3 Autonomous System

The Autonomous System (AS) is the electronic backbone of the vehicle. It comprises all of the electronic components that allow H14DV to function fully autonomously. This includes several control units that control the individual subsystems, a compute unit that runs the high-level software, and several sensor nodes that supply the sensor data. The development of the AS is described in [7]. A simplified diagram of the Autonomous System is shown in Figure 4. Also, a more detailed version containing all of the components of the Autonomous System is shown in A.2. One of the main objectives of the Autonomous System is to ensure safety, by constantly monitoring the vehicle and controlling the actuators on a low-level. The Autonomous System is controlled by the Autonomous System Control Unit (ASCU). It functions as the center of the system and supervises all processes. The Driverless Compute

## 2 Problem Description and Analysis

Unit (DCU) is used to run the high-level software. This job is performed by an Intel NUC NUC8i7BEH whose specifications can be found in A.1. [7]

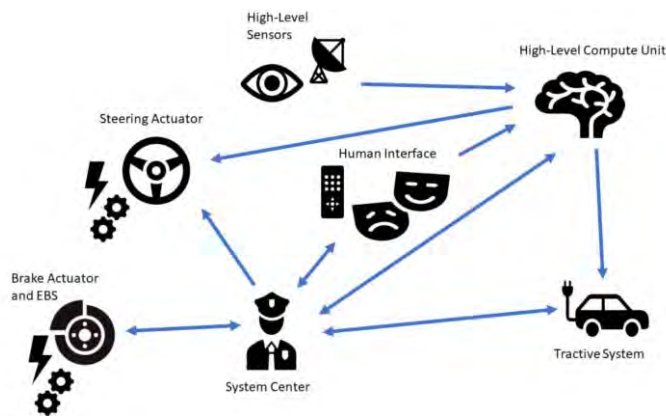


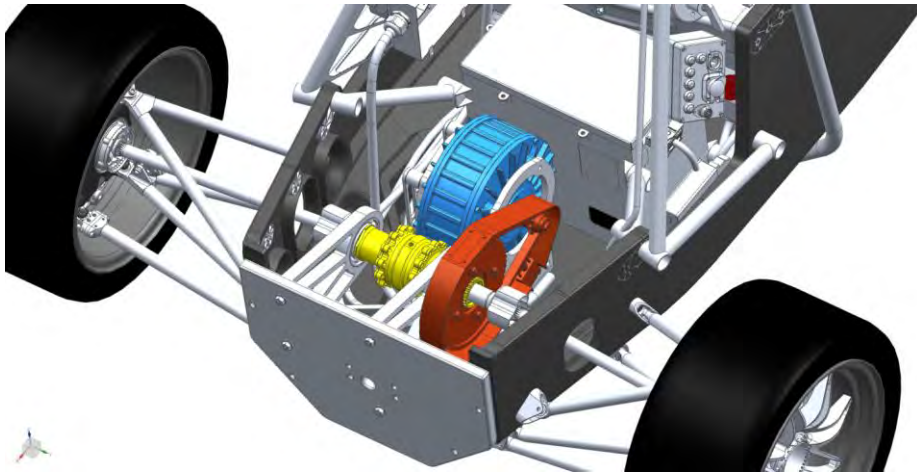
Figure 4: Simplified diagram of the Autonomous System [7]

### 2.2.4 Sensors

The vehicle is equipped with several sensors in order to sense its surrounding and to localize itself within the environment. For perception, a 3D-Light Detection and Ranging (LiDAR) sensor is mounted to the front of the monocoque and a forward-facing camera is attached to the top center of the vehicle. A high-precision 2-axis optical sensor is used to measure the vehicle's speed and angle of rotation. The vehicle is also equipped with a Global Navigation Satellite System (GNSS) sensor with dual antenna and a 9-axis Inertial Measurement Unit (IMU) sensor, which includes gyroscopes and accelerometers. Moreover, wheel speed, steering angle, and brake pressure sensors are used in the vehicle. More information about the sensors can be found in [36].

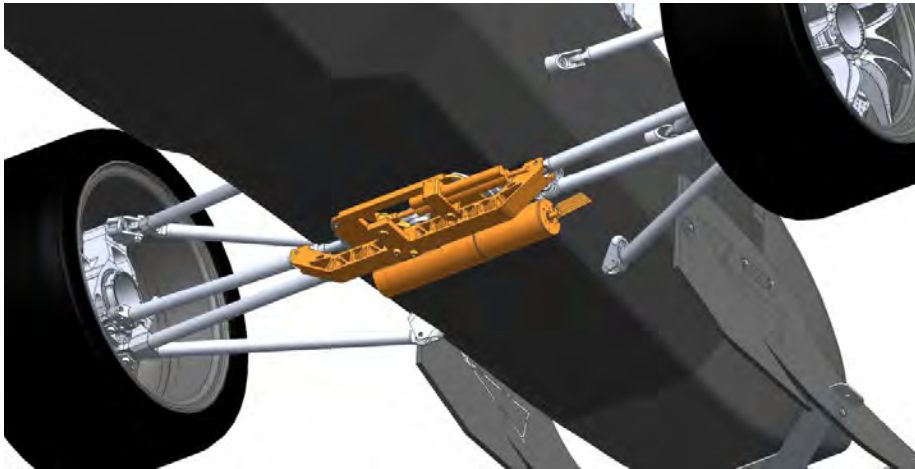
### 2.2.5 Actuators

The vehicle is propelled by a single electric motor which is located close to the rear axle, shown in Figure 5. The motor is connected via a chain drive to a limited slip differential on the rear axle. The electric motor is a permanent magnet synchronous disc motor (PMSM) from the Heinzmann GmbH & Co. KG and is designed for a nominal 48 Volt DC-bus-voltage. It delivers a nominal torque of 17,5 Nm at a rotational speed of 3000 rpm. It is rated for a maximum input power of 11,1 kW, a maximum torque of 45 Nm, and a maximum rotational speed of 3840 rpm. The motor is controlled by a Roboteq GBL2660S motor controller. The data sheets of the motor and the controller can be found in A.6 and A.7.



**Figure 5: The rear side of H14DV showing the electric drive, including the electric motor (blue), the chain drive (red), and the differential (yellow) in CAD.**

The vehicle is steered by the autonomous steering system, which is mounted below the vehicle to the left and right lower tie bar of the existing steering system, as shown in Figure 6. An electric BLDC motor rotates a ball screw drive, which converts the rotational motion of the motor into the linear steering motion. The actuator is controlled by a Maxon EPOS 4 50/15 motor controller. The data sheets of the motor and the controller can be found in A.8 and A.9.



**Figure 6: The autonomous steering actuator (orange) in CAD**

The vehicle uses two different types of brake systems, a Service Brake System (SB) and an Emergency Brake System (EBS). The service brake uses the electric drive as a generator to convert the kinetic energy from the vehicle's motion into electrical energy, and as a result reducing the speed of the vehicle. It is used in regular driving situations and helps to control the speed of the vehicle. The EBS is only used in safety critical situations. It uses a pneumatic

actuator to create the brake pressure, which is applied via a hydraulic circuit to the disk brakes of the vehicle. To ensure safety, the EBS is controlled directly by the ASCU.

### 2.2.6 Software

The software is split into low- and high-level software. The low-level software runs on several different control units and controls specific individual components. The high-level software is concerned with the vehicle as a whole and is divided into three modules: State estimation, motion planning, and vehicle control.

The task of the state estimation module is to compute the current state of the vehicle and the environment. To achieve this, the state estimation fuses the data from the different sensors, maps the surrounding with the fused data, and localizes the vehicle within the map. The state estimation module outputs an estimation of the vehicle's current state to the motion planning module and vehicle control module. The motion planning module uses a map of the environment and the current state of the vehicle to calculate a proper plan of motion for the vehicle through the racetrack. This plan of motion is called "trajectory" and consists of a reference path and a speed profile. The reference path is comprised of a set of target waypoints, which are defined in 2D-coordinates. The speed profile is a list that defines an appropriate target speed for each waypoint, depending on the curvature of the track and the driving conditions. The trajectory is fed to the vehicle control module. The vehicle control module controls the motion of the vehicle and ensures that the vehicle tracks a given reference trajectory. It is the highest level of control and contains several cascaded loops, which control the actuators on a low-level. The module provides the appropriate reference commands to the low-level controllers, so that the vehicle can carry out the planned motion. Most parts of the high-level software are not developed yet. Therefore, these are only assumptions of how the software is planned to be implemented.

The high-level software will be deployed to the DCU. The team uses the open-source middleware Robot Operating System (ROS) to manage all the high-level software processes and the interactions between the processes. ROS splits the software into multiple parts, called nodes. Each node runs its own process and exchanges data with the other nodes via, so called, topics. The vehicle control module provides the vehicle control node. The node will be launched when the system state for autonomous driving mode is activated, which is managed by the ASCU.

## 2.3 The Control Structure

The vehicle control problem can be split into the tasks of controlling the vehicle's longitudinal and lateral motion. Thus, the vehicle control module can be split into individual submodules for longitudinal and lateral control. The longitudinal control submodule produces the



## 2 Problem Description and Analysis

required propulsion commands which keep the vehicle's longitudinal speed close to the reference speed. The lateral control submodule produces the required steering commands which keep the vehicle on the reference path. The vehicle control module forms two closed feedback control loops when combined with the rest of the high-level software, the actuators, and the sensors. The block diagram in Figure 7 illustrates the control structure.

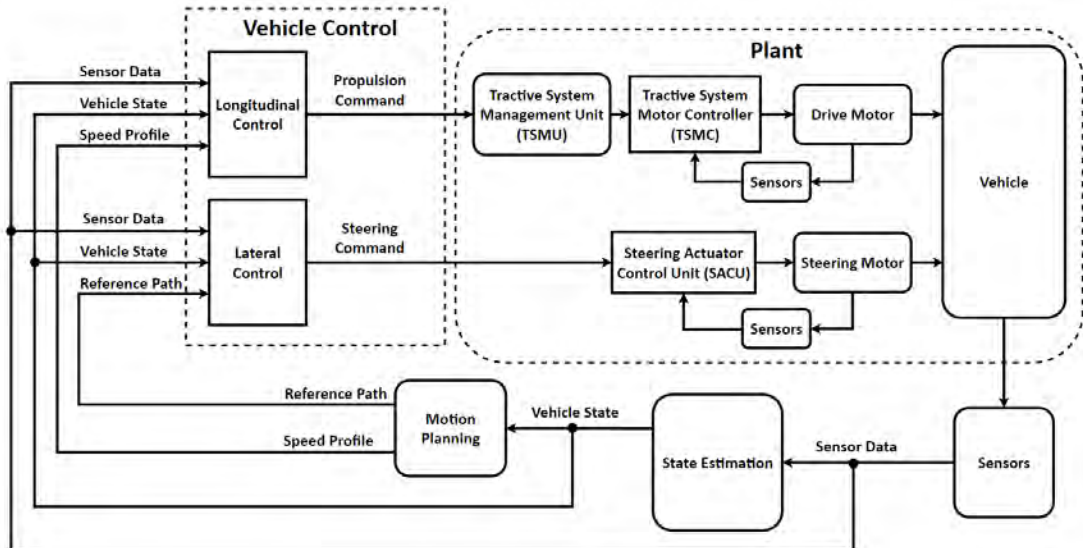


Figure 7: The block diagram illustrates the control structure by showing the feedback loops for longitudinal and lateral motion control.

The vehicle control module is the bridge between the high-level software and the actuators. It receives inputs from the state estimation and the motion planning modules, as well as several direct sensor measurements. The state estimation provides the current state of the vehicle, which includes the vehicle's current position in the form of its 2D-coordinates and its yaw angle. The motion planning module provides a discrete reference trajectory, in the form of a reference path and a speed profile. The outputs of the vehicle control module are the propulsion and steering commands. These commands are sent via the vehicle's Controller-Area-Network (CAN) bus to the particular low-level control units. The propulsion commands are transferred through the Tractive System Management Unit (TSMU) to the Tractive System Motor Controller (TSMC), which controls the rotational speed of the electric drive motor. The steering commands are fed to the Steering Actuator Control Unit (SACU), which controls the electric steering motor.

Figure 7 shows that both actuators are controlled using a cascaded control structure. The TSMC and the drive motor as well as the SACU and the steering motor each form an inner loop inside their respective feedback loop. The first inner loop controls the torque of the drive motor  $\tau_{D,act}$  using a current sensor and a PID controller, shown in Figure 8. The second inner loop controls the position of the steering motor  $\varphi_{S,act}$  with a PID controller and an

incremental encoder sensor, shown in Figure 9. The steering motor control loop is cascaded further, controlling the torque  $\tau_{S,act}$  and the velocity  $\dot{\varphi}_{S,act}$  of the actuator using PI-controllers. [37]

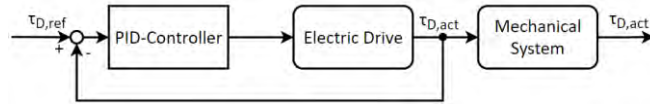


Figure 8: Block diagram showing the feedback loop controlling the torque of the drive motor

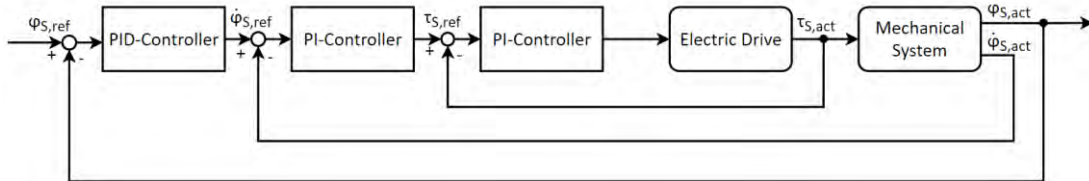


Figure 9: Block diagram showing the cascaded feedback loop controlling the position of the steering motor

## 2.4 Testing and Tuning

The work on H14DV is currently still in progress. Essential components like the electric drivetrain and the accumulator are not built into the vehicle at this point in time. Therefore, it is not possible yet to test and tune the vehicle control system using the real vehicle. Once the vehicle is complete, the time available for testing will also be limited. Thus, testing and tuning the vehicle control module entirely on the real vehicle is infeasible. To overcome these issues, a model-based testing and tuning approach can be used. This allows for rapid testing and tuning of the vehicle control system, prior to expensive tests with the real vehicle. However, this approach requires a mathematical model that represents the motion behavior of the vehicle and a simulation environment to simulate the control process. Unfortunately, HAWKS Racing has neither an appropriate model of the entire vehicle, nor a vehicle simulation. Therefore, to develop the vehicle control module requires first the development of a simulation as well as a vehicle model.

# 3 Theoretical Background

This chapter shall provide the theoretical knowledge that is needed to develop the vehicle control system. It will describe the essentials of kinematic and dynamic vehicle modeling in section 3.1. Then, it will introduce two types of path tracking errors in section 3.2. Finally, it will provide the state of the art of vehicle control, by presenting several different control techniques commonly used in autonomous vehicles in section 3.3.

## 3.1 Vehicle Modelling

For model-based control development a proper vehicle model is pivotal. Mathematically modeling the motion of a vehicle is commonly achieved by using kinematic and/or dynamic equations.

### 3.1.1 Reference Frames

To model the motion of a vehicle the two reference frames, shown in Figure 10, will be used. The inertial reference frame, in red, is fixed to the earth with the cartesian coordinates  $X$ ,  $Y$ , and  $Z$ . Its  $Z$ -axis points anti-directional to the vertical gravitational vector. The body reference frame, in yellow, is attached to a fixed reference point on the vehicle. This fixed reference point can, for example, be the vehicle's center of gravity, as it is the case in H14DV. The body reference frame uses the cartesian coordinates  $x$ ,  $y$ , and  $z$ . Its  $x$ -axis points in the vehicle's forward direction, its  $y$ -axis to the vehicle's left side, and its  $z$ -axis in the vehicle's top direction. [2]

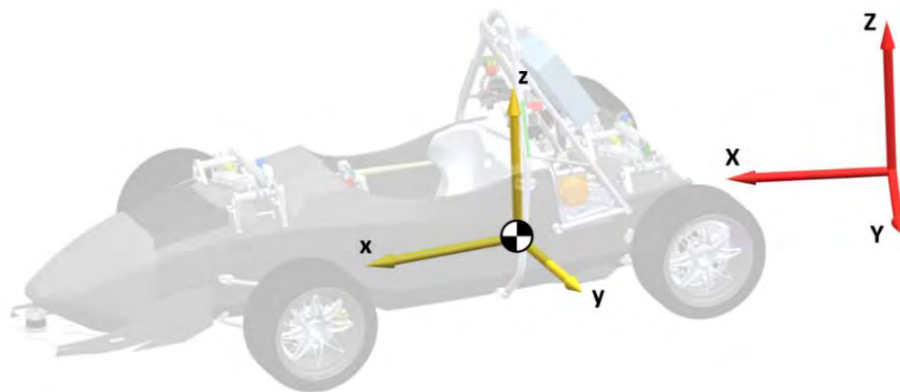


Figure 10: The orientation of the inertial frame (red) and the body frame (yellow)

### 3.1.2 Kinematic Modeling

One of the simplest vehicle models is the kinematic bicycle model, illustrated in Figure 11. The model is constructed solely from geometric relationships and relies on the assumptions that the vehicle moves at a constant speed, no slip occurs at the wheels, and the vehicle only moves within the horizontal  $XY$ -plane of the inertial coordinate frame. It can represent the motion of a slow-moving vehicle relatively well, as long as its underlying assumptions are met. The bicycle model lumps the two front wheels and the two rear wheels into one imaginary front wheel and one imaginary rear wheel, respectively. The two wheels are connected by a rigid link. The front wheel is able to rotate around its midpoint to allow for steering. The advantage of the bicycle model, compared to a four-wheel model, is that it simplifies the equations of motion because there is only one steering angle. At each instant in time, the vehicle rotates around an instantaneous center of rotation, labeled with the letter  $O$ . The equations of motion of the kinematic bicycle model depend on a chosen reference point on the vehicle, which is labeled  $C_{Set}$ . The kinematic equations will change with the position of the selected reference point. [9][10]

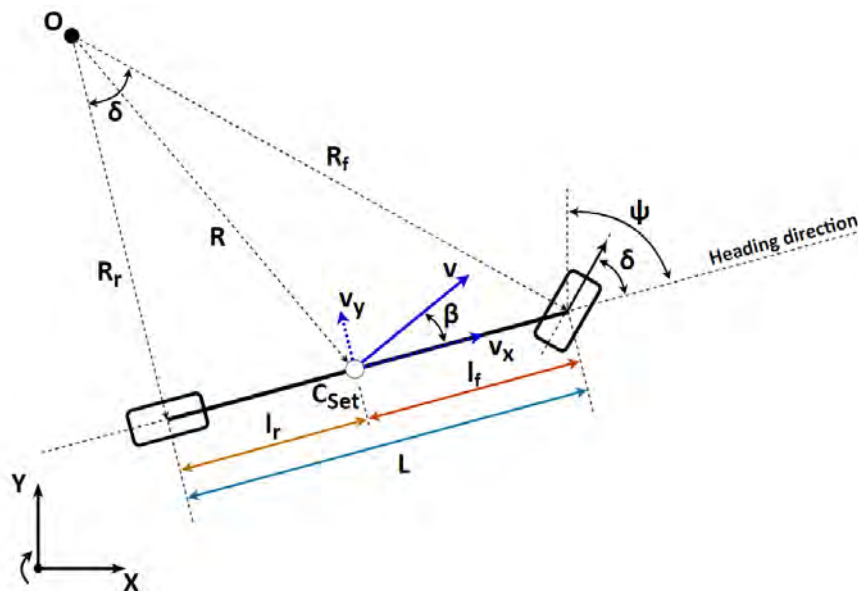


Figure 11: Kinematic Bicycle Model Depiction

To express the equations of motion, the following dimensions and angles are defined. The distances between the reference point  $C_{Set}$  and the front and the rear axles are called  $l_f$  and  $l_r$ , respectively. The sum of  $l_f$  and  $l_r$  is equal to the wheelbase  $L$ . The distance between the points  $O$  and  $C_{Set}$  is called the curve radius  $R$ , and the distances between the point  $O$  and the center of the front and the rear axles are called  $R_f$  and  $R_r$ , respectively. The yaw angle  $\psi$  is the difference between the heading direction of the vehicle and the  $X$ -axis of the inertial

reference frame. It's rate of change is called the yaw rate  $\dot{\psi}$ . The steering angle  $\delta$  is the angle between the heading direction of the vehicle and the heading direction of the front wheel. Finally, the slip angle  $\beta$  is defined as the difference between the velocity vector  $v$ , which points in the actual direction of motion of the vehicle, and the heading direction of the vehicle. Based on these variables and the made assumptions, the motion of the vehicle can be described with the equations (1) to (3). The complete derivation is described in [10]. [2][9][10][12]

$$\dot{X} = v \cos(\psi + \beta) \quad (1)$$

$$\dot{Y} = v \sin(\psi + \beta) \quad (2)$$

$$\dot{\psi} = \frac{v}{R} = \frac{v}{l_f + l_r} \cos \beta \tan \delta, \quad (3)$$

where  $\beta = \tan^{-1}\left(\frac{l_r}{l_f + l_r} \tan \delta\right)$

The steering angle  $\delta$  can be expressed as a function of the wheelbase  $L$  and the length  $R_r$ , as shown in equation (4). Applying small-angle approximation, allows to eliminate the inverse tan function from the equation and to replace the length  $R_r$  with the curve radius  $R$ . The simplified result can be seen in equation (5). [12]

$$\tan(\delta) = \frac{L}{R_r} \Rightarrow \delta = \tan^{-1}\left(\frac{L}{R_r}\right) \quad (4)$$

$$\delta \approx \frac{L}{R} \quad (5)$$

#### 3.1.3 Dynamic Modeling

Dynamic models describe a system using the forces and moments acting on it. They are either derived from first principles, which rely on knowledge of the system, or are constructed from empirical data. A combination of the two approaches is also possible. They can be arbitrarily complex. A high-fidelity model can reflect the dynamics of a system very accurately, however the higher complexity can complicate the design and the computation of the model. More complicated systems can be split into individually modeled subsystems, in order to reduce the complexity of the system and to better represent its dynamics. Typical components of a vehicle that are modeled in subsystems are for instance the tires, aerodynamics, and actuators. To simplify the overall vehicle dynamics, the vehicle model can be split into a longitudinal and a lateral 2D-model. [13]

### Longitudinal Vehicle Dynamics

The longitudinal vehicle dynamic model, illustrated in Figure 12, views the vehicle from the  $xz$ -plane of the body frame. The vehicle travels on an incline with angle  $\xi$ . The model only considers the forces that affect the vehicle's forward and backward motion. The forces that act on the vehicle are the longitudinal front and rear tire forces  $F_{xf}$  and  $F_{xr}$ , the rolling resistance forces  $R_{xf}$  and  $R_{xr}$ , the aerodynamic drag force  $F_{aero}$ , and the  $x$ -component of the gravitational force  $F_{gx}$ . These forces can be grouped by their sign into tractive and resistive forces. By applying Newton's second law of motion, shown in equation (6), the equation for the longitudinal vehicle motion follows in equation (7). The inertial term  $m\ddot{x}$ , on the left side of the equation, consists of the vehicle's total mass and the acceleration of the vehicle in the longitudinal direction. The tractive and resistant forces are obtained from individually modeled subsystems. The design of these subsystems is the actual task in the design process of a dynamic model. [12]

$$m\ddot{x} = \sum F_{tractive} - F_{resistant} \quad (6)$$

$$m\ddot{x} = F_{xf} + F_{xr} - F_{aero} - R_{xf} - R_{xr} - F_{gx} \quad (7)$$

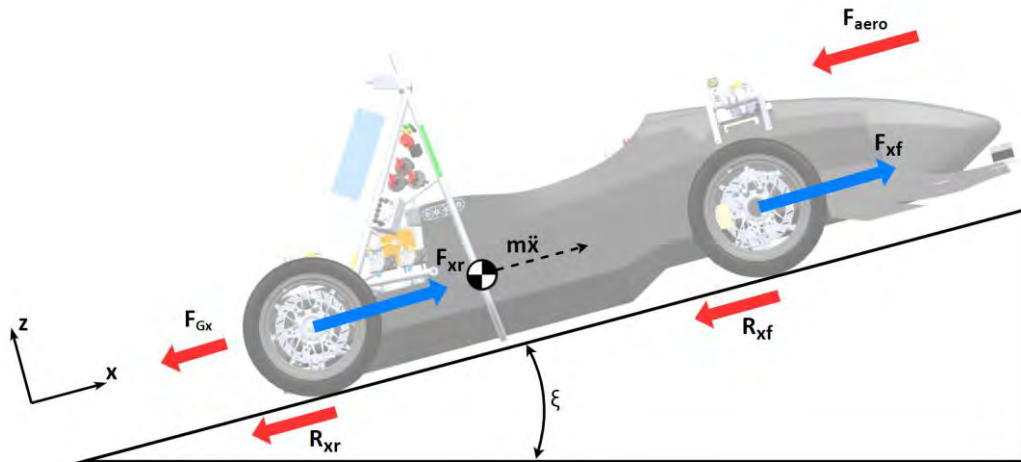


Figure 12: Longitudinal Vehicle Dynamic Model

### Lateral Vehicle Dynamics

Similar to the kinematic model, a bicycle model can be used in the lateral dynamic model, illustrated in Figure 13, to simplify the vehicle's equations of motion. The assumptions regarding speed and motion with the inertial frame made for the kinematic model are kept, while the no-slip assumption is removed. Also, the definition of the variables  $\delta, \beta, l_f, l_r$ , and

### 3 Theoretical Background

$v$  from Figure 11 is kept. The reference point is set at the center of gravity of the vehicle, with the total vehicle mass  $m$ . [12]

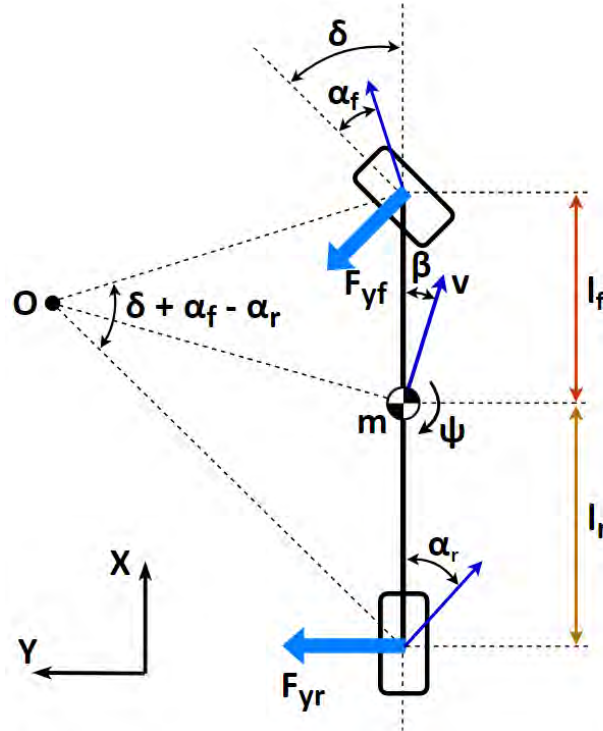


Figure 13: Lateral Vehicle Dynamic Model

The tire slip angles  $\alpha_f$  and  $\alpha_r$  are added to the model, which are defined as the difference between the actual direction of motion of the front and rear wheel, and the heading direction of the front and rear wheel, respectively. They can be expressed with the equations (8) and (9). Further, the lateral tire forces  $F_{yf}$  and  $F_{yr}$  are added to the model. For small slip angles, they can be approximated through their linear relationship with the front and rear wheel slip angles  $\alpha_f$  and  $\alpha_r$ . This linear relationship, described in equations (10) and (11), is characterized by the front and rear tire's lateral cornering stiffness factors  $C_{yf}$  and  $C_{yr}$ . Other influences, like drag and rolling resistance forces are neglected to simplify the model. [2]

$$\alpha_f = \delta + \beta - \frac{l_f}{v} \dot{\psi} \quad (8) \quad \alpha_r = \beta + \frac{l_r}{v} \dot{\psi} \quad (9)$$

$$F_{yf} = C_{yf} \alpha_f \quad (10) \quad F_{yr} = C_{yr} \alpha_r \quad (11)$$

The total lateral acceleration of the vehicle  $\ddot{y}$  in the inertial frame consists of the lateral acceleration  $\ddot{y}_{lat}$  and the centripetal acceleration  $\ddot{y}_{cen}$  from the rotation of the vehicle. The tangential and centripetal acceleration can be expressed in terms of the velocity of the

vehicle, the rate of change of the slip angle, and the yaw rate, seen in equation (12). By applying Newton's second law of motion, the equations (13) and (14) follow to describe the vehicle's lateral motion. Combining equations (12) and (13) results in equation (15). [14]

$$\dot{y} = \dot{y}_{lat} + \dot{y}_{cen} = v\dot{\beta} + v\dot{\psi} \quad (12)$$

$$m\ddot{y} = F_{yf} + F_{yr} \quad (13)$$

$$I_z\ddot{\psi} = l_f F_{yf} - l_r F_{yr} \quad (14)$$

$$mv(\dot{\beta} + \dot{\psi}) = F_{yf} + F_{yr} \quad (15)$$

## 3.2 Path Tracking

For feedback control a reference signal and a feedback signal are necessary. The numerical difference between the two signals is the error signal. There are two types of errors in lateral motion control, the crosstrack error and the heading error. The crosstrack error  $\varepsilon$  is the distance between a chosen reference point on the vehicle and the closest point on the reference path. It expresses how far the vehicle is away from the reference path. The heading error  $\theta$  is defined within the inertial frame as the difference between the angle of the path  $\vartheta$  and the vehicle's yaw angle  $\psi$ , at the vehicle's reference point along the path. It expresses how well the vehicle is aligned with in the direction of the reference path. In order for the vehicle to track the reference path accurately, both the heading error and the crosstrack error must converge to zero over time. [11]



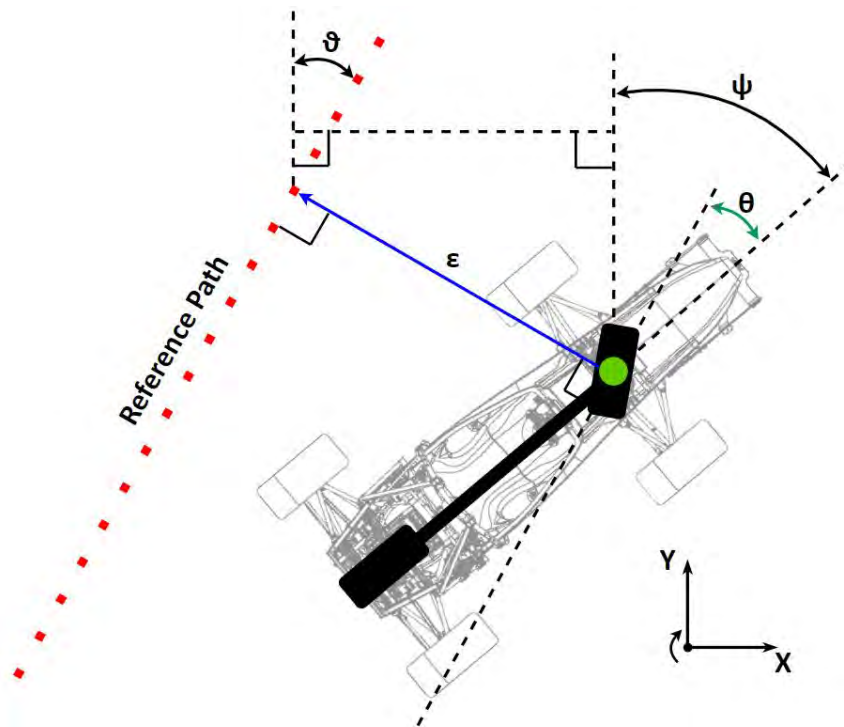


Figure 14: Illustration of the crosstrack error  $\epsilon$  (blue) and the heading error  $\theta$  (green) in the bicycle model with the reference point in the center of the front axle

## 3.3 Control Techniques

This section will present the state of the art of vehicle control by introducing a selection of different control approaches, commonly used in autonomous vehicles.

### 3.3.1 PID Control

One of the most common feedback control methods is proportional–integral–derivative (PID) control. PID control, illustrated in Figure 15. It uses closed-loop feedback to repeatedly calculate an error value  $e(t)$  from a desired reference value and a measured process variable. The error signal is fed into the PID controller, which applies a correction based on a proportional, an integral, and a derivate term. The corrected error signals are summed to the output signal of the PID controller, which is fed to the plant. [23]

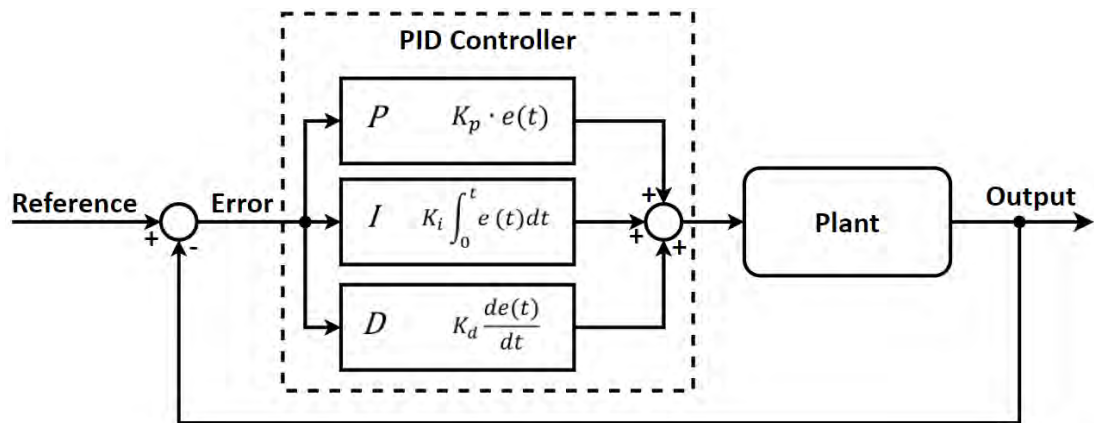


Figure 15: PID Control block diagram [25]

The proportional term of the controller produces an output value that is proportional to the current error value, adjusted by the gain  $K_p$ . A purely proportional controller generally operates with a steady-state error because it requires a non-zero error to produce an output. [25]

The integral term is the sum of the instantaneous error over time multiplied by the gain  $K_i$ . It keeps track of the past inputs by summing the running total of the inputs over time, thereby converging the steady-state error over time to zero. A common problem with the integral term in practice is integrator windup. Integrator windup occurs because every real actuator reaches its physical limitations at some point, leading to a saturation of its output. The integral term is not able to detect this error by itself, resulting in an incorrect integral sum. Thus, in practice an anti-windup method is added to the controller, which turns the integral term off when windup is detected. [25]

The derivative term is calculated by multiplying the rate of change of the error over time with the derivative gain  $K_d$ . The term allows the controller to react proactively to the changing error, which can prevent overshooting the reference value. In practice, a low pass filter is usually added to the input the derivative term, to block unwanted frequencies from propagating to the controller. This reduces unwanted noise in the system, which is unavoidably introduced by the sensors or external disturbances acting on the system. [25]

It is not necessary to use all three terms when designing a PID controller. Depending on the application of the controller, it can be advantageous to set the gains of the integrator or the derivative term to zero. This reduces the complexity of the controller and simplifies the tuning process. Table 1 shows how increasing the independent control terms influences the response of a closed-loop system. Due to these effects, PI-controllers are commonly used in applications which require a minimal steady-state error, such as speed control applications, while PD-controllers are used in applications that require the controller to react quickly to changes or disturbances, like for instance temperature control. [24]

	Rise Time	Overshoot	Setting Time	Steady State Error	Stability
$K_p$	Decrease	Increase	Small change	Decrease	Degrade
$K_i$	Decrease	Increase	Increase	Large decrease	Degrade
$K_d$	Small change	Decrease	Decrease	Small change	Improve if $K_d$ small

Table 1: The effects of increasing the independent PID control gains on the system response [24]

PID control is a control technique used in many different control applications. In vehicle control it can be used for longitudinal and lateral control. In order to use it for lateral control, a kinematic vehicle model can be used to calculate the crosstrack error, which serves as the controlled variable. It should be noted that PID control is only a disturbance rejector and behaves solely reactive. It is not able to predict future behavior, and thus it always takes the controller some time interval to drive the error to zero. [23]

### 3.3.2 Feedforward Control

Feedforward control is an open-loop control method that uses knowledge about the process to provide a predictive system response to reference changes or disturbances. Feedforward controllers are often used in combination with feedback controllers, as depicted in Figure 16, due to their complementary relationship. The control variable adjustment is based on knowledge about the process in the form of a mathematical model or previous measurements. [28]

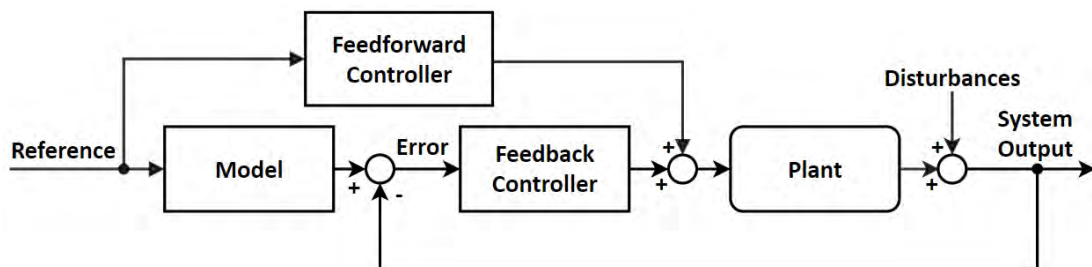


Figure 16: Feedforward control structure block diagram [30]

In the case of the control of autonomous vehicles, the reference changes are usually planned by a trajectory planner, based on the curvature of the path that the vehicle is supposed to follow. The motion planner computes the trajectory at least a few seconds before the motion commands are executed by the actuators. The prior knowledge about the reference changes can be utilized, with the help of a feedforward controller, to respond to reference changes in the moment they occur and reduce the accumulating error. As with any real system, the plant of the vehicle can never be modeled perfectly. Because of that, the feedforward controller

can only reduce the impact of the reference change or disturbance, and not remove it completely. A feedforward controller in combination with a feedback controller can achieve great performance, with the feedforward controller predictively correcting for a large amount of the error, while the feedback controller reactively corrects for the remaining error. [29]

#### 3.3.3 Gain-Scheduling Control

Gain-scheduling, or varying parameter control, is an approach that uses a linear controller with varying parameters to control a nonlinear plant at specific linearized operating points. The parameters vary with respect to the operating conditions, which are controlled by “scheduling variables”, illustrated in Figure 17. The scheduling variables are fed from the plant to a scheduling envelope. The scheduling envelope uses the scheduling variables and a selection mechanism, such as a look-up-table, to select a set of specific control parameters. These control parameters are fed into the controller to update its current control parameters. [26][27]

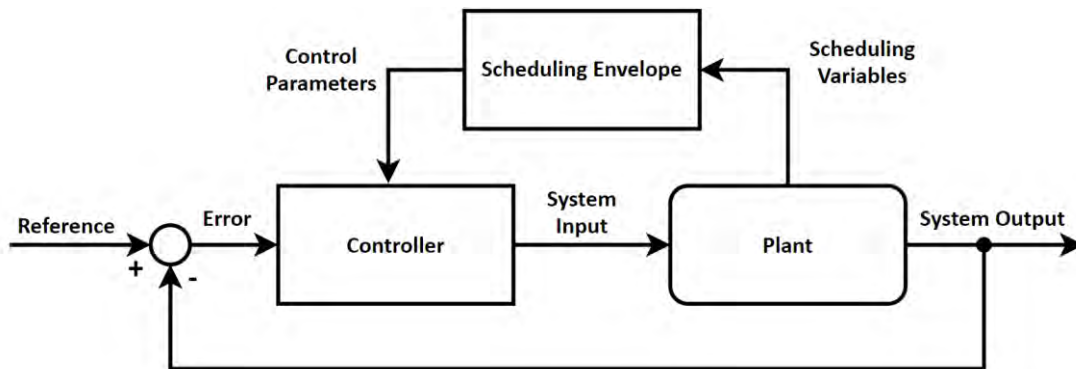


Figure 17: Feedforward control block diagram [26]

It is important to note that the performance and stability is only guaranteed at the selected operating points that the controller is tuned for. Hence, the control performance must be tested and tuned across the entire scheduling envelope to ensure the proper robustness of the controller. Another crucial aspect to consider is the method of transitioning between the sets of parameters. The simplest transitional method is a switch that changes the sets of parameters when a scheduled variable crosses a certain threshold. However, this can lead to an unwanted bumpy behavior. Alternatively, methods such as a transient-switch, a fuzzy-logic, or a function can be used to smooth out the transition between the sets of parameters. In vehicle control, gain-scheduling can be used for both longitudinal and lateral control. For lateral control, the longitudinal speed of the vehicle can be used as the scheduling variable. Depending on the vehicle’s longitudinal speed, the steering controller would thus vary its steering behavior. For longitudinal control, a road gradient can be used as the scheduling

variable, as it is done in [43]. Compared to fixed parameter control, gain-scheduling can lead to an improved performance, especially in highly non-linear systems. [26][43]

#### 3.3.4 Pure Pursuit Control

The Pure Pursuit control method is a geometric path tracking algorithm which relies on the kinematic bicycle model, described in chapter 3.1.2. It was first proposed in 1985 and is one of the earliest path tracking approaches. The controller selects a target point on the reference path, a fixed distance ahead of the vehicle, and computes the steering commands needed to intersect with this point. The target point continues to move forward as the vehicle moves in the direction of the path, while the steering angle is continuously adjusted. The control law of the Pure Pursuit controller is based the calculation of the curvature of a circular arc, shown in Figure 18, with the radius  $R_r$  that connects the center of the vehicle's rear axle and the selected target point on the reference path, ahead of the vehicle. The target point is determined by the look-ahead distance  $l_d$ . The angle between the vehicle's heading direction and the look-ahead direction is called  $\alpha$ . It can be calculated in the inertial frame from the vehicle's current position with the coordinates  $X_{vh}$  and  $Y_{vh}$ , the target point with the coordinates  $X_{tp}$  and  $Y_{tp}$ , and the yaw angle  $\psi$  using the arc tangent function, seen in equation (16). [16]

$$\alpha = \tan^{-1} \left( \frac{Y_{tp} - Y_{vh}}{X_{tp} - X_{vh}} \right) - \psi \quad (16)$$

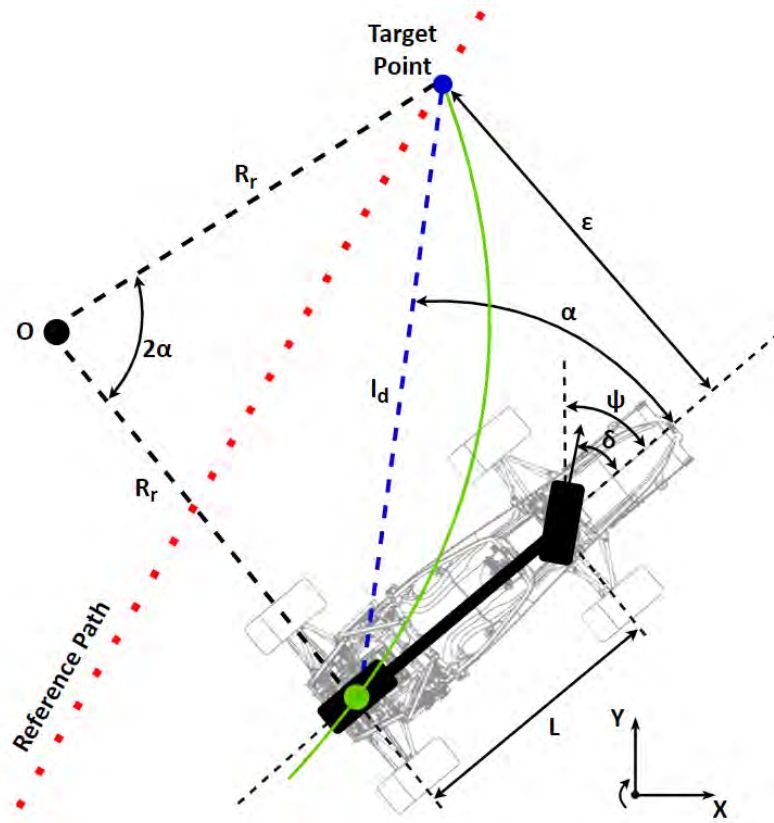


Figure 18: The underlying geometry of the Pure Pursuit control law

The Pure Pursuit control law is derived from the geometry of the triangle formed by the center of the vehicle's rear axle, the target point on the reference path, and the instantaneous center point  $O$ , seen in Figure 18. By applying the law of sines to this triangle, equation (17) follows. The equation can be simplified to equation (18) and further to equation (19), using trigonometric identities. The equation can then be rearranged to equation (20) to express the curvature of the circular path  $\kappa$  created by the pure pursuit controller, which is the inverse of the arc's radius  $R_r$ . With this curvature and the equation (5) from the kinematic bicycle model, the steering angle command can be formulated in equation (21). [15]

$$\frac{l_d}{\sin(2\alpha)} = \frac{R_r}{\sin\left(\frac{\pi}{2} - \alpha\right)} \quad (17)$$

$$\frac{l_d}{2\sin(\alpha)\cos(\alpha)} = \frac{R_r}{\cos(\alpha)} \quad (18)$$

$$\frac{l_d}{\sin(\alpha)} = 2R_r \quad (19)$$

$$\kappa = \frac{1}{R_r} = \frac{2\sin(\alpha)}{l_d} \quad (20)$$

$$\delta(t) = \tan^{-1}(\kappa L) = \tan^{-1}\left(\frac{2L\sin(\alpha)}{l_d}\right) \quad (21)$$

It should be noted that a pure pursuit controller with a fixed look-ahead distance does not consider the vehicle's speed, which can lead to unwanted varying lateral accelerations at different speeds. To address this problem, the lookahead distance can be obtained through a function with the lookahead scaling constant  $K_{ld}$  that is proportional to the vehicle's longitudinal speed  $v_x$  expressed in equation (22). For saturation, this function can also be limited to a minimum and maximum value of  $l_{dmin}$  and  $l_{dmax}$ . The complete equation of the pure pursuit control law is described in equation (23). [15]

$$l_d = K_{ld}v_x \quad (22)$$

$$\delta(t) = \tan^{-1}\left(\frac{2L\sin(\alpha)}{K_{ld}v_x}\right), \text{ where } l_d \in [l_{dmin}, l_{dmax}] \quad (23)$$

To analyze the behavior of the pure pursuit controller, the crosstrack error  $\varepsilon$  can be obtained through its trigonometric relationship with the angle  $\alpha$  in equation (24). By combining equation (24) with equation (20), the following relationship between curvature and crosstrack error can be established in equation (25). This equation reveals that the pure pursuit controller acts like a proportional controller with a gain factor that decreases in a nonlinear manner as the lookahead distance increases. [15]

$$\sin(\alpha) = \frac{\varepsilon}{l_d} \quad (24)$$

$$\kappa = \frac{2}{l_d^2}\varepsilon \quad (25)$$

In [15] the characteristics of the pure pursuit controller are described after testing it on different racetracks and varying its tuning parameters. The results, illustrated in Figure 19, show that a with a smaller lookahead the tracking becomes more accurate and more oscillatory, while with a larger lookahead the tracking becomes less accurate and less oscillatory. Thus, there is a trade-off between tracking performance and stability. The controller is characterized by its asymptotic stability, which converges locally. The experiments also showed a respectable robustness to discontinuities in the path and demonstrated the controller's characteristic of "cutting corner" on curved paths. [15][16]

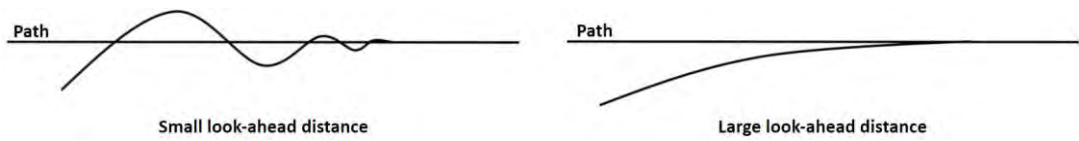


Figure 19: Effect of look-ahead distance for small and large values [15]

The pure pursuit control algorithm is commonly used in autonomous vehicles and robotics because of its simplicity. As it is purely based on kinematics, its performance decreases when the vehicle's motion is affected by tire slip, as is the case in aggressive race driving maneuvers. Nonetheless, for a normal driving behavior and at low speeds a pure pursuit controller can perform quite well. [16]

#### 3.3.5 Stanley Control

The Stanley control method is a geometric path tracking controller that was developed by the Stanford Racing Team, the winning team of the 2005 DARPA Grand Challenge, which is a racing event for autonomous vehicles. The controller was named after their vehicle, "Stanley", and was designed for off-road driving because the event was held in a desert. The control law was designed to achieve a high level of stability, which was crucial to drive in the desert's rough terrain consisting of loose sand and mud puddles. The Stanley controller, illustrated in Figure 20, is based on the kinematic bicycle model, described in chapter 3.1.2. It augments these equations further with additional terms to better capture the motion of the vehicle. The vehicle's reference point is set to the center of the front axle. The Stanley controller is a nonlinear feedback controller that uses the crosstrack and the heading error to control the vehicle. Unlike the pure pursuit controller, it does not use a look-ahead distance. [17]



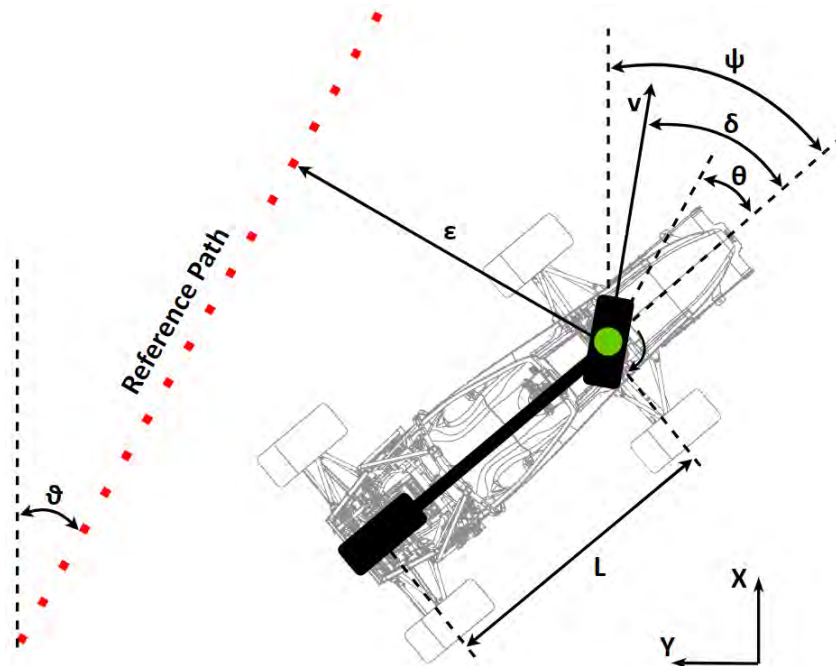


Figure 20: The underlying geometry of the Stanley control law

The Stanley control law consists of several terms. The first term keeps the front wheel aligned with the reference path. This is accomplished by setting the steering angle  $\delta$  equal to the heading error  $\theta$ . A second term then is added to eliminate the crosstrack error. This term uses the cross-track error  $\epsilon$ , which is proportionally adjusted through the gain  $k_{S,p}$  and inversely adjusted through the longitudinal speed of the vehicle  $v_x$ . The adjusted crosstrack error is then mapped by an arc tangent function to the angular range from minus  $\pi/2$  to plus  $\pi/2$ , which saturates the term's influence on the control law. Next, the steering angle command is restricted to the minimum and a maximum steering angles  $\delta_{min}$  and  $\delta_{max}$ , which models the effect that the steering wheel can only be turned a finite amount in each direction. The up to here derived control law is known as the simplified Stanley control law and is shown in equation (26). [15][18]

$$\delta = \theta + \tan^{-1}\left(\frac{k_{S,p}\epsilon}{v_x}\right), \quad \delta(t) \in [\delta_{min}, \delta_{max}] \quad (26)$$

At lower speeds, the simplified Stanley control law behaves quite aggressively to noisy speed measurements. Thus, the controller can further be improved by adding the softening gain  $k_{soft}$  to the denominator inside the arc tangent function. This damps the steering response and improves stability. A similar problem occurs at higher speeds where the controller can act overly aggressive, due to the increased lateral accelerations experienced at higher speeds. To counteract this problem a third term is added to the control law, which damps

the response by adding a derivate term. This term depends on the yaw rate  $\dot{\psi}$  and is scaled by the yaw rate gain  $k_{d,yaw}$ , as seen in equation (27). [18]

$$\delta = \theta + \tan^{-1}\left(\frac{k_{s,p}\varepsilon}{k_{soft} + v_x}\right) + k_{d,yaw}\dot{\psi}, \quad \delta(t) \in [\delta_{min}, \delta_{max}] \quad (27)$$

Finally, a fourth term can be added to the steering command, which mitigates the time delay and overshoot from the steering actuator. This term computes the steering rate  $\dot{\delta}$  using the currently measured steering angle  $\delta_{meas}$  and the next planned steering angle  $\delta_{traj}(i + 1)$  on the trajectory. Both steering angles are discrete values with the index  $i$ . The steering rate is scaled by the steering rate gain  $k_{d,steer}$ . The final Stanley control law is described by equation (28). [18]

$$\begin{aligned} \delta = \theta + \tan^{-1}\left(\frac{k_{s,p}\varepsilon}{k_{soft} + v_x}\right) + k_{d,yaw}\dot{\psi} \\ + k_{d,steer}\left(\delta_{meas}(i) - \delta_{traj}(i + 1)\right), \end{aligned} \quad (28)$$

$$\delta(t) \in [\delta_{min}, \delta_{max}]$$

The Stanley controller is characterized by its asymptotic stability which converges globally, meaning that no matter the initial yaw angle and position, in principle, the controller will always lead the vehicle back on the reference path. Another interesting characteristic of the Stanley controller is that its crosstrack error decay rate is entirely independent of the vehicle's speed. Thus, the vehicle always converges with the path at the same time, no matter the velocity at which it travels. The performance of the simplified Stanley controller, described in equation (26), was tested in [15]. The results show that as  $k_{s,p}$  increases, the tracking accuracy increases and the vehicle becomes less stable. It is also concluded that the controller has difficulty with discontinuity of a path and that it performed relatively well at lower and moderate speeds. [15]

#### 3.3.6 Model Predictive Control

Model Predictive Control (MPC) is a modern feedback control method that utilizes a model of the plant and an optimizer to iteratively predict optimal control actions. MPC is also called Receding Horizon Control because at each time step it solves an optimization problem over a short time horizon, which recedes as time moves forward. While the system executes the control actions, the controller already solves the next optimization problem to find the appropriate control action for the upcoming time interval. MPC control relies on three control parameters, which are crucial to the controller's performance and contribute to the computational complexity of the MPC algorithm. These parameters, illustrated in Figure 21, are the sample time, the prediction horizon, and the control horizon. The sample time defines

### 3 Theoretical Background

the length of a single time step. Each time step is marked on the horizontal axis with the current time step labeled  $k$ . The prediction horizon is the number of predicted future time steps and shows how far the controller predicts into the future. The control horizon defines the number of control time steps available to the optimizer. The higher the control horizon, the better the prediction, however this comes at the cost of the increased number of computations that are performed. [20][21]

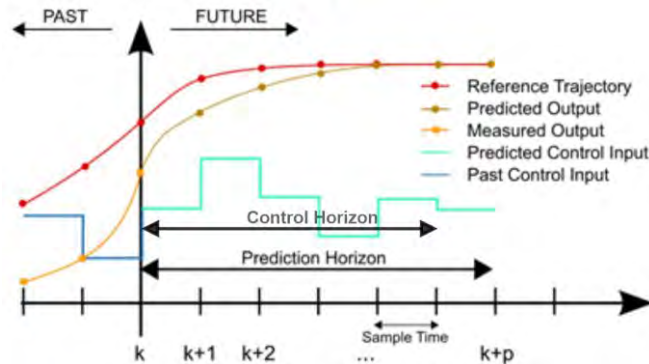


Figure 21: Illustration of the MPC parameters: Sample time, prediction horizon, and control horizon [20]

A key characteristic of MPC is that it can be used for multi-input multi-output (MIMO) systems, meaning it can control multiple processes simultaneously, while considering all the interactions between the system variables. Another feature of MPC is that the states and outputs can be explicitly constrained to stay within secure operating limits. For example, these constraints can account for aspects like undesirable velocities and actuator limitations. Figure 22 illustrates the MPC control structure in a block diagram. The MPC controller consists of a model of the plant and an optimizer. At each time step, the model receives the future input from the optimizer, the past output from the plant, and the past inputs to the plant. With this information, the model outputs several predicted states for the next time-step, which are compared with the reference. The errors are fed into the optimizer, which computes the optimal control inputs for the control horizon. The optimizer uses an objective function and considers the predefined constraints. Finally, the optimal control signal is fed to the plant, while the entire process repeats for the next time step. [19][21]

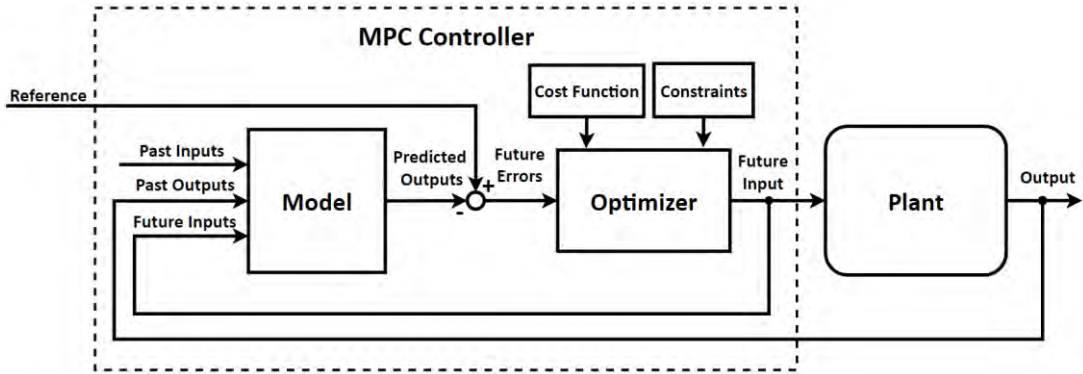


Figure 22: Model Predictive Control block diagram [19]

Model Predictive Control can be used with linear and non-linear models. MPC that uses a linear time-invariant model is called Linear MPC and is usually described with state space equations in discrete form. In this linear state space model, the future states  $\mathbf{x}_{t+1}$  are linearly related to the current states  $\mathbf{x}_t$  and the inputs  $\mathbf{u}_t$ . As shown in equation (29), the states and inputs are transformed by the time-invariant matrices  $A$  and  $B$  respectively, which describe the system. The linear MPC approach pursues to find the optimal set of control inputs  $U$  over a finite control horizon, shown in equation (30), using a linear objective function, shown in equation (31). The objective function that is used for the optimization is a quadratic cost function, which is also called the linear quadratic regular (LQR). The LQR sums the weighted control performance and input aggressiveness over a finite horizon. The square weight matrices  $Q$  and  $R$  have the same number of rows as the number of states in the state vector  $\mathbf{x}_t$ .  $Q$  and  $R$  can be adjusted to achieve a particular optimal objective. The optimization of the LQR cost function provides a closed-form solution, meaning the solution can be represented in the form of a linear function. This function is proven to be asymptotically stable with local convergence and can be solved within a certain amount of time. [22]

$$\mathbf{x}_{t+1} = A\mathbf{x}_t + B\mathbf{u}_t \quad (29)$$

$$U = \{\mathbf{u}_t, \mathbf{u}_{t+1}, \mathbf{u}_{t+2}, \dots\} \quad (30)$$

$$\min_{U = \{\mathbf{u}_t, \mathbf{u}_{t+1}, \mathbf{u}_{t+2}, \dots\}} J(\mathbf{x}(t), U) = \sum_{j=t}^{t+T-1} \mathbf{x}_{j,t}^T Q \mathbf{x}_{j,t} + \mathbf{u}_{j,t}^T R \mathbf{u}_{j,t} \quad (31)$$

Modeling dynamic systems only using linear functions is difficult however, since many systems don't behave linearly. MPC can also use non-linear models. If the MPC model is non-linear, but can be approximated by linear models through linearization, the methods of adaptive MPC and gain-scheduled MPC can be used. The first method, adaptive MPC computes linear models for the relevant operating points, as the operating conditions

### 3 Theoretical Background

---

change. At each time step, the plant model is updated with these linear models, which have the same number of states and constraints across different operating points. The second method, gain-scheduled MPC, works by creating an individual linear MPC controller for each operating point and switching between them according to the operating conditions. This requires an algorithm that changes between the states. It allows the MPC controller to have different states and constraints across the operating conditions. If the non-linear MPC cannot be approximated by linear models through linearization, it is still possible to use non-linear models, as long as the function is differentiable. However, solving non-linear objective functions is only possible through numerical methods because no general closed form solution exists. Solving complex models numerically, can lead to a high computational effort, which can take too long to be solved in real-time. Thus, the MPC model is crucial in real-time applications. [21]

The advances in computing hardware and programming have made more complex MPC models viable, even in real-time applications, like driverless cars. The main advantages of MPC are its ability to achieve excellent results, through its optimization approach, and its property to define explicit constraints. The main disadvantage of MPC is its increased computational requirements, which rely on fast solvers to return optimal solutions within a short amount of time. The task of computing a MPC problem becomes more complex with an increase in the number of system states and constraints, as well as the length of the control horizon and the prediction horizon. MPC shows a lot of promise to be used in autonomous vehicles, especially in autonomous racing, as it can improve performance on a wide operating range, needed for highly aggressive racing maneuvers. [16][22]

# 4 Requirements Analysis

This chapter will define the requirements that the vehicle control system must fulfill. First, it will list the stakeholders interested in the system in section 4.1. Then, it will formulate two system use cases in section 4.2. Finally, it will establish the requirements for the vehicle control system in section 4.3.

## 4.1 Stakeholder Analysis

The development of the vehicle control system is part of the HAWKS Driverless project and intersects with many parts of the project. Several different stakeholders are interested in the progress and the results of this system. Table 2 lists the stakeholders, their position, and analyzes their interest in the vehicle control system.

Position	Stakeholder	Interest
Technical director of the HAWKS Driverless Team	Moritz Höwer	A functioning and robust control system that can be implemented and tested in time and is resource efficient.
Person responsible for state estimation module	Philipp Gehrmann	Information relevant for state estimation (For example: max. lookahead distance)
Person responsible for motion planning module	Currently not assigned	The interface between the motion planning module and the vehicle control module.
Persons responsible for low-level actuator control	Christian Schappmann and Michel Wegner	A control system that provides appropriate actuator commands to the low-level controllers.
System User	The HAWKS Racing Team	A functioning control system that can be tuned and tested quickly.
Event organizer	FSG	A system that is rules compliant and safe.

Table 2: List of stakeholders interested in the vehicle control system

### 4.2 Use Cases

The presented use cases describe the scenarios in which the vehicle control system is used during the FSG competition.

#### 4.2.1 Regular Driving

During regular driving, the vehicle control system must control H14DV autonomously on the Acceleration, Skidpad, and Autocross/Trackdrive racetrack. The vehicle will start at a small distance in front of the start/finish line, marked by the orange cones. At the beginning of the discipline, the vehicle will receive a telemetric start signal from one of the Formula Student officials, which activates the autonomous driving mode and launches the vehicle control system. The vehicle control system must then accelerate and steer the vehicle, based to the data received from the motion planning module. During regular driving, the vehicle shall reach a minimum speed of 12.5 km/h (3.5 m/s) and shall not exceed a maximum speed of 20 km/h (5.6 m/s). The vehicle shall also accelerate no faster than 3 m/s<sup>2</sup>. To simplify motion planning and because the vehicle will only travel at low speeds, the vehicle may simply be given a constant reference speed throughout the complete discipline. Also, the system uses the centerline of the track as reference to control its lateral position. The distance between the outside of the wheels and the cones is roughly 0.8 m on each side. The vehicle must not hit any cones. Once it reaches the stop area, the vehicle control system will be turned off and a controlled stop will be performed. Figure 23 shows the general setup of the regular driving use case. The image is from KA-Racing team from the Karlsruher Institut für Technologie and shows their driverless vehicle being tested on an Autocross/Trackdrive racetrack.



Figure 23: The image shows the driverless vehicle from KA-Racing during testing on an Autocross/Trackdrive racetrack [42]

### 4.2.2 Brake Test Driving

During the autonomous brake test, the vehicle control system must control H14DV on a straight-line track. The vehicle must accelerate from 0 to at least 40 km/h (11.1 m/s) within 20 m. This means the vehicle must accelerate with an average acceleration of  $3.1 \text{ m/s}^2$ . To achieve this, full load will be applied to the H14DV's electric drive. After passing the 20 m marker, the EBS will be manually activated and all electronic components in the vehicle will be turned off, meaning the vehicle control system will be turned off. An image of a FSG brake test is shown in Figure 24.



Figure 24: The driverless vehicle from the TU Delft driverless team performing the autonomous brake test at the FSG in 2019 [45]

## 4.3 System Requirements

The requirements for the vehicle control system are derived from the problem analysis and the described use cases. They are split into functional and non-functional requirements and are listed in Table 3 and Table 4. The functional requirements describe what the system must be capable of doing, while the non-functional requirements describe how the system must perform a certain function.



### 4.3.1 Functional Requirements

Req. ID	Description
R1.01	The system must drive the lateral distance between the vehicle and the reference path to zero over time.
R1.02	The system must drive the difference between the vehicle's longitudinal speed and the reference speed to zero over time.
R1.03	The system must be designed so that the vehicle does not come into contact with any of the cones.
R1.04	The system must keep the vehicle asymptotically stable throughout the vehicle's operating range, which includes speeds of up to 45 km/h (12,5 m/s) and accelerations of up to 4 m/s <sup>2</sup> .
R1.05	The system must be able to control the vehicle on the racetracks of the Acceleration, Skidpad, Autocross, and Trackdrive discipline, where it must reach a minimum speed of 12.5 km/h (3.5 m/s) and may reach a maximum speed of up to 20 km/h (5.6 m/s).
R1.06	The system must be able to control the vehicle during the autonomous brake test, where it must accelerate the vehicle to 40 km/h (11.1 m/s) within a distance of 20 m.
R1.07	The system must be able to handle minor disturbances, such as wind striking the vehicle.

Table 3: Functional Requirements for the vehicle control system

### 4.3.2 Non-Functional Requirements

Req. ID	Description
R2.01	The system must be implemented in software.
R2.02	The system must possess real-time capability, meaning the control algorithms must be computable in less than 10 ms, in order to provide fast enough control commands.
R2.03	The system must be simple in its design and tuning process.
R2.04	The system must not draw more than 15 % of any of the DCU's computational resources (CPU performance, RAM).
R2.05	The system must control the lateral motion of the vehicle so that it does not overshoot the center line of the racetrack by more than 0.8 m.
R2.06	The system must control the lateral motion of the vehicle so that it does not exceed a steady-state accuracy of more than 0.2 m from the center line of the racetrack.
R2.07	The system must damp the lateral motion of the vehicle in a way that no enduring oscillatory steering is perceived.
R2.08	The system must control the longitudinal motion of the vehicle so that it does not overshoot the reference speed by more than 5 km/h (1,4 m/s).
R2.09	The system must control the longitudinal motion of the vehicle so that it does not exceed a steady-state accuracy of more than 1 km/h (0.3 m/s).
R2.10	The system's speed with which it reaches its steady-state values must be balanced, so that neither one of the previous requirements is violated.
R2.10	The system must be designed and tested using a model-based design approach. For this purpose, a simulation and a vehicle model must be used. The vehicle model must represent the motion of H14DV in a way that reasonably balances the tradeoff between accuracy and complexity. This means the level of detail of the model must be limited to a practical extent, in order to avoid unnecessary complication. The physical effects and limitations of the vehicle must be analyzed and considered within the design of the model. This includes for instance factors like time delays, range limitations, and saturation effects.

Table 4: Non-functional Requirements for the vehicle control system

# 5 Concept Development

This chapter aims to develop a concept for the vehicle control system. The different state-of-the-art control techniques, described in chapter 3, are compared and evaluated in section 5.1. Based on this assessment and the defined requirements, a concept for the lateral and for the longitudinal controller will be formulated in section 5.2.

## 5.1 Evaluation and Comparison

In general, more complex control approaches promise better performance of the controller. The control engineers of the Formula Student team AMZ from the ETH Zürich, one of the most experienced and successful teams, confirm this conclusion. Based on their experience, they compared the performance and development time of simple control techniques, such as PID and Pure Pursuit, as well as more complex approaches, like MPC and Reinforcement Learning control (RL), shown graphically in Figure 25. AMZ recommends starting with a simple control technique if no previous experience with more advanced techniques exists. [31]

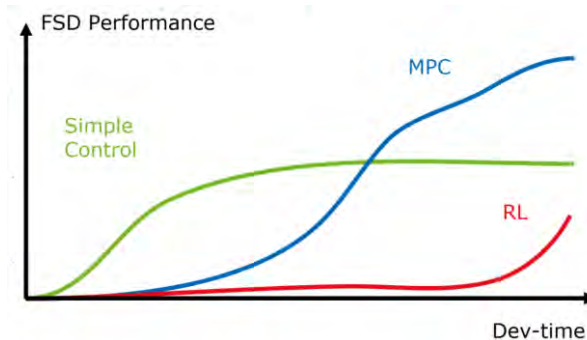


Figure 25: Diagram comparing the performance and the development time of different types of control approaches based on the experience from AMZ. [31]

The comparison also shows that the model, on which the controller is based on, is vital to the performance of the controller. The kinematic bicycle model can represent the motion of a vehicle moving at low speeds and moderate driving conditions quite well. AMZ even reported that at speeds slower than 10.8 km/h (3 m/s) their kinematic model represented the motion of their vehicle better than their dynamic model. Hence, they switch between their models depending on the speed of the vehicle. For higher speeds and in situations in which the no-slip assumption is no longer applicable a dynamic model is needed to accurately reflect the

response of the vehicle. However, the higher fidelity of the model complicates the control problem and is more involved to develop. [32]

In order to evaluate and compare the different techniques for lateral and longitudinal control, the following criteria are established, based on the requirements defined in chapter 4. For longitudinal control the criteria are the operating limitations, the computational requirements, the development complexity in terms of controller design and tuning, the stability, and the potential performance. For lateral control the same criteria apply, as well as, the modelling approach, on which the controller is based on, and the operating limitations.

### 5.1.1 Lateral Control Techniques

The comparison of the lateral control techniques shows that techniques which rely on static control laws and kinematic models, like gain-scheduled PID, Pure Pursuit, and Stanley have lower computational requirements and are more readily implemented, due to their simplicity. For normal driving behavior and moderate speeds, they have a decent path tracking performance. At higher speeds and more aggressive driving behavior, they become less reliable, however. The Pure Pursuit and the Stanley controller are similar in that both rely on the geometry of the kinematic bicycle model. They also both proportionally relate their control gains to the forward speed of the vehicle and use the arc tangent function. Nonetheless, they differ significantly by their individual penalization of the heading and the crosstrack error. Also, the two controllers differ in that the Pure Pursuit controller uses the lookahead distance and the Stanley controller uses additional damping terms to improve its performance. The gain-scheduled PID relies less on geometry, which is why its performance is lower. On the other hand, control techniques which rely on optimal control and dynamic models require significantly more computational resources and are more complex to implement. They achieve better path tracking performance at higher speeds and are able to account for tire slip. Linear MPC has the great advantage of having a guaranteed solvability through a closed-form solution, and thus being asymptotically stable with local convergence. Unfortunately, even the simplest models, like the kinematic bicycle model, use non-linear functions, which limits the practical use of linear MPC. This problem can be overcome through linearization, like applied in [15], however, this compromises on the accuracy of the vehicle model. Table 5 summarizes the evaluation of the techniques used for lateral control.

		Control techniques				
		Gain-scheduled PID	Pure Pursuit	Stanley	Linear MPC	Non-Linear MPC
Criteria	Model	Kinematic	Kinematic	Kinematic	Kinematic or Dynamic	Kinematic or Dynamic
	Operating limitations	Low to moderate speeds	Low to moderate speeds	Low to moderate speeds	No limitations	No limitations
	Computational requirements	Low	Low	Low	Moderate to high	Very high
	Controller design and tuning complexity	Moderate to high	Low	Moderate	High	High
	Stability	Local*	Local*	Global*	Local*	Not guaranteed
	Path Tracking performance	Low	Moderate	Moderate	High	High

**Table 5: The summarized characteristics and results of the evaluation of control techniques for lateral control.**  
**Legend: \*: Asymptotic stability with described convergence.**

### 5.1.2 Longitudinal Control Techniques

The proposed concepts for longitudinal control can be split into two general groups of solutions, PID control and MPC. PID control solutions have low computational requirements and have low to moderate complexity. PID controllers with fixed parameters can have performance issues when the controlled system is not linear and when sudden reference changes occur. These performance issues can be mitigated through gain-scheduling and feedforward control at the cost of increased complexity. As mentioned before, MPC solutions require significantly more computational resources and are more complex to implement. Their performance is generally higher, especially for strongly non-linear systems. Table 6 summarizes the evaluation of the techniques used for longitudinal control.

		Control techniques				
		PID with fixed gains	Gain-Scheduled PID	Feedforward + PID	Linear MPC	Non-Linear MPC
Criteria	Computational requirements	Low	Low	Low	Moderate to high	Very high
	Controller design and tuning complexity	Low to moderate	Moderate	Low to moderate	High	High
	Stability	Local*	Local*	Local*	Local*	Not guaranteed
	Performance	Low to moderate	Moderate	Moderate to high	High	High

**Table 6: The summarized results of the evaluation of control techniques for longitudinal control**  
 Legend: \*: Asymptotic stability with described convergence.

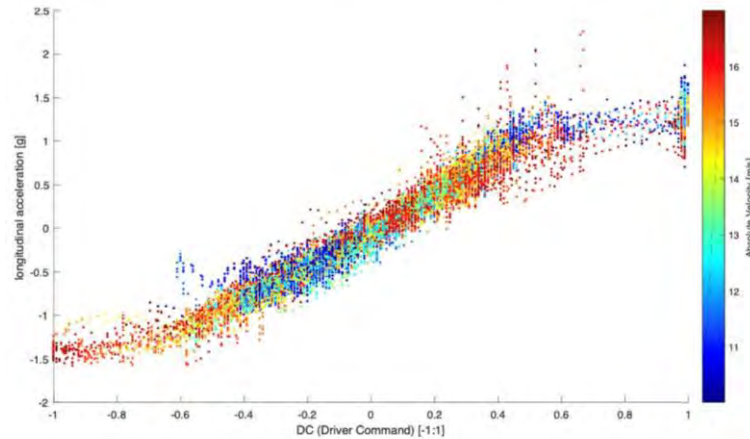
## 5.2 Concept Decision

### 5.2.1 Control Concept

For the concept of the lateral controller the Pure Pursuit control approach is chosen. This approach is chosen because its properties align best with the defined requirements. The vehicle will only travel at relatively low speeds and will only accelerate slowly, thus a controller based on a kinematic model is completely sufficient, if not better than a dynamic model. The controller is expected to perform well, within the required speed range. Its simple control law allows for a fast implementation and tuning process, which is important because the time available for development and testing is limited. Also, Pure Pursuit requires little computational resources and is sufficiently stable, as long as the lookahead distance is chosen properly. Through tuning, its characteristic of cutting-corners can be minimized or can even be taken advantage of for path optimization. In terms of performance, the Stanley Control approach is similarly suited. However, the increased stability, which is not necessarily required, comes at the cost of additional complexity. The gain-scheduled PID approach would require extensive time for tuning and would most likely not achieve a better performance than the Pure Pursuit approach. The main problem with MPC is the difficulty of developing a good model. Linear vehicle models are not accurate enough at low speeds, while non-linear models do not guarantee stability and require a high amount of computational resources.

For longitudinal control the PID approach with fixed parameters is chosen. The concept is chosen because the vehicle is assumed to behave linearly at low accelerations. This assumption is based the data obtained from AMZ, illustrated in Figure 26, which shows that

AMZ's electric vehicle behaves linearly at low accelerations. If in reality the vehicle should behave more non-linearly than expected, the PID controller can be further improved through gain-scheduling. A feedforward controller is not planned to be implemented because the vehicle will only travel at low speeds, which makes this control method ineffective and unnecessary. The PID approach is chosen over MPC because of its lower computational requirements and its lower complexity in terms of tuning and development.



**Figure 26: The longitudinal acceleration in relationship to the propulsion command of the electric Formula Student vehicle from AMZ [31]**

### 5.2.2 Vehicle Model Concept

A dynamic bicycle model approach is chosen to represent the motion of H14DV. Because the model is not utilized for optimization within the controller, like in MPC, its computational requirements are not as important. A dynamic model is capable of describing the system in a lot more detail than a kinematic model, which leads to a better representation of the system. A lot of the information that is needed to dynamically model H14DV has been documented as part of the HAWKS project and can be used within the vehicle model's design. The vehicle model is planned to include three degrees of freedom (3DOF), which allow for translational movement in the  $X$  and  $Y$  direction, and for rotational movement around the  $z$ -axis of the vehicle. The dynamic bicycle model is intended to consider the vehicle's dimensions and inertia, as well as the effects resulting from aerodynamics, tires, and actuators. The suspension of the vehicle is not modeled because of the low speeds and accelerations of the vehicle as well as the nearly flat road surface with no serious road bumps.

# 6 Implementation

This chapter will describe the implementation of the vehicle control system simulation. In section 6.1, it will discuss the simulation environment that is used and the general setup of the simulation. In section 6.2, it will describe the implementation of the trajectory generation. Afterwards, it will explain the design of the vehicle model in section 6.3 and the design of the controllers in section 6.4. Finally, it will show the implementation of the data visualization in section 6.5.

## 6.1 Simulation Environment and Setup

To simulate the vehicle model of H14DV and the vehicle control system, the simulation software Simulink from MathWorks is utilized. Simulink is a graphical MATLAB-based programming environment for modeling, simulating, and analyzing dynamical systems. It is widely used in the field of controls and digital signal processing for model-based design. Simulink is chosen due to the software toolboxes it provides. These toolboxes ease and speed up the implementation process by supplying many pre-built modeling blocks. The simulation uses the Vehicle Dynamics Blockset toolbox and the Automated Driving toolbox.

The simulation setup is split into four sections: Trajectory generation, vehicle control, vehicle modelling, and data visualization. The highest-level of the created simulation is shown in Figure 27 and in A.19. The setup is based on the example [39] from MathWorks. The simulation also inputs data through a MATLAB setup script, which is listed in A.16.

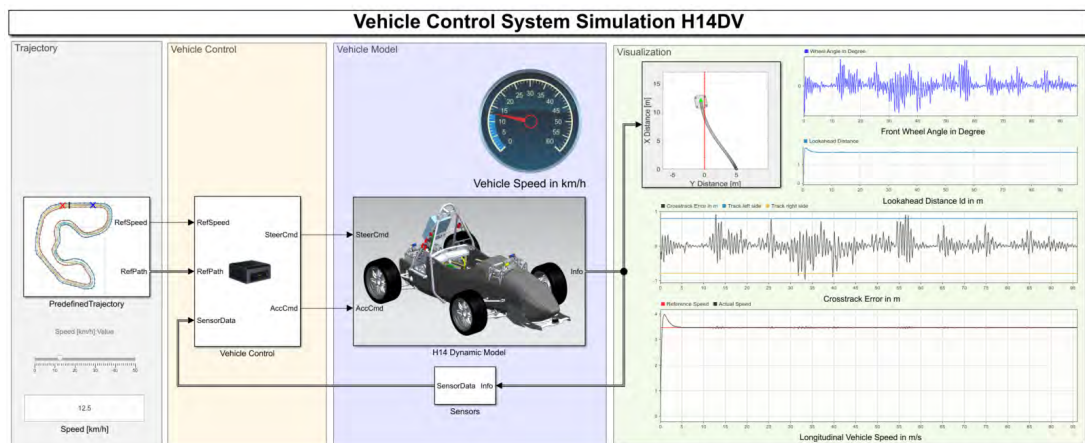


Figure 27: The highest-level of the simulation of H14DV created in Simulink



## 6.2 Trajectory Generation

The trajectory generation block is located on the left side in Figure 27. It provides the reference path and the reference speed to the controllers. The trajectories used in this simulation are not generated by the motion planning module, as shown in the control structure in Figure 7, but are generated manually prior to the simulation. They are created using the Driving-Scenario-Designer application from MathWorks. The application allows the definition of 2D-trajectories using a graphical interface. The trajectories are defined with waypoints. Each waypoint has a  $X$ - and  $Y$ -coordinate as well as a defined vehicle target speed. The data is stored in arrays that can be accessed from Simulink. The created trajectories resemble the racetracks of the FSG disciplines and are shown in Figure 28, Figure 29, and Figure 30. The Acceleration and the Skidpad trajectories are created using the dimensions specified in the FSG competition handbook. The Autocross/Trackdrive racetrack is not published by the FSG. Therefore, the trajectory is constructed from data, obtained from AMZ, of the FSG's 2019 Autocross/Trackdrive racetrack, shown in Figure 31.

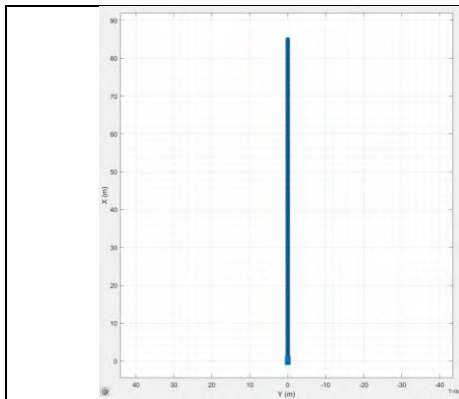


Figure 28: The created Acceleration trajectory

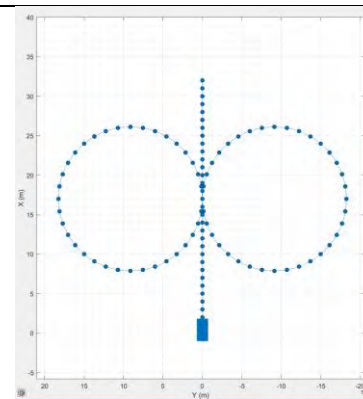


Figure 29: The created Skidpad trajectory

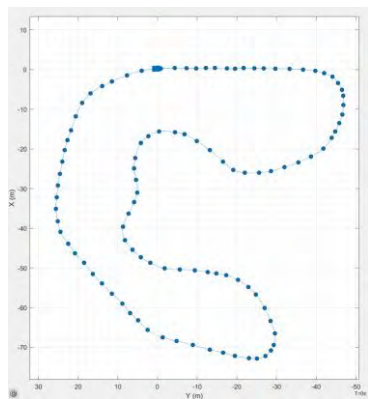


Figure 30: The created Autocross/Trackdrive trajectory

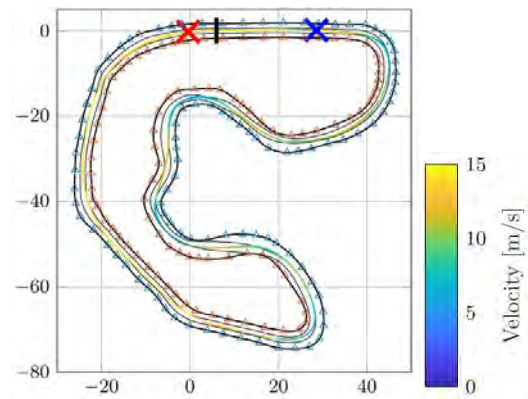


Figure 31: Trajectory data of the FSG's 2019 Autocross/Trackdrive racetrack from AMZ

## 6.3 Vehicle Model Design

The vehicle model is constructed from several individual subsystem models, shown in Figure 32. These subsystems include a vehicle body model, a tire model, a steering model, and a powertrain model. The following sections explain each subsystem model.

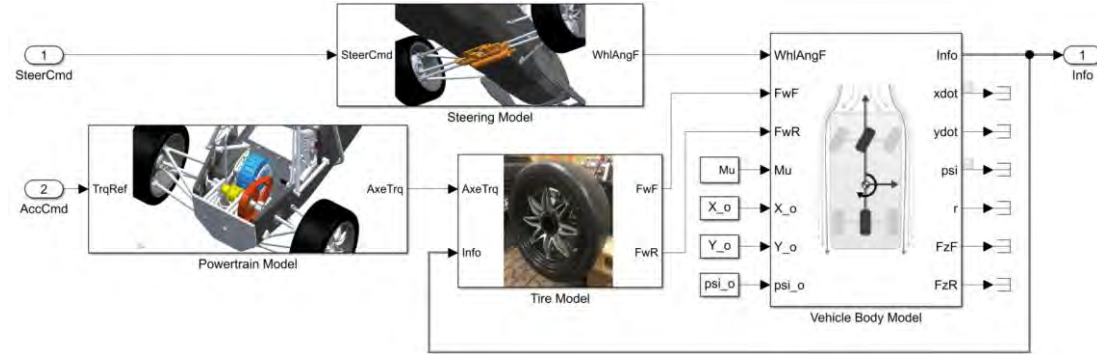


Figure 32: The subsystem models of the vehicle model

### 6.3.1 Vehicle Body Model

The vehicle body model is the main block of the vehicle model. It is implemented by the “Vehicle Body 3DOF Single Track” block from the Vehicle Dynamics Blockset, which models the dynamics of a rigid two-wheel vehicle and computes its longitudinal, lateral, and yaw motion. The dynamic equations of motion of the model are described by equation (32), (33), and (34). They calculate the vehicle’s translational longitudinal and lateral accelerations  $\ddot{x}$  and  $\ddot{y}$  at its center of gravity as well as the vehicle’s rotational acceleration  $\ddot{\psi}$  around its  $z$ -axis. The tangential accelerations are calculated from the radial accelerations, the forces acting on the vehicle, and the vehicle’s mass  $m$ . The radial accelerations  $\dot{y}\dot{\psi}$  and  $-\dot{x}\dot{\psi}$  are the product of the translational and rotation velocities. The forces considered in this model are the front and rear tire forces  $F_{xf}$ ,  $F_{xr}$ ,  $F_{yf}$  and  $F_{yr}$  as well as the total external forces  $F_{xext}$  and  $F_{yext}$  acting on the vehicle. The rotational acceleration  $\ddot{\psi}$  is computed from the moments caused from the longitudinal tire forces  $F_{xf}$  and  $F_{xr}$ , the total external moments  $M_{ext}$  acting on the vehicle, and the moment of inertia of the vehicle around its  $z$ -axis. The horizontal distance from the vehicles COG to the front axle is labeled  $a$  and to the rear axle is labeled  $b$ . The only external forces and moments accounted for in this model are caused by the aerodynamic drag. The complete documentation of the Simulink block is presented in A.11.

$$\ddot{x} = \dot{y}\dot{\psi} + \frac{F_{xf} + F_{xr} + F_{xext}}{m} \quad (32)$$

$$\dot{y} = -\dot{x}\dot{\psi} + \frac{F_{yf} + F_{yr} + F_{yext}}{m} \quad (33)$$

$$\ddot{\psi} = \frac{aF_{xf} - bF_{xr} + M_{zext}}{I_{zz}} \quad (34)$$

As shown in Figure 32, the block is configured to input the steering angle of the front wheel, the longitudinal and lateral tire forces of the front and rear wheel, the friction coefficient, and the initial position of the vehicle. It outputs the signal bus “Info”, which contains the entire accessible data generated by the model. The block accounts for the vehicle’s total mass, dimensions, and moment of inertia around its z-axis with constant parameters. The values of the parameters are based on measurements of the individual components. A list of all the individual measurements is shown in A.10. Some of the parameters can only be estimated, as not all components are fully assembled yet. The aerodynamic drag and lift coefficients  $C_d$  and  $C_l$  are obtained from an air flow simulation using the simulation software ANSYS, shown in Figure 33.

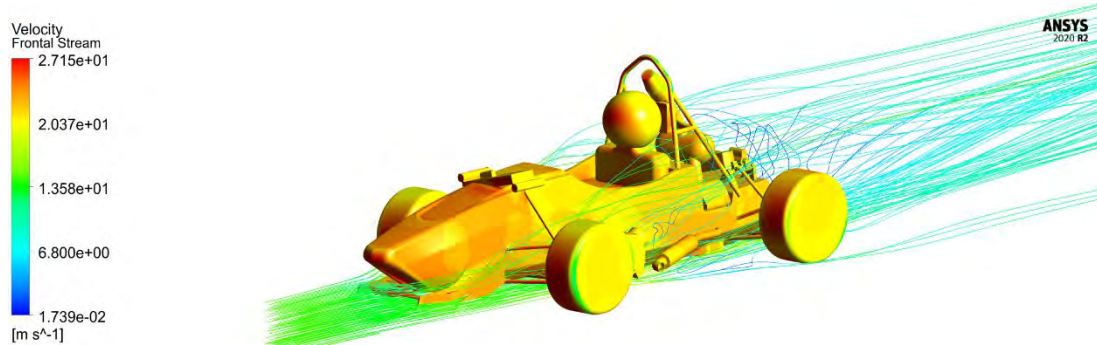


Figure 33: The air flow simulation of H14 with ANSYS

The most important vehicle parameters are summarized in Table 7.

Parameter	Symbol	Value	Unit
Wheelbase	$L$	1550	mm
Track width	$W$	1200	mm
Total mass of the vehicle	$m$	210	kg
Moment of inertia around the z-axis	$I_{zz}$	82.9	kg·m <sup>2</sup>
Air drag coefficient acting along vehicle-fixed x-axis	$C_d$	0.5716	-
Air drag coefficient acting along vehicle-fixed z-axis	$C_l$	0.2365	-

Table 7: Vehicle Parameters

As mentioned before, the vehicle body model considers aerodynamic drag. The aerodynamic drag forces and moments are calculated for the  $x$ -,  $y$ -, and  $z$ -direction. Equation (35) and

(36) show the calculation of the forces and moments for the  $x$ -direction exemplarily. The equations depend on the frontal area of the vehicle  $A_f$ , the respective air drag and lift coefficients  $C_d$  and  $C_l$ , the atmospheric specific gas constant  $R$ , the environmental temperature  $T$ , the absolute pressure  $P_{abs}$ , and the relative air speed  $\bar{w}$ . The frontal area is approximated to be roughly  $1 \text{ m}^2$ .

$$F_{dx} = -\frac{1}{2TR} C_d A_f P_{abs} (\bar{w})^2 \quad (35)$$

$$M_{dx} = -\frac{1}{2TR} C_d A_f P_{abs} (\bar{w})^2 L \quad (36)$$

### 6.3.2 Tire Model

The tires are the interface between the vehicle and the road. Thus, their force generating behavior is critical to the dynamics of the vehicle. Because tire modeling is an intensely researched field in vehicle dynamics, a vast number of different tire models have been proposed in literature. One of the most widely used tire models is the Pacejka tire model. It is an empirical model that provides a combination of accurate force prediction and convenient computation. The Pacejka tire model will be used to model the front and rear longitudinal tire forces  $F_{xf}$  and  $F_{xr}$  in this model. The Pacejka tire model uses the “magic formula”, shown in equation (37) and (38). The formula cannot be derived, thus its name, the “magic formula”. It depends on the respective tire slip ratio  $\lambda$ , the respective tire normal force  $F_z$ , the road-tire friction coefficient  $\mu$ , as well as the four empirical tire parameters  $B_x, C_x, D_x$  and  $E_x$  which depend on the properties of the tire. The slip ratio  $\lambda$  is defined as the difference between the longitudinal speed of the vehicle  $v_x$  and the product of the rotational speed of the tire  $\omega_t$  and the tire radius  $r_t$ , shown respectively in equation (39) and (40), for the front and rear tire. The rotational speed of the tire is determined from torque applied on the tire  $T_t$  and the tire’s moment of inertia  $I_{yy}$ , as shown in equation (41). The tire’s moment of inertia is calculated from its mass  $m_t$  and its radius  $r_t$ , as shown in equation (42).

$$F_{xf} = D_x \sin \left[ C_x \tan^{-1} \left( B_x \lambda_f - E_x (B_x \lambda_f - \tan^{-1}(B_x \lambda_f)) \right) \right] \mu F_{zf} \quad (37)$$

$$F_{xr} = D_x \sin \left[ C_x \tan^{-1} (B_x \lambda_r - E_x (B_x \lambda_r - \tan^{-1}(B_x \lambda_r))) \right] \mu F_{zr} \quad (38)$$

$$\lambda_f = \frac{\omega_{tf} r_t - v_x}{v_x} \quad (39)$$

$$\lambda_r = \frac{\omega_{tr} r_t - v_x}{v_x} \quad (40)$$

$$\omega_t = \frac{1}{I_{yy}} \int_0^t T_t dt \quad (41)$$

$$I_{yy} = m_t r_t^2 \quad (42)$$

The model parameters are obtained by fitting the curve of the magic formula to the empirical data points attained through experiments. For this purpose, the tire data is plotted on a graph, as shown in Figure 34, showing the longitudinal force  $F_x$  of the respective tire over the slip ratio.

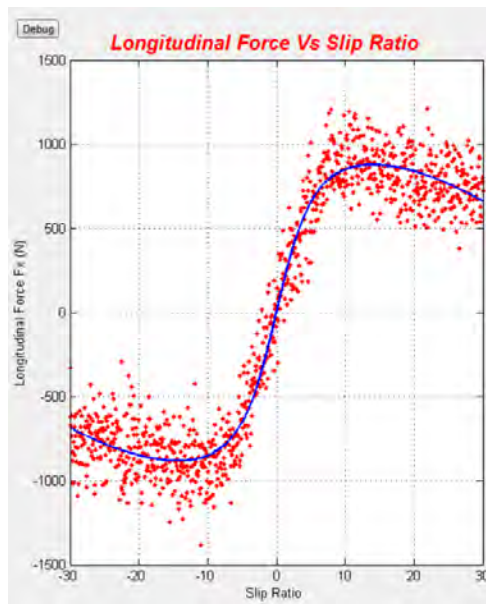


Figure 34: The curve of the magic formula (in blue) is fitted to empirical tire data (in red) by adjusting the tire parameters  $B_x$ ,  $C_x$ ,  $D_x$  and  $E_x$  [46]

To implement the Pacejka tire model for the longitudinal tire forces the “Longitudinal Wheel”-block from the Vehicle Dynamics Blockset is used. The block is able to model the longitudinal behavior of an ideal wheel, the wheel’s rolling resistance and the forces created by a disk or drum brake. However, to keep the model simple, the block is configured to only model the longitudinal tire forces by applying equations (37) to (40). The complete documentation of the “Longitudinal Wheel”-block is found in A.12. Because no tire data for the used Continental 205/470R13-tire is available, the tire data of the slightly larger Continental 205/510R13-tire is used. The data is obtained from the Formula Student Team of the University of Stuttgart, Rennteam Stuttgart, and can be found in A.13. The model parameters are obtained by fitting the curve of the magic formula with MATLAB to the tire data of the Continental 205/510R13-tire, shown Figure 35. The MATLAB code is shown in A.17. The determined parameters are listed in Table 8.

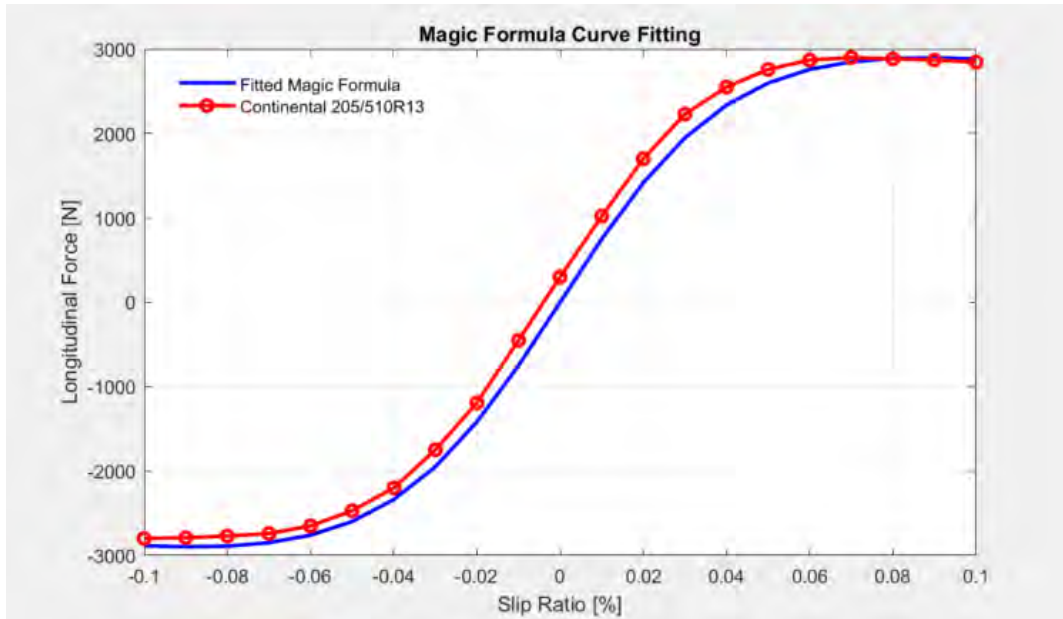


Figure 35: The Magic Formula (blue) is fitted to the longitudinal tire data of the Continental 205/510R13 tire (red)

The lateral tire forces are obtained using a linear tire model, similar to the model introduced in section 3.1.3. A linear tire model is used because only small slip angles are assumed to occur. Also, the model is already integrated into the Vehicle-Body-3DOF-block used in the vehicle body model, which makes its use convenient. The model calculates the front and rear lateral tire forces  $F_{yf}$  and  $F_{yr}$  by applying equation (43) and (44). The forces are proportional to the tire cornering stiffness  $C_y$ , the respective slip angles  $\alpha_f$  and  $\alpha_r$ , the friction coefficient  $\mu$ , and the respective axial load ratio. The axial load ratio is obtained by dividing the respective normal forcing acting of the tire  $F_z$  by the nominal normal force  $F_{znom}$ .

$$F_{yf} = -C_y \alpha_f \mu \frac{F_{zf}}{F_{znom}} \quad (43)$$

$$F_{yr} = -C_y \alpha_r \mu \frac{F_{zr}}{F_{znom}} \quad (44)$$

The tire cornering stiffness  $C_y$  is calculated in equation (45) from the slope of the lateral tire data of the Continental 205/510R13-tire, shown in Figure 36. The original data is also obtained from the Rennteam Stuttgart and can be found in A.14. The MATLAB code is shown in A.18.

$$C_y = \frac{\Delta F_y}{\Delta \alpha_{f,r}} = \frac{2650 \text{ N} - (-2350 \text{ N})}{(-4^\circ - 4^\circ) \cdot \frac{\pi}{180}} = -35.8 \frac{\text{kN}}{\text{rad}} \quad (45)$$

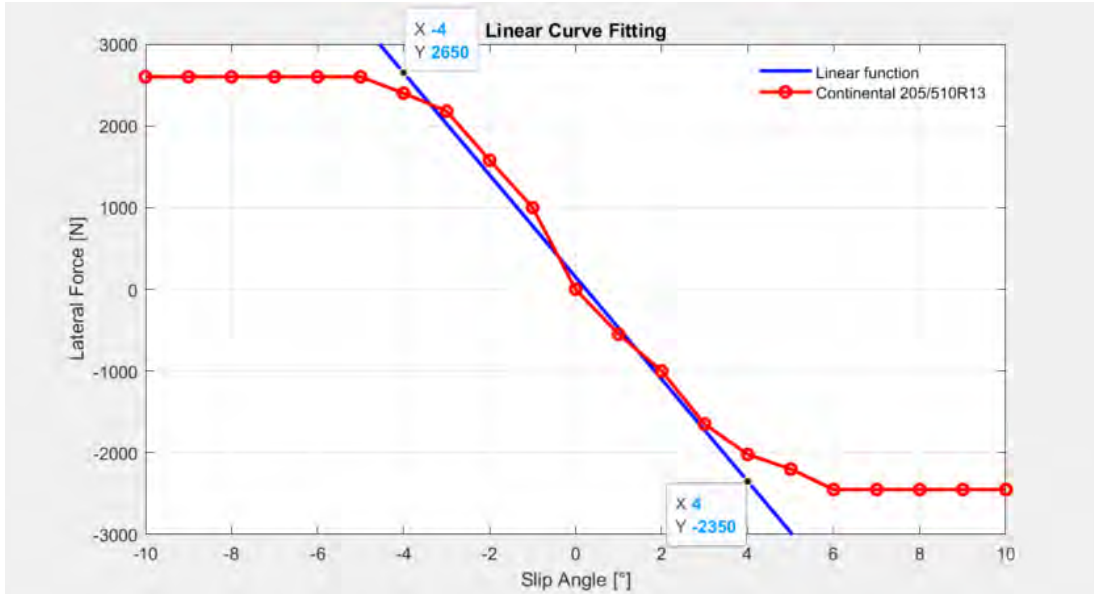


Figure 36: A linear function (blue) is fitted to the lateral tire data of the Continental 205/510R13 tire (red) for small slip angles

The determined longitudinal tire model parameters, the lateral cornering stiffness, the used friction coefficient, and the assumed normal force are listed in Table 8.

Parameter	$B_x$	$C_x$	$D_x$	$E_x$	$I_{yy}$	$C_y$	$\mu$	$F_z$
Value	16	1.65	3.5	0.01	0.4	-35.8	0.72	1150
Unit	-	-	-	-	kg·m <sup>2</sup>	kN/rad	-	N

Table 8: The determined longitudinal and lateral tire model parameters [43]

### 6.3.3 Steering system model

The steering system model, shown in Figure 37, represents the physical behavior of the autonomous steering system. The system is modeled as a first order lag element (PT<sub>1</sub>), with the transfer function, shown in equation (46). Its parameters, the proportional constant  $K_S$  and the lag time constant  $T_S$ , can be determined from tests performed with the real system. However, because the steering system is not fully assembled at this point, the values of the parameters can only be estimated. The estimated values are presented in Table 9. Once, the system is operational,  $K_S$  can be determined from the slope of the system's static characteristic curve, shown in Figure 38. The curve describes the relationship between the

steering command (system input) and the position of the steering rod (system output). The lag time constant  $T_S$  can be determined from the system's dynamic characteristic curve, shown in Figure 39. This curve plots the progress of the steering position over the time.  $T_S$  is determined from the time it takes the system to reach 63 % of its reference value. The output of the steering system is limited to an absolute maximum steering angle of  $\pm 45$  degrees.

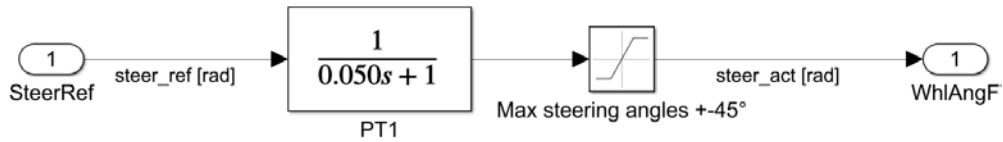
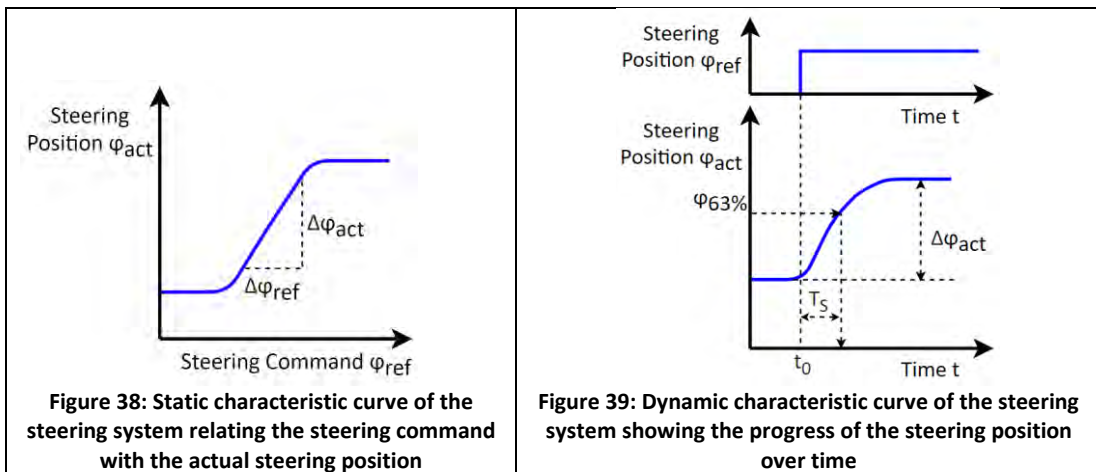


Figure 37: Simulink model of the autonomous steering system

$$G_{steer}(s) = \frac{K_S}{1 + T_S s} \quad (46)$$

Parameter	$K_S$	$T_S$
Value	1	0.05

Table 9: Estimation of the steering system parameters



### 6.3.4 Powertrain model

The powertrain model, shown in Figure 40, represents the physical behavior of the electric drive motor and the chain drive. The drive motor and the low-level controller are modeled, like the steering system, as a  $PT_1$ -element. The parameters can also only be estimated at this point, as the electric motor has not been delivered yet from the manufacturer. The estimated parameter values are presented in Table 10 and are determined similarly as in section 6.3.3.



Parameter	$K_{drive}$	$T_{drive}$
Value	1	0.010

Table 10: Estimation of the drive motor parameters

As mentioned in chapter 2, the electric drive will not only be used to propel the vehicle forward, but also for active braking. For this purpose, the electric motor will be used as a generator. The brake force is generated by the induced currents in the motor magnets. The magnets are induced through the magnetic fields of the electric coils on the rotating motor shaft. The created brake force is also known as back electromotive force (Back EMF). To keep the model simple the brake force is modeled with same the PT<sub>1</sub>-element as used for propulsion. The motor model is limited to a peak current of 330 A, as listed in the data sheet. The chain drive block is modeled as an ideal torque transmitter. It translates the motor torque to the axle torque, by multiplying by a gear transmission ratio of 57:12.

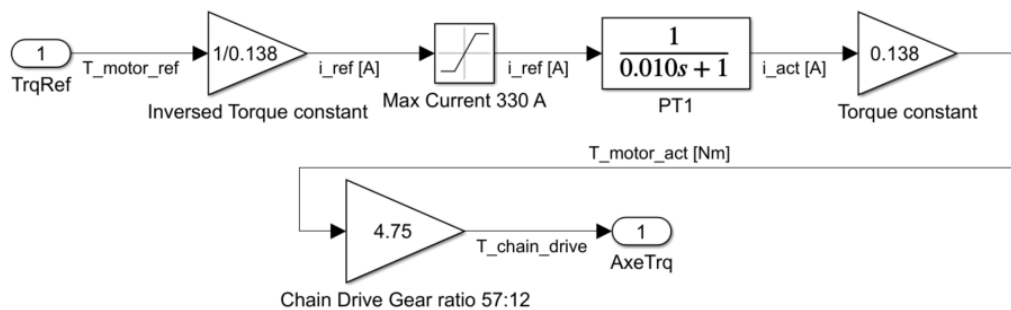


Figure 40: Simulink model of the electric powertrain

## 6.4 Controller Design

This section describes the design of the vehicle control block. The block consists of a data analyzer, the lateral controller, and the longitudinal controller. The two controllers implement the control concepts chosen in chapter 5.

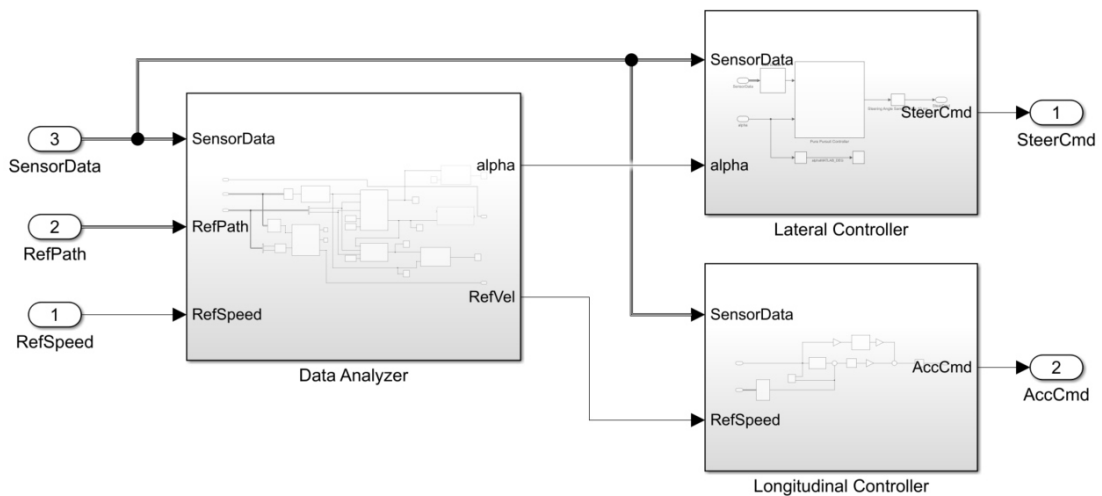


Figure 41: The vehicle control block in Simulink

### 6.4.1 Data Analyzer

The data analyzer block, shown in Figure 42, computes the appropriate reference values for the lateral and the longitudinal controller from the data it receives. It receives the trajectory data from the trajectory generation block and the sensor data from the vehicle model block. The data is sampled at a rate of 10 Hz to simulate the sensor sample time and the run time of the data fusion process. The value is based the results of [6]. The rather slow sampling rate is due to the rotational speed of the LiDAR-Sensor. After sampling the data, the block analyzes the information using an algorithm that is partially written in the MATLAB programming language and partially uses a Simulink block from the Automated Driving toolbox. The upper three function blocks *getRearSetpoint*, *getRefPoint*, *getRefPointVel* select the appropriate target speed from the speed profile depending on the position of the vehicle. This is achieved by finding the closest waypoint of the reference path to the vehicle setpoint, which is located in the center of the rear axle. The function block code can be found in A.15. The reference angle  $\alpha$ , needed for the Pure Pursuit algorithm, is computed using the *getAlpha* block from the Automated Driving toolbox. The block calculates the angle using the position of the vehicle and the reference path. The inner working of the block is not documented by MathWorks.

## 6 Implementation

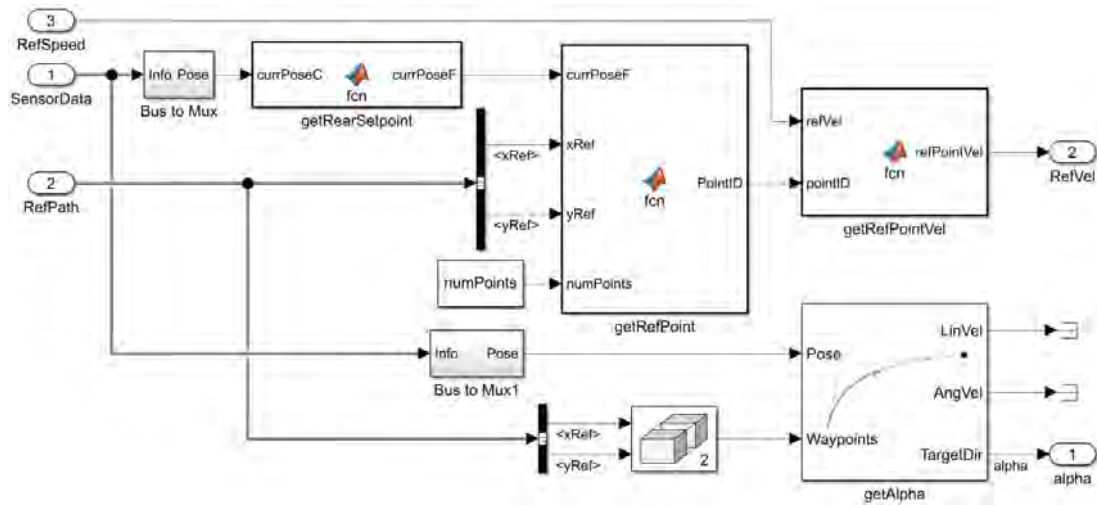


Figure 42: The data analyzer block in Simulink

### 6.4.2 Lateral controller

The lateral controller block, shown in Figure 43, implements the Pure Pursuit controller from equation (23). It inputs the angle  $\alpha$  and the longitudinal speed of the vehicle  $v_x$ . The lookahead distance is computed using the longitudinal speed and the lookahead scaling factor  $K_{ld}$ . It is limited to a minimum and a maximum value, which are determined during testing. The control command is computed with the angle  $\alpha$  and the lookahead distance. The controller outputs the steering angle commands at a rate of 100 Hz.

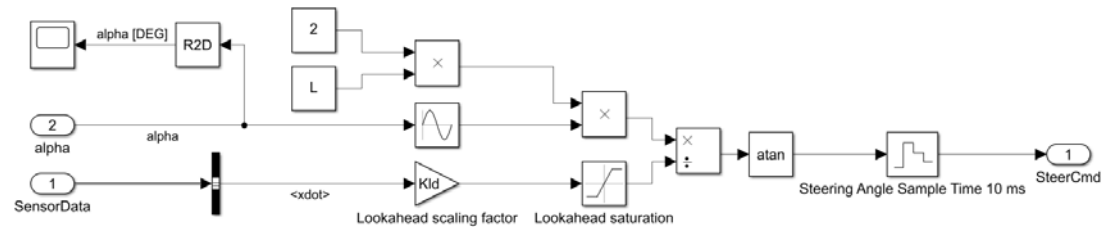


Figure 43: The lateral controller block in Simulink

### 6.4.3 Longitudinal controller

The longitudinal controller block, shown in Figure 44, implements the PI-Controller with fixed parameters. The model inputs the reference speed and the actual speed to calculate the error. To implement the PI-Controller the “PID Controller”-block from Simulink is used. The controller’s output is sampled at a rate of 100 Hz.

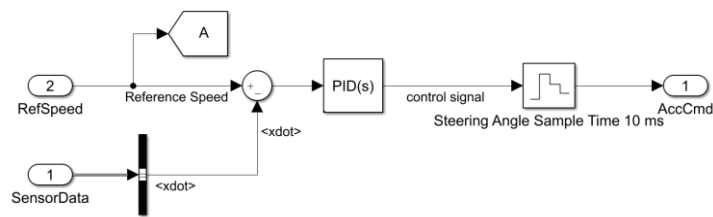


Figure 44: The longitudinal controller block in Simulink

## 6.5 Data Visualization

The data visualization section is located on the right side of Figure 27 and is shown in Figure 45. It provides several visualizations of the data that is generate during the simulation. It contains a 2D-visualization that depicts the vehicle from a birds-eye view and plots the reference path as well as the actual path that the vehicle traveled. Further, it includes separate graphs to visualize the crosstrack error, the vehicle’s front wheel angle, the value of the lookahead distance, and the vehicle’s longitudinal speed. The blue and yellow line in the crosstrack error visualization show the position of the cones, which mark the side of the racetracks.

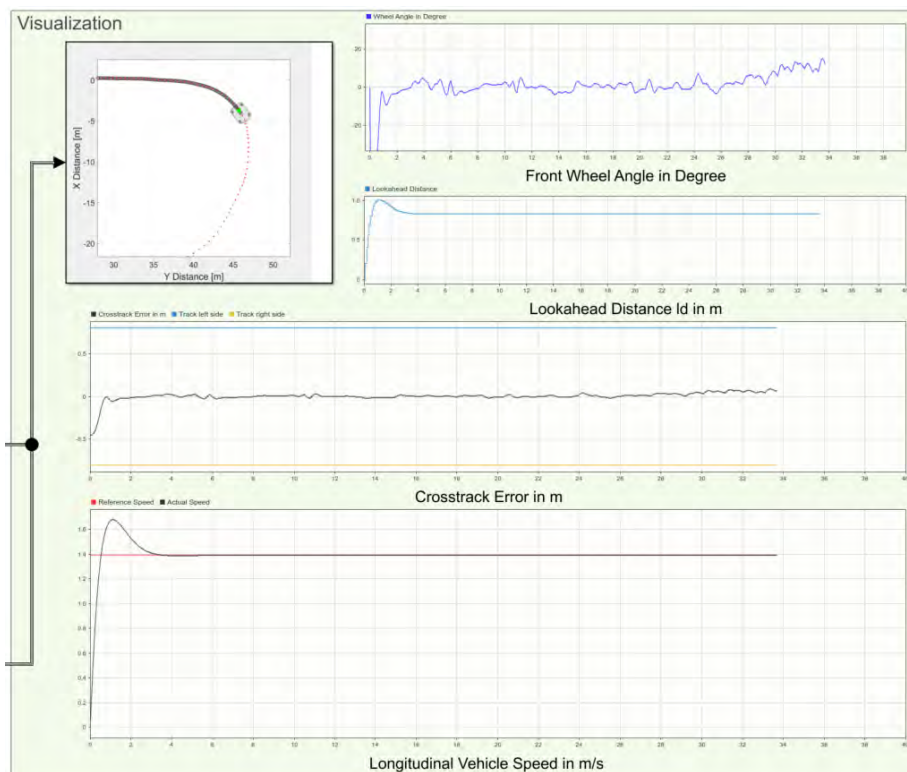


Figure 45: The Data Visualization Section

# 7 Testing and Results

This chapter will describe how the vehicle control system was tested and will present the test results. The vehicle control system was tested using the simulation setup described in chapter 6. Tests were performed on the Acceleration, Skidpad, and Autocross/Trackdrive racetracks. The test results will be described in section 7.1 for the longitudinal controller and in section 7.2 for the lateral controller.

## 7.1 Longitudinal Controller Testing

The longitudinal controller was first tested on the Acceleration racetrack. The steering model was turned off in order to simulate the vehicle's longitudinal motion only. To tune the controller the Ziegler–Nichols tuning method for PID control was first attempted. However, the resulting controller gains did not produce a desired system response. Thus, the controller was simply tuned by trial and error. Only the proportional and integral term of the PID controller were used for tuning because the derivative term caused a high amount of instability in the system. Therefore, in practice instead of PID, a PI controller was designed. The controller gains selected were  $K_p = 1.5$  and  $K_i = 1.5$ . Simulations were performed at constant longitudinal reference speeds of 12.5, 15.0, and 17.5 km/h. The simulation results are shown in Figure 46, Figure 47, and Figure 48.

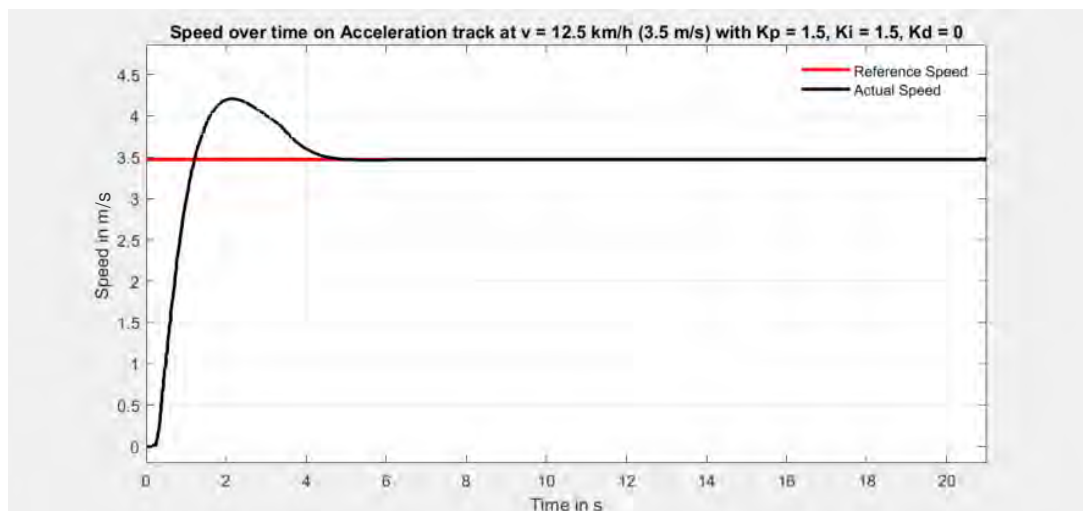


Figure 46: The vehicle's reference and actual speed are plotted over time on the Acceleration racetrack at  $v_{ref} = 12.5$  km/h (3.5 m/s), with  $K_p = 1.5$ ,  $K_i = 1.5$ , lap time = 21 s

## 7 Testing and Results

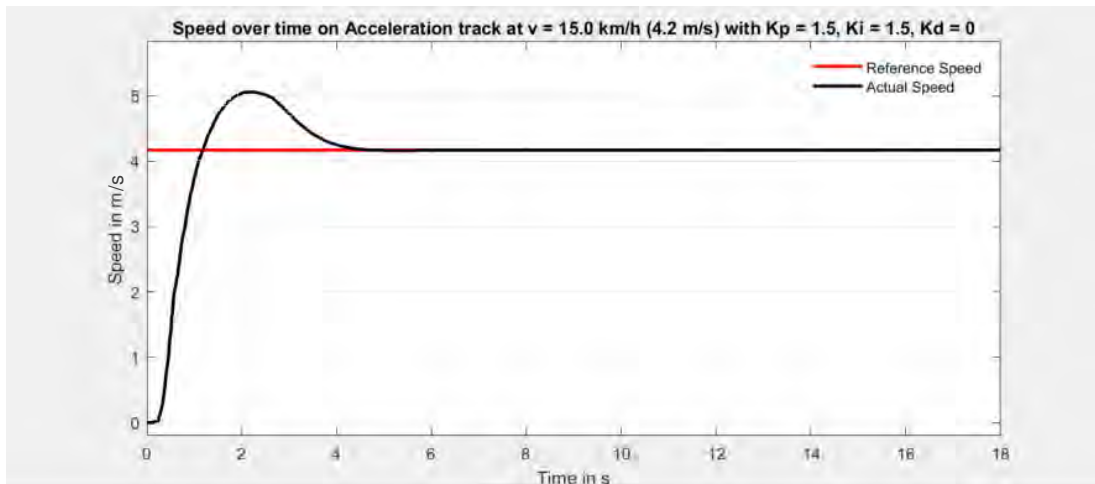


Figure 47: The vehicle's reference and actual speed are plotted over time on the Acceleration racetrack at  $v_{ref} = 15.0$  km/h (4.2 m/s), with  $K_p = 1.5$ ,  $K_i = 1.5$ , lap time = 18 s

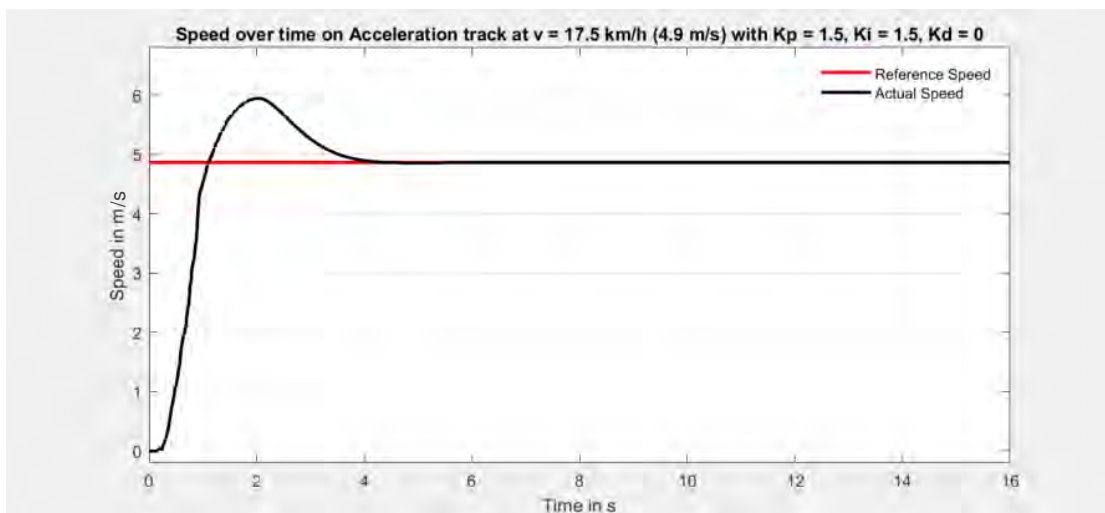


Figure 48: The vehicle's reference and actual speed are plotted over time on the Acceleration racetrack at  $v_{ref} = 17.5$  km/h (4.9 m/s), with  $K_p = 1.5$ ,  $K_i = 1.5$ , lap time = 16 s

The longitudinal controller was then tested on the Skidpad racetrack, shown in Figure 49. The test was performed at a reference speed of 15.0 km/h and with a value of 0.5 for the lookahead scaling constant  $K_{ld}$ .

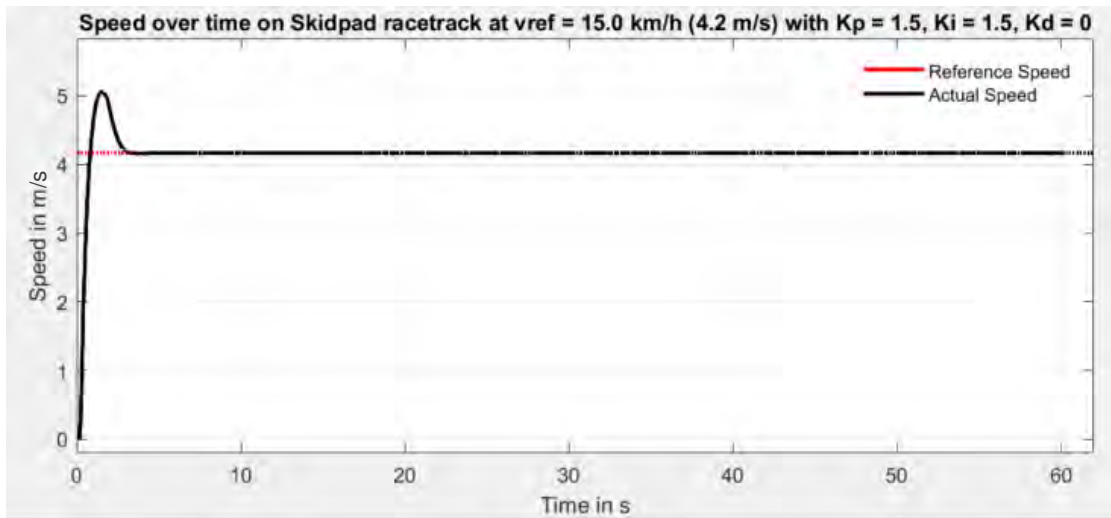


Figure 49: The vehicle's reference and actual speed are plotted over time on the Skidpad racetrack at  $v_{ref} = 15.0$  km/h (4.2 m/s), with  $K_p = 1.5$ ,  $K_i = 1.5$ , lap time = 62 s

Finally, the controller was tested on the Autocross/Trackdrive racetrack, shown in Figure 50. The horizontal axis displays the progress in time in s and the vertical axis displays the speed in m/s. The test was performed using a speed profile, which consistently adjusted the speed depending on the curvature of the path. The reference speed varied between 4.3 km/h (1.2 m/s) and 16.2 km/h (4.5 m/s). The value for  $K_{ld}$  was chosen to be 0.6.

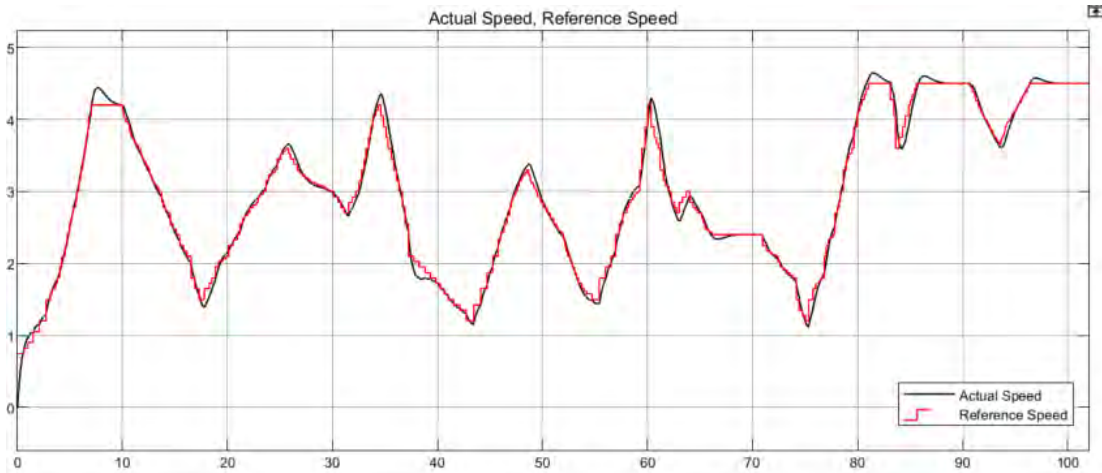


Figure 50: The vehicle's reference and actual speed are plotted over time on the Autocross/Trackdrive racetrack at varying reference speeds, with  $K_p = 1.5$ ,  $K_i = 1.5$ , lap time = 102 s

Also, a simulation of the autonomous brake test was performed, shown in Figure 51. For the brake test different controller gains had to be chosen due to the higher accelerations of the vehicle. The gains selected were  $K_p = 0.5$  and  $K_i = 0.1$ . The vehicle reached the required

## 7 Testing and Results

minimum speed of 40 km/h after a distance of 14.5 m and within a time of 2.1 s. The average acceleration of the simulated vehicle during the brake test was calculated in equation (47).

	$\bar{a} = \frac{\Delta \text{Speed}}{\Delta \text{Time}} = \frac{12.5 \text{ m/s}}{2.1 \text{ s}} = 5.3 \text{ m/s}^2$	(47)
--	---	------

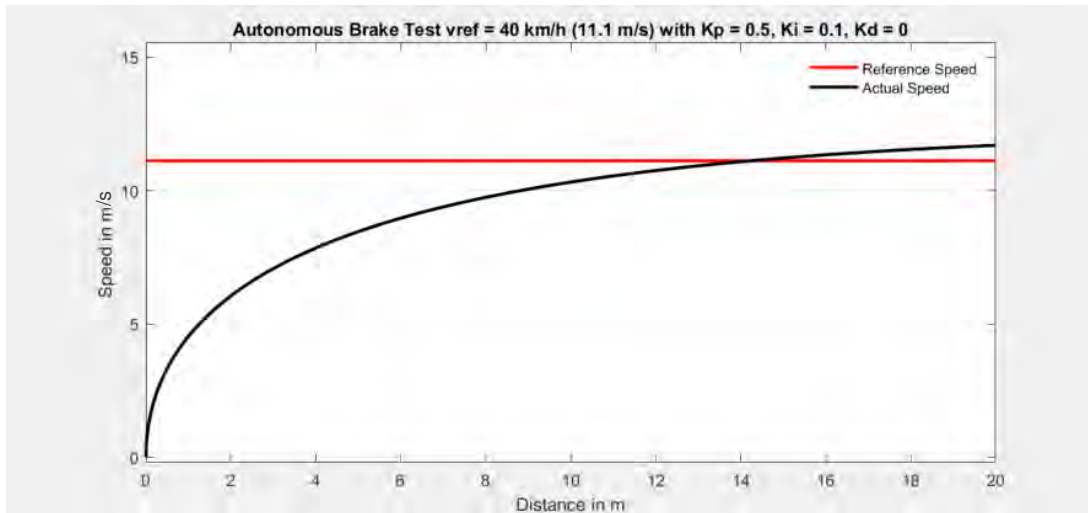


Figure 51: Simulation of the autonomous brake test at  $v_{ref} = 40.0 \text{ km/h}$  (11.1 m/s), with  $K_p = 0.5$ ,  $K_i = 0.1$

## 7.2 Lateral Controller Testing

The lateral controller was first tested on the Autocross/Trackdrive racetrack. Simulations were performed at constant longitudinal speeds of 12.5, 15.0, and 17.5 km/h. At each speed, different values for the lookahead scaling constant  $K_{ld}$  were simulated. The simulation results are shown in Figure 52, Figure 53, and Figure 54. The figures plot the crosstrack error over the racetrack distance. The straight blue- and magenta-colored line mark the left and right side of the track. They are both 0.8 m away from the center to simulate the remaining distance between the wheel and the cones. The recorded data is depicted in different colors to differentiate between the different values of  $K_{ld}$ .



## 7 Testing and Results

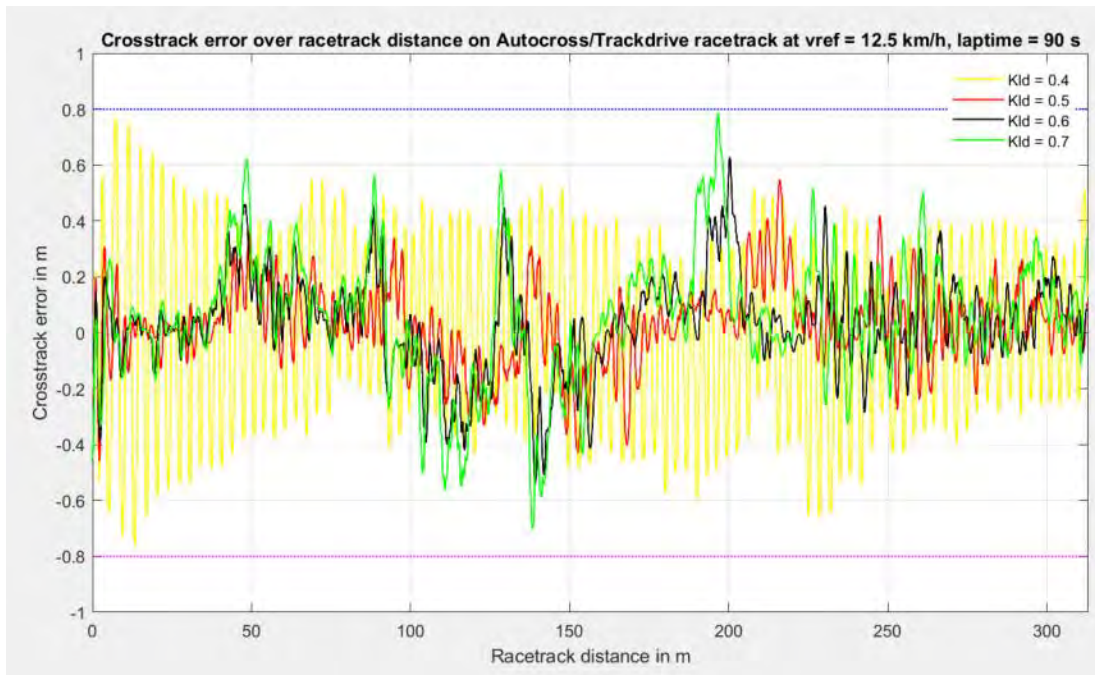


Figure 52: The crosstrack error is plotted over the distance of the Autocross/Trackdrive track at  $v = 12.5$  km/h (3.5 m/s), lap time = 90 s

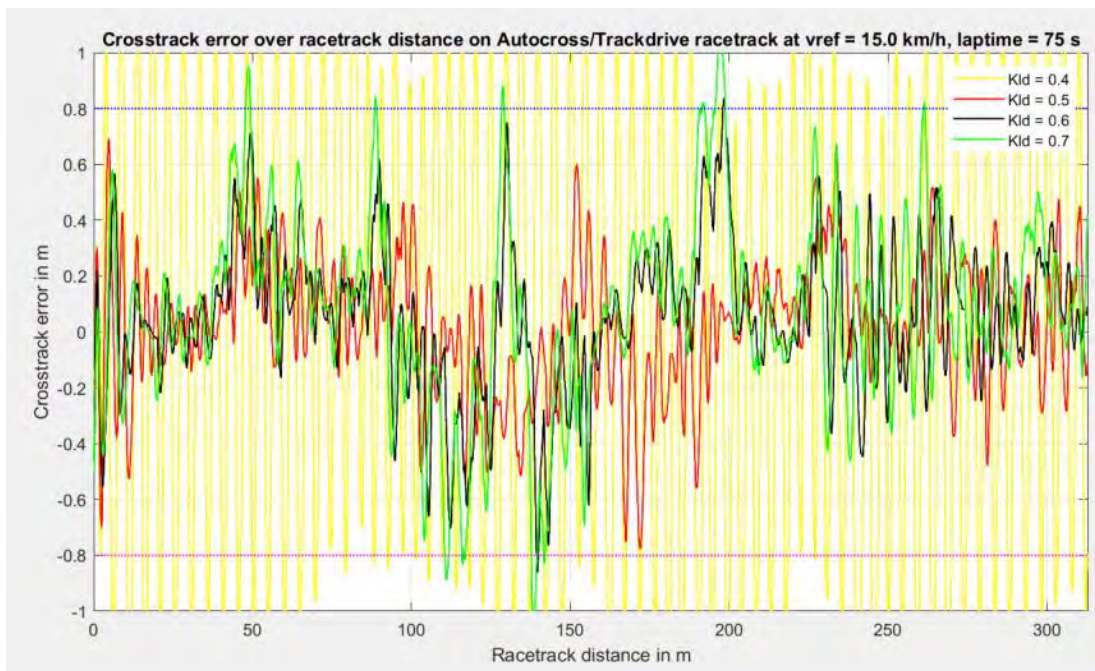


Figure 53: The crosstrack error is plotted over the distance of the Autocross/Trackdrive track at  $v = 15.0$  km/h (4.2 m/s), lap time = 75 s

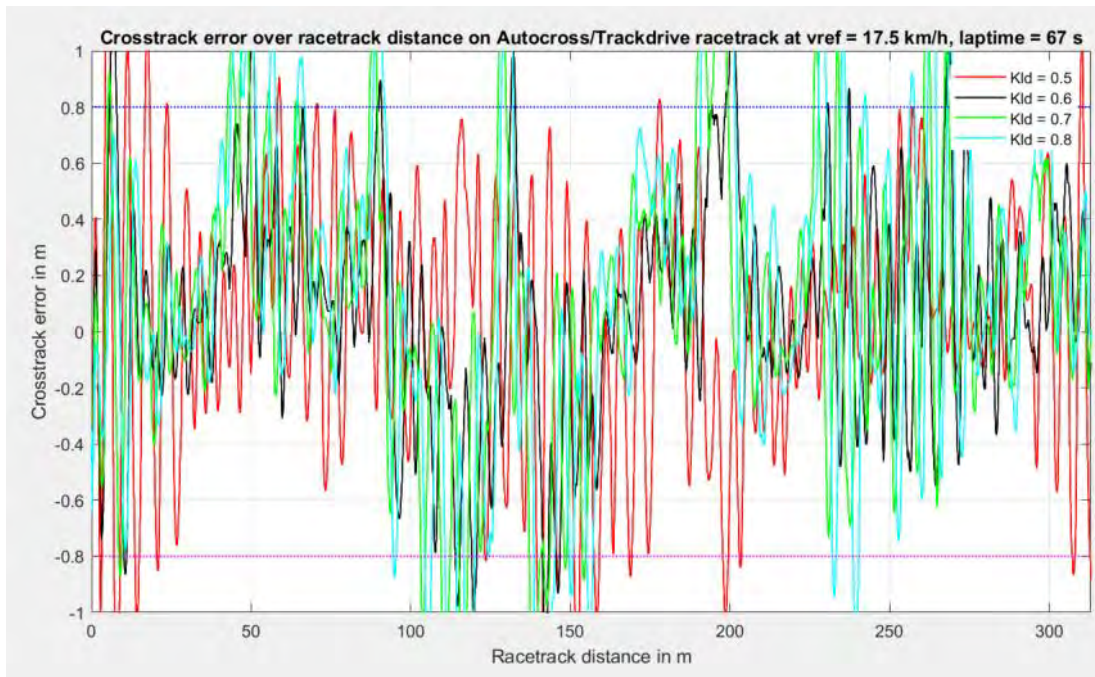


Figure 54: The crosstrack error is plotted over the distance of the Autocross/Trackdrive track at  $v = 17.5 \text{ km/h}$  ( $4.9 \text{ m/s}$ ), lap time = 67 s

The lateral controller was then tested on the Autocross/Trackdrive racetrack with the same varying reference speeds and value for  $K_{ld}$ , as in the simulation in Figure 50. The results are shown in Figure 55.

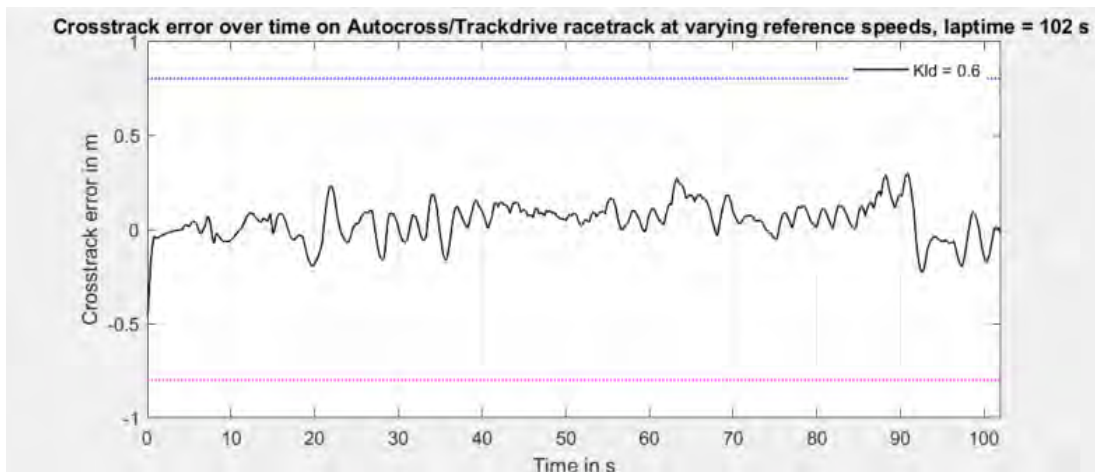


Figure 55: The crosstrack error is plotted over the distance of the Autocross/Trackdrive track at varying speeds with  $K_{ld} = 0.6$ , lap time = 102 s

## 7 Testing and Results

Afterwards, the lateral controller was tested on the Skidpad racetrack, shown in Figure 56, and the Acceleration racetrack, shown in Figure 57. The tests were performed at a speed of 15.0 km/h and with a value for  $K_{ld}$  of 0.5. In the Acceleration simulation, the vehicle started with an offset of 0.1 m to the center of the track to introduce an initial error into the system.

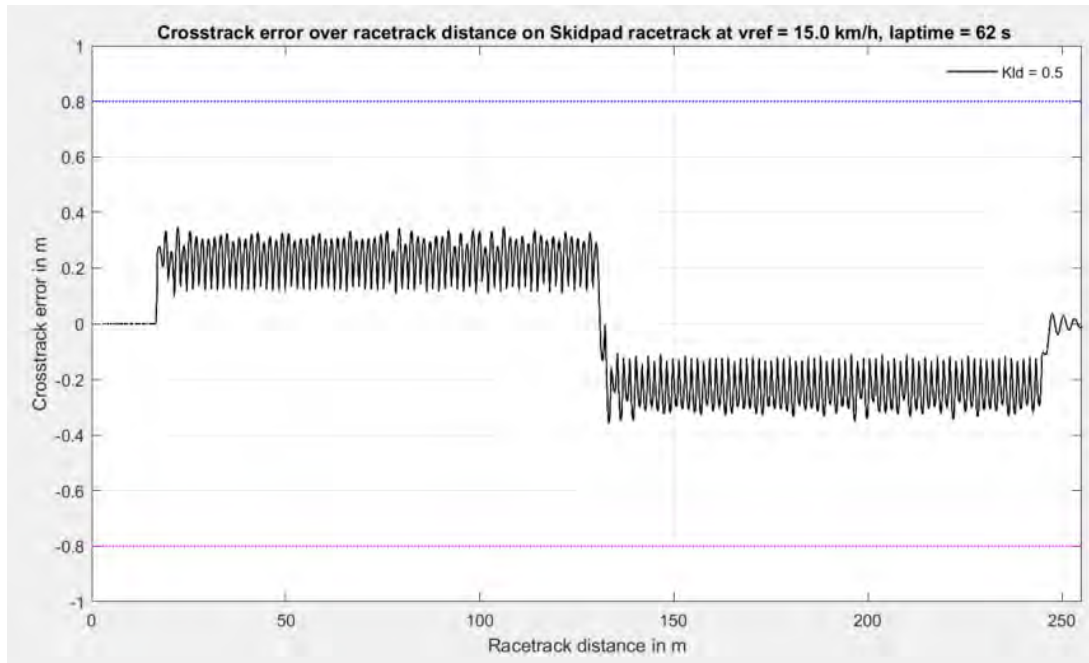


Figure 56: The crosstrack error is plotted over the distance of the Skidpad racetrack at  $v = 15$  km/h with  $K_{ld} = 0.5$ , lap time = 62 s

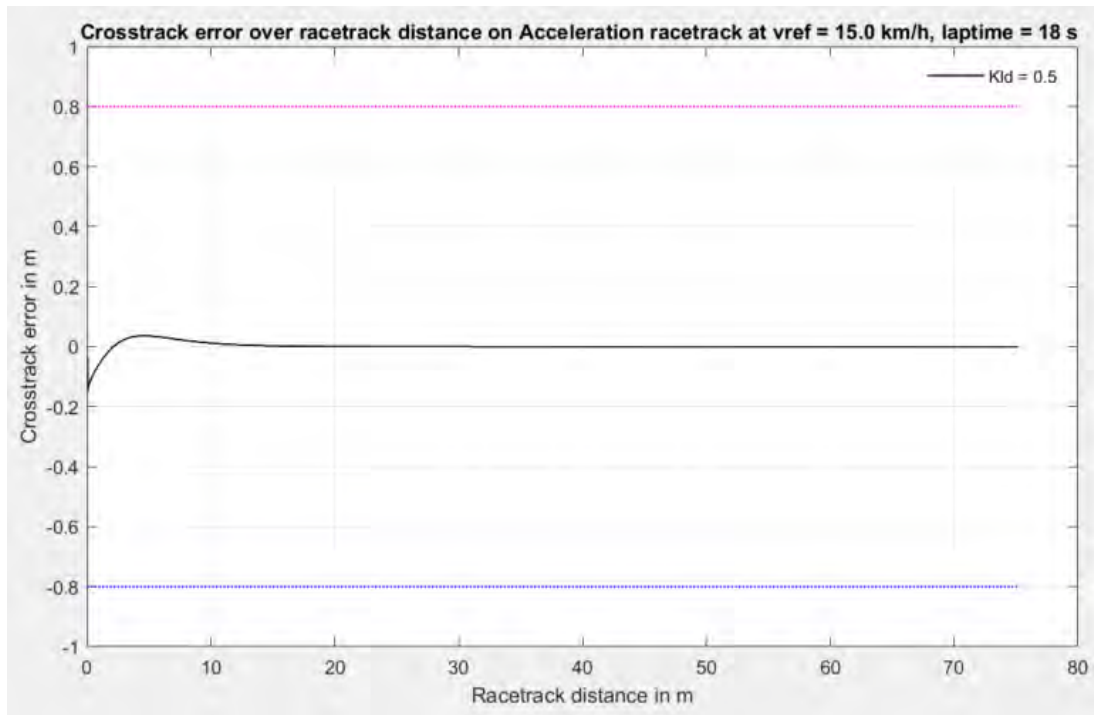


Figure 57: The crosstrack error is plotted over the distance of the Acceleration track at  $v = 15 \text{ km/h}$  with  $K_{ld} = 0.5$ , lap time = 62 s

Finally, the lateral performance of the brake test, simulated earlier in Figure 51, is shown in Figure 58. The value of is set to  $K_p = 1.1$  to increase the controller's stability.

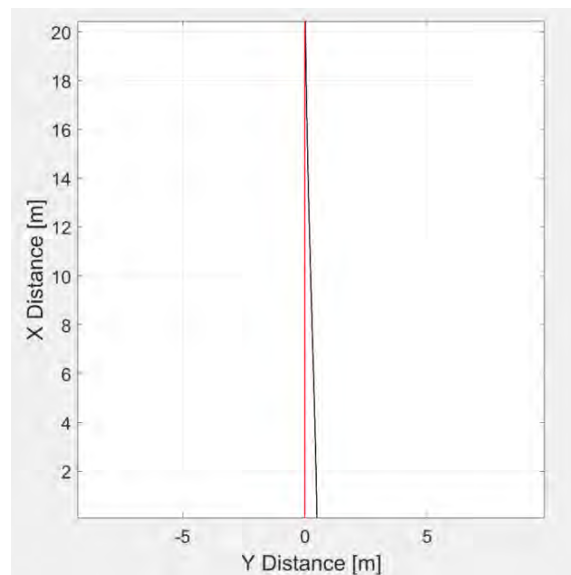


Figure 58: Simulation of the autonomous brake test at  $v_{ref} = 40.0 \text{ km/h}$  (11.1 m/s), with  $K_{ld} = 1.1$

# 8 Evaluation

This chapter will evaluate and discuss the results presented in chapter 7. It will compare the results to the requirements from chapter 4, and determine if the vehicle control system fulfills the set requirements.

## 8.1 Interpretation of the Test Results

The simulation results of the longitudinal controller show that the PI-controller with fixed parameters is able to control the vehicle within a speed range of 0 to 20 km/h (5,6 m/s) on the FSG racetracks. In this speed range the vehicle remains stable and achieves a consistent system response. The vehicle was tuned to respond fast to reference changes and overshoots the reference value by roughly 3.6 km/h (1 m/s). It damps the overshoot quickly and reaches the reference value with high steady state accuracy in less than 3.5 s. However, it must be mentioned that only aerodynamic disturbances were simulated. Thus, the steady state accuracy is assumed to be greater in reality. At speeds above 20 km/h the vehicle becomes less stable. This is due to the non-linear effects of the vehicle appearing at higher accelerations, which the linear controller cannot properly compensate. The gains of the PI-controller must be adjusted in order to achieve a better control performance. For the autonomous brake test, where the vehicle must reach a speed of 40 km/h (11.1 m/s), the control gains were adjusted to the higher acceleration. The controller remained stable and reached the required speed within 14.5 m.

The simulation results of the lateral controller confirm the characteristics of the Pure Pursuit controller described in chapter 3. The controller performs significantly better at lower speeds. A smaller lookahead distance results in more accurate path tracking, however, it also increases instability due to the higher number of oscillations in the steering system. A larger lookahead distance on the other hand results in less accurate path tracking, while being more stable. The test results shown in Figure 53 indicate that a value of 0.5 for the lookahead scaling constant  $K_{ld}$  provides the best system performance at a speed of 15.0 km/h (4.2 m/s) on the Autocross/Trackdrive racetrack. Therefore, this value was also tested on the Skidpad and the Acceleration racetrack at a speed of 15.0 km/h and achieved satisfying results. The results shown in Figure 54 demonstrate that the controller becomes too instable and inaccurate at constant speeds of roughly 17.5 km/h (4.9 m/s) or above on the Autocross/Trackdrive racetrack. Also, from Figure 55 can be concluded that, unsurprisingly,

the crosstrack error can be significantly reduced by actively adjusting the speed of the vehicle in proportion to the curvature of the path.

## 8.2 Review of the Requirements

Req. ID	Description
R1.01	<p>The system must drive the lateral distance between the vehicle and the reference path to zero over time.</p> <p>Result: <b>[Achieved]</b> Successfully demonstrated in simulation for speeds of up to 15 km/h (Figure 53).</p>
R1.02	<p>The system must drive the difference between the vehicle's longitudinal speed and the reference speed to zero over time.</p> <p>Result: <b>[Achieved]</b> Successfully demonstrated in simulation for speeds of up to 20 km/h (Figure 50).</p>
R1.03	<p>The system must be designed so that the vehicle does not come into contact with any of the cones.</p> <p>Result: <b>[Achieved]</b> Successfully demonstrated in simulation for speeds of up to 15 km/h (Figure 53).</p>
R1.04	<p>The system must keep the vehicle asymptotically stable throughout the vehicle's operating range, which includes speeds of up to 45 km/h (12,5 m/s) and accelerations of up to 4 m/s<sup>2</sup>.</p> <p>Result: <b>[Partially Achieved]</b> The controller keeps the vehicle laterally asymptotically stable on all FSG racetracks up to a speed of 15 km/h (4.2 m/s). Additionally, it keeps the vehicle stable on the straight-line track during the autonomous brake test. (Figure 46 to 58).</p>
R1.05	<p>The system must be able to control the vehicle on the racetracks of the Acceleration, Skidpad, Autocross, and Trackdrive discipline, where it must reach a minimum speed of 12.5 km/h (3.5 m/s) and may reach a maximum speed of up to 20 km/h (5.6 m/s).</p> <p>Result: <b>[Achieved]</b> Successfully demonstrated in simulation for speeds of up to 15 km/h (Figure 57).</p>
R1.06	<p>The system must be able to control the vehicle during the autonomous brake test, where it must accelerate the vehicle to 40 km/h (11.1 m/s) within a distance of 20 m.</p>

## 8 Evaluation

	<p>Result: <b>[Achieved]</b> Successfully demonstrated in simulation (Figure 51 and Figure 57).</p>
R1.07	<p>The system must be able to handle minor disturbances, such as wind striking the vehicle.</p> <p>Result: <b>[Achieved]</b></p>
R2.01	<p>The system must be implemented in software.</p> <p>Result: <b>[Achieved]</b> The system was implemented in MATLAB and Simulink.</p>
R2.02	<p>The system must possess real-time capability, meaning the control algorithms must be computable in less than 10 ms, in order to provide fast enough control commands.</p> <p>Result: <b>[Not testable yet]</b> The real-time capability can only be tested on the real system.</p>
R2.03	<p>The system must be simple in its design and tuning process.</p> <p>Result: <b>[Achieved]</b> The applied control are simple in their design and the entire system can be tuned with only three tuning parameters.</p>
R2.04	<p>The system must not draw more than 15 % of any of the DCU's computational resources (CPU performance, RAM).</p> <p>Result: <b>[Achieved]</b> The amount of computational resources needed for the vehicle control system are neglectable due to its simple design.</p>
R2.05	<p>The system must control the lateral motion of the vehicle so that it does not overshoot the center line of the racetrack by more than 0.8 m.</p> <p>Result: <b>[Achieved]</b> Successfully demonstrated in simulation for speeds of up to 15 km/h (Figure 53).</p>
R2.06	<p>The system must control the lateral motion of the vehicle so that it does not exceed a steady-state accuracy of more than 0.2 m from the center line of the racetrack.</p> <p>Result: <b>[Achieved]</b> Successfully demonstrated in simulation for speeds of up to 15 km/h (Figure 53).</p>
R2.07	<p>The system must damp the lateral motion of the vehicle in a way that no enduring oscillatory steering is perceived.</p> <p>Result: <b>[Achieved]</b> Successfully demonstrated in simulation for speeds of up to 15 km/h (Figure 53).</p>

## 8 Evaluation

---

R2.08	<p>The system must control the longitudinal motion of the vehicle so that it does not overshoot the reference speed by more than 5 km/h (1,4 m/s).</p> <p>Result: <b>[Achieved]</b> Successfully demonstrated in simulation for speeds of up to 17.5 km/h (Figure 49).</p>
R2.09	<p>The system must control the longitudinal motion of the vehicle so that it does not exceed a steady-state accuracy of more than 1 km/h (0.3 m/s).</p> <p>Result: <b>[Achieved]</b> Successfully demonstrated in simulation for speeds of up to 17.5 km/h (Figure 49).</p>
R2.10	<p>The system's speed with which it reaches its steady-state values must be balanced, so that neither one of the previous requirements is violated.</p> <p>Result: <b>[Achieved]</b></p>
R2.10	<p>The system must be designed and tested using a model-based design approach. For this purpose, a simulation and a vehicle model must be used. The vehicle model must represent the motion of H14DV in a way that reasonably balances the tradeoff between accuracy and complexity. This means the level of detail of the model must be limited to a practical extent, in order to avoid unnecessary complication. The physical effects and limitations of the vehicle must be analyzed and considered within the design of the model. This includes for instance factors like time delays, range limitations, and saturation effects.</p> <p>Result: <b>[Achieved]</b></p>



# 9 Conclusion and Outlook

This chapter will summarize the significant results of this thesis and will provide an outlook on how the development of the vehicle control module is planned to be continued.

## 9.1 Conclusion

The objective of this thesis was the development of a vehicle control system for an autonomous Formula Student vehicle. First, the problem was described and analyzed by discussing the details of the Formula Student competition, the vehicle H14DV, the control structure, and the issue of testing. Then, the essentials of vehicle modeling and the state of the art of autonomous vehicle control were presented by surveying several different control techniques commonly used in autonomous vehicles. Afterwards, the requirements for the vehicle control system were analyzed and the surveyed control techniques were evaluated. Several concepts for longitudinal and a lateral control of H14DV were proposed and compared with each other. The concept of a PI-controller with fixed parameters for longitudinal control and a Pure Pursuit controller for lateral control were chosen to best fulfill the defined requirements. The controllers and the vehicle model were designed and implemented in MATLAB/Simulink to allow for a model-based control system development. The system was tested and tuned through simulation of the vehicle dynamics on trajectories of FSG racetracks. Finally, the simulation results were evaluated and compared to the requirements. The implemented controllers achieved to fulfill all of the defined requirements. The vehicle control system demonstrated in simulation to be stable and accurate enough to be used for speeds of up to 15 km/h (4.2 m/s) on the FSG racetracks. It was also demonstrated that the system is capable of controlling H14DV during an autonomous brake test.

## 9.2 Outlook

Although the system achieved in simulation all of the defined requirements, there are certainly a few aspects that could be improved in the future. The PI-controller used for longitudinal control could use automatic gain-scheduling to improve the control performance and to allow for a greater operating range. Also, a different simulation environment could be used to improve the design process of the controllers and to allow for better testing. While MATLAB/Simulink did simplify the design process by providing a tool to create the vehicle trajectories, several useful toolboxes, and a functioning vehicle

visualization, the software lacked the key functionality of being able to incorporate other programming languages. Simulink does provide C- and MATLAB-function blocks, however, these cannot be compared to actual source code. Due to this issue with MATLAB/Simulink I suggest using a more sophisticated simulation environment in the future, such as Gazebo or Webots. These open-source robotic simulators allow the integration of C++ and Python source code and can also be used in combination with ROS.

The next step in the realization of this project will be to implement the vehicle control in source code and to integrate it into the ROS framework. Then, the interaction between the vehicle control module system and the other high-level software modules can be tested. Further, the communication between the high- and low-level components will have to be established using the CAN-bus, so that the actuators can be controlled. Also, the low-level controllers will have to be tuned and the  $PT_1$  parameters for the vehicle model will have to be determined through experiments. Once all the hardware components are operational and individually tested, the vehicle as a whole will be tested extensively to ensure safety and to collect sensor data. With this data the developed vehicle model will be validated and improved. The vehicle control system will be adjusted to the improved model and then the first real autonomous test drive can occur under safe conditions. Throughout the testing process, the parameters of the controller are fine-tuned so that the controller's performance can be perfected. After completing the testing of the vehicle control system on the autonomous vehicle, H14DV is planned to participate in the Formula Student Germany 2021. Here it will attempt to successfully pass the scrutineering and to complete all four disciplines. The knowledge and the acquired skills gained through the development of H14DV will be applied in the development of the next generation of vehicles, built by the HAWKS Racing team. In the future, more advanced control techniques, like MPC, can be implemented to further improve the control of the vehicle. For the future there is still plenty to work ahead, however the foundation for the vehicle control system is established.

# Bibliography

- [1] Statista (2020). By 2030, one in 10 vehicles will be self-driving globally. [online] Available at: [https://mailchi.mp/statista/autonomous\\_cars\\_20200206?e=145345a469](https://mailchi.mp/statista/autonomous_cars_20200206?e=145345a469). Accessed 13 October 2020.
- [2] Isermann, R. (Ed.) (2006). *Fahrdynamik-Regelung: Modellbildung, Fahrerassistenzsysteme, Mechatronik*; (1st edn, ATZ/MTZ-Fachbuch). Wiesbaden: Vieweg.
- [3] Formula Student Germany (2019). FSG Competition Handbook 2020, Version: 1.0, Rev-112717a. [online] Available at: [https://www.formulastudent.de/fileadmin/user\\_upload/all/2020/rules/FSG20\\_Competition\\_Handbook\\_v1.0.pdf](https://www.formulastudent.de/fileadmin/user_upload/all/2020/rules/FSG20_Competition_Handbook_v1.0.pdf). 13 September 2020.
- [4] Formula Student Germany (2019). Formula Student Rules 2020, Version: 1.0, Rev-1fcc153. [online] Available at: [https://www.formulastudent.de/fileadmin/user\\_upload/all/2020/rules/FSG20\\_Competition\\_Handbook\\_v1.0.pdf](https://www.formulastudent.de/fileadmin/user_upload/all/2020/rules/FSG20_Competition_Handbook_v1.0.pdf). Accessed 21 October 2020.
- [5] Intel® (2018). NUC Kits NUC8i7BEK and NUC8i7BEH Product Brief. [online] Available at: <https://www.intel.de/content/dam/www/public/us/en/documents/product-briefs/nuc8i7beh-product-brief.pdf>. Accessed 31 December 2020.
- [6] Gehrman, P. (2020). *Entwicklung einer LiDAR- und kamerabasierten Wegmarkierungserkennung für autonome Fahrzeuge*. HAW Hamburg.
- [7] Schwarz, L. (2020). *Entwicklung des elektronischen Systems für ein fahrerloses Rennauto*. HAW Hamburg.
- [8] Heinzmann GmbH & Co. KG (2016). Datenblatt PMS 120 - 48 VDC - 3000 rpm - ID 705\_b - WB 402\_b.
- [9] Waslander, S. (2020). Lesson 2: The Kinematic Bicycle Model - Module 4: Vehicle Dynamic Modeling | Coursera. [online] Available at: <https://www.coursera.org/lecture/intro-self-driving-cars/lesson-2-the-kinematic-bicycle-model-Bi8yE>. Accessed 25 October 2020.
- [10] Schildbach, G. (2020). Vehicle Dynamics & Control - 05 Kinematic bicycle model - YouTube. [online] Available at: [https://www.youtube.com/watch?v=HqNdBiej23I&ab\\_channel=Prof.GeorgSchildbach%2CUniversityofLuebeck](https://www.youtube.com/watch?v=HqNdBiej23I&ab_channel=Prof.GeorgSchildbach%2CUniversityofLuebeck). 13 September 2020.
- [11] Waslander, S. (2020). ME 597: Autonomous Mobile Robotics Section 4 – Control, ME 597. [online] Available at: [http://wavelab.uwaterloo.ca/sharedata/ME597/ME597\\_Lecture\\_Slides/ME597-7-Control.pdf](http://wavelab.uwaterloo.ca/sharedata/ME597/ME597_Lecture_Slides/ME597-7-Control.pdf). Accessed 31 December 2020.
- [12] Rajamani, R. (2012). *Vehicle Dynamics and Control*. Boston, MA: Springer US.

## Bibliography

---

- [13] Waslander, S. (2020). Lesson 4: Longitudinal Vehicle Modeling - Module 4: Vehicle Dynamic Modeling | Coursera. [online] Available at: <https://www.coursera.org/lecture/intro-self-driving-cars/lesson-4-longitudinal-vehicle-modeling-V8htX>. Accessed 1 January 2021.
- [14] Waslander, S. (2020). Lesson 5: Lateral Dynamics of Bicycle Model - Module 4: Vehicle Dynamic Modeling | Coursera. [online] Available at: <https://www.coursera.org/lecture/intro-self-driving-cars/lesson-5-lateral-dynamics-of-bicycle-model-1Opvo>. Accessed 2 January 2021.
- [15] Jarrod M. Snider (2009). *Automatic Steering Methods for Autonomous Automobile Path Tracking: CMU-RI-TR-09-08*. Pittsburgh, Pennsylvania.
- [16] Paden, B., Cap, M., Yong, S. Z., Yershov, D., & Frazzoli, E. (2016). *A Survey of Motion Planning and Control Techniques for Self-driving Urban Vehicles*. [online] Available at: <http://arxiv.org/pdf/1604.07446v1>. Accessed 31 December 2020.
- [17] Buehler, M., Iagnemma, K., & Singh, S. (2007). *The 2005 DARPA grand challenge: The great robot race* (Springer tracts in advanced robotics, Vol. 36). Berlin, New York: Springer.
- [18] Gabriel M. Hoffmann, Claire J. Tomlin (2007). *Autonomous Automobile Trajectory Tracking for Off-Road Driving: Controller Design, Experimental Validation and Racing*. Stanford, CA. [online] Available at: [http://ai.stanford.edu/~gabeh/papers/hoffmann\\_stanley\\_control07.pdf](http://ai.stanford.edu/~gabeh/papers/hoffmann_stanley_control07.pdf). Accessed 13 September 2020.
- [19] Ladoiye, J. S., Neculescu, D. S., & Sasiadek, J. (2018). Force Control of Surgical Robot with Time Delay using Model Predictive Control. In *15th International Conference on Informatics in Control, Automation and Robotics*.
- [20] Madani, K. (2018). *15th International Conference on Informatics in Control, Automation and Robotics, Porto, Portugal, 29.07.2018 - 31.07.2018* : SCITEPRESS - Science and Technology Publications.
- [21] MathWorks (2021). Understanding Model Predictive Control, Part 3: MPC Design Parameters - YouTube. [online] Available at: [https://www.youtube.com/watch?v=dAPRaml6k7Q&list=PLn8PRpmsu08ozoeoXgxPSBKLyd4YEHww8&index=3&ab\\_channel=MATLAB](https://www.youtube.com/watch?v=dAPRaml6k7Q&list=PLn8PRpmsu08ozoeoXgxPSBKLyd4YEHww8&index=3&ab_channel=MATLAB). Accessed 5 January 2021.
- [22] Waslander, S. (2021). Lesson 4: Advanced Steering Control - MPC - Module 6: Vehicle Lateral Control | Coursera. [online] Available at: <https://www.coursera.org/lecture/intro-self-driving-cars/lesson-4-advanced-steering-control-mpc-WHPvn>. Accessed 5 January 2021.
- [23] Farag, W. (2020). Complex Trajectory Tracking Using PID Control for Autonomous Driving. *International Journal of Intelligent Transportation Systems Research*.
- [24] Ang, K. H., Chong, G., & Li Yun (2005). PID control system analysis, design, and technology. *IEEE Transactions on Control Systems Technology*.
- [25] Douglas, B. (2018). *Understanding PID Control*. [online] Available at: <https://www.youtube.com/playlist?list=PLn8PRpmsu08pQBgjxYFXsODEF3Jqmm-y>. Accessed 5 January 2021.
- [26] Douglas, B. (2018). Control Systems in Practice, Part 2: What is Gain Scheduling?. [online] Available at: [https://www.youtube.com/watch?v=YiUjAV1bhKs&ab\\_channel=MATLAB](https://www.youtube.com/watch?v=YiUjAV1bhKs&ab_channel=MATLAB). Accessed 5 January 2021.

## Bibliography

---

- [27] Leith, D. J. (2000) *Survey of Gain-Scheduling Analysis & Design*. Glasgow. [online] Available at: [http://mural.maynoothuniversity.ie/1834/1/1001965764\\_link\\_19992.pdf](http://mural.maynoothuniversity.ie/1834/1/1001965764_link_19992.pdf). Accessed 5 January 2021.
- [28] Douglas, B. (2018). Control Systems in Practice, Part 3: What is Feedforward Control?. [online] Available at: [https://www.youtube.com/watch?v=FW\\_ay7K4jPE&ab\\_channel=MATLAB](https://www.youtube.com/watch?v=FW_ay7K4jPE&ab_channel=MATLAB). Accessed 13 September 2020.
- [29] Sailan, K., & Kuhnert, K. (2013). Modeling and Design of Cruise Control System with Feedforward for all Terrain Vehicles. In *Second International Conference on Advanced Information Technologies and Applications, November 02, 2013* (pp. 339–349): Academy & Industry Research Collaboration Center (AIRCC).
- [30] Vavra, C. (2019). The basics of model-following control. [online] Available at: <https://www.controleng.com/articles/the-basics-of-model-following-control/>. Accessed 7 January 2021.
- [31] Dangel, M., Vazquez, J. (2020). „FSG Driverless Academy 20200829 + 20200926 – Waymo LLC - How to: Controls“. FSG Driverless Academy.
- [32] J. Kabzan et al. (2019). „AMZ Driverless: The Full Autonomous Racing System“. [online] Available at: <http://arxiv.org/pdf/1905.05150v1>. Accessed 31 December 2020.
- [33] *FSG: Partners & Supporters*. [online] Available at: <https://www.formulastudent.de/partners/>. Accessed 31 December 2020.
- [34] Maxon (2020). Datasheet - EPOS4 Compact 50/15 CAN, digital positioning controller, 15 A, 10 - 50 VDC. [online] Available at: <https://www.maxongroup.com/maxon/view/product/control/Positionierung/520886>. Accessed 31 December 2020.
- [35] Nidec Motor Corporation (2020). GBL26xx Motor Controller Datasheet. [online] Available at: <https://www.roboteq.com/products/products-brushless-dc-motor-controllers/gbl2660-detail>. Accessed 13 September 2020.
- [36] Bandow, P. (2019). „Sensorfusion in autonomen Fahrzeugen“. HAW Hamburg.
- [37] Nidec Motor Corporation (2019). User Manual - Advanced Brushed and Brushless Digital Motor Controllers. [online] Available at: <https://www.roboteq.com/docman-list/motor-controllers-documents-and-files/documentation/user-manual/272-roboteq-controllers-user-manual-v17/file>. Accessed 31 December 2020.
- [38] MathWorks (2021). 3DOF rigid vehicle body to calculate longitudinal, lateral, and yaw motion. [online] Available at: <https://www.mathworks.com/help/vdynblks/ref/vehiclebody3dof.html>. Accessed 31 December 2020.
- [39] MathWorks Student Competitions Team (2020). Vehicle Path Tracking Using Pure Pursuit Controller. [online] Available at: <https://github.com/mathworks/vehicle-pure-pursuit/releases/tag/v1.1.1>. Accessed 16 October 2020.
- [40] Simulink (2020). Longitudinal wheel with disc, drum, or mapped brake. [online] Available at: <https://www.mathworks.com/help/vdynblks/ref/longitudinalwheel.html>. Accessed 31 December 2020.

## Bibliography

---

- [41] Formula Student Germany (2019) 05\_Tires\_in\_FSG\_FSG\_Academy\_20191114. [online] Available at: [https://www.formulastudent.de/fileadmin/user\\_upload/all/2019/academy/20191116\\_Magna/Slides\\_aus\\_root/05\\_Tires\\_in\\_FSG\\_FSG\\_Academy\\_20191114\\_Untergruppenbach.pdf](https://www.formulastudent.de/fileadmin/user_upload/all/2019/academy/20191116_Magna/Slides_aus_root/05_Tires_in_FSG_FSG_Academy_20191114_Untergruppenbach.pdf). Accessed 4 September 2020.
- [42] KA-Racing e.V. (2019). KA-Racing - KIT19d Trackdrive - YouTube. [online] Available at: [https://www.youtube.com/watch?v=B3KfAhKuy-s&ab\\_channel=KA-Racing.V.](https://www.youtube.com/watch?v=B3KfAhKuy-s&ab_channel=KA-Racing.V.) . Accessed 4 January 2021.
- [43] Engineering ToolBox (2004). Friction and Friction Coefficients. [online] Available at: [https://www.engineeringtoolbox.com/friction-coefficients-d\\_778.html](https://www.engineeringtoolbox.com/friction-coefficients-d_778.html). Accessed 15 February 2020.
- [44] Kadir, A., et al. (2014): Vehicle Speed Control Using Gain Scheduling PID with Experimental Throttle-in-the-Loop. International Review on Modelling and Simulations, Vol. 7, no. 4.
- [45] Delft Driverless (2019). Formula Student Germany 2019 - Driverless Diaries #4. [online] Available at: [https://www.youtube.com/watch?v=memXWrWbLlg&ab\\_channel=DelftDriverless](https://www.youtube.com/watch?v=memXWrWbLlg&ab_channel=DelftDriverless). Accessed 23 February 2021.
- [46] Gnanasambandham, C. (2015). MF\_Tire\_GUI: Magic Formula Parameter Visualization and Tire Model Fitting GUI. [online] Available at: [https://www.mathworks.com/matlabcentral/fileexchange/48568-mf\\_tire\\_gui](https://www.mathworks.com/matlabcentral/fileexchange/48568-mf_tire_gui). Accessed 25 February 2021.

# Appendix

## A.1 Intel NUC Specifications [5, P. 4]

### INTEL® NUC KIT NUC8I7BEH

#### Technical Specifications

<p><b>Processor</b></p> <ul style="list-style-type: none"> <li>• Intel® Core™ i7-8559U processor (2.7 GHz – 4.5 GHz, Quad Core, 8MB Cache, 28W TDP)</li> </ul> <p><b>Graphics</b></p> <ul style="list-style-type: none"> <li>• Iris® Plus Graphics 655</li> <li>• HDMI*, 2.0a port with 4K at 60 Hz</li> <li>• USB Type-C port with DisplayPort* 1.2</li> </ul> <p><b>System Memory</b></p> <ul style="list-style-type: none"> <li>• Two DDR4 SO-DIMM sockets (up to 32 GB, 2400 MHz), 1.2V</li> </ul> <p><b>Intel® Optane™ Memory</b></p> <ul style="list-style-type: none"> <li>• Room for up to 32 GB in an M.2 slot</li> </ul> <p><b>Storage Capabilities</b></p> <ul style="list-style-type: none"> <li>• Micro SDXC slot with UHS-I support on the side</li> <li>• One M.2 connector supporting 2.5" HDD or 22x80 M.2 SSD</li> <li>• One SATA 3 port for connection to 2.5" HDD or SSD (up to 9.5 mm thickness)</li> </ul> <p><b>Peripheral Connectivity</b></p> <ul style="list-style-type: none"> <li>• Intel® Gigabit LAN</li> <li>• Thunderbolt™ 3 with data transfer up to 40 Gbps</li> <li>• Four USB 3.1 ports (two back panel ports and two front ports including one charging port)</li> <li>• Two additional USB 2.0 ports via internal header</li> </ul>	<p><b>Front Panel Header</b></p> <ul style="list-style-type: none"> <li>• Reset, HDD LED, Power LEDs, power on/off</li> </ul> <p><b>Mechanical Chassis Size</b></p> <ul style="list-style-type: none"> <li>• 4.60" x 4.40" x 2.01"</li> <li>• 117 mm x 112 mm x 51 mm</li> </ul> <p><b>Baseboard Power Requirements</b></p> <ul style="list-style-type: none"> <li>• 19V, 90W power brick</li> </ul> <p><b>Environment Operating Temperature</b></p> <ul style="list-style-type: none"> <li>• 0° C to +40° C</li> </ul> <p><b>Storage Temperature</b></p> <ul style="list-style-type: none"> <li>• -20° C to +60° C</li> </ul> <p><b>Product Safety Regulations and Standards</b></p> <ul style="list-style-type: none"> <li>• IEC 60950-1</li> <li>• UL 60950-1</li> <li>• EN 60950-1</li> <li>• CAN/CSA-C22.2 No. 60950-1</li> </ul> <p><b>EMC/RF Regulations and Standards (Class B)</b></p> <ul style="list-style-type: none"> <li>• CISPR 32</li> <li>• FCC CFR Title 47, Chapter 1, Part 15, Subparts B, C, E</li> <li>• ICES-003</li> <li>• EN 55032</li> <li>• EN 55024</li> <li>• ETSI EN 300 328</li> </ul>	<p><b>System Bios</b></p> <ul style="list-style-type: none"> <li>• Intel® Dual Band Wireless-AC 9560 (802.11ac), 2x2, up to 1.73 Gbps</li> <li>• Dual Mode Bluetooth* 5.0</li> </ul> <p><b>System Bios</b></p> <ul style="list-style-type: none"> <li>• 64-Mb Flash EEPROM with Intel® Platform Innovation Framework for EFI Plug and Play</li> <li>• Advanced configuration and power interface V3.0b, SMBIOS2.5</li> <li>• Intel® Visual BIOS</li> <li>• Intel® Express BIOS update support</li> </ul> <p><b>Hardware Management Features</b></p> <ul style="list-style-type: none"> <li>• Processor fan speed control</li> <li>• Voltage and temperature sensing</li> <li>• Fan sensor inputs used to monitor fan activity</li> <li>• ACPI-compliant power management control</li> </ul> <p><b>Expansion Capabilities</b></p> <ul style="list-style-type: none"> <li>• Two internal USB 2.0 ports via two 1x4 headers (for replaceable lid support)</li> <li>• One Consumer Electronics Control header</li> </ul> <p><b>Audio</b></p> <ul style="list-style-type: none"> <li>• Up to 7.1 surround audio via HDMI</li> <li>• Headphone/microphone jack on the front panel</li> <li>• Dual-array microphones on the front panel</li> </ul>	<p><b>Front Panel Header</b></p> <ul style="list-style-type: none"> <li>• ETSI EN 301 489-1</li> <li>• ETSI EN 301 489-17</li> <li>• ETSI EN 301 893</li> <li>• EN 62311</li> <li>• AS/NZS 2772.2</li> <li>• AS/NZS 4268</li> <li>• VCCI V-2, V-3, V-4</li> <li>• KN-32</li> <li>• KN-24</li> <li>• CNS 13438</li> </ul> <p><b>Environmental Regulations</b></p> <ul style="list-style-type: none"> <li>• RoHS Directive 2011/65/EU</li> <li>• WEEE Directive 2012/19/EU</li> <li>• China RoHS – Management Methods for Restricted Use of Hazardous Substances in Electrical and Electronic Products</li> </ul>
---	---	---	---

Look for Intel® NUC with Intel Inside® at [www.intel.com/NUC](http://www.intel.com/NUC)

experience  
what's inside™

\*Benchmark results were obtained prior to implementation of recent software patches and feature updates intended to address exploits referred to as Spectre and Meltdown. These results are not intended for use in medical, life-saving, or life-sustaining applications. Intel may make changes to specifications and product descriptions at any time, without notice.

All products, data, and figures specified are preliminary based on current expectations, and are subject to change without notice. Availability in different countries may vary.

Actual Intel® NUC kit may differ from the image shown.

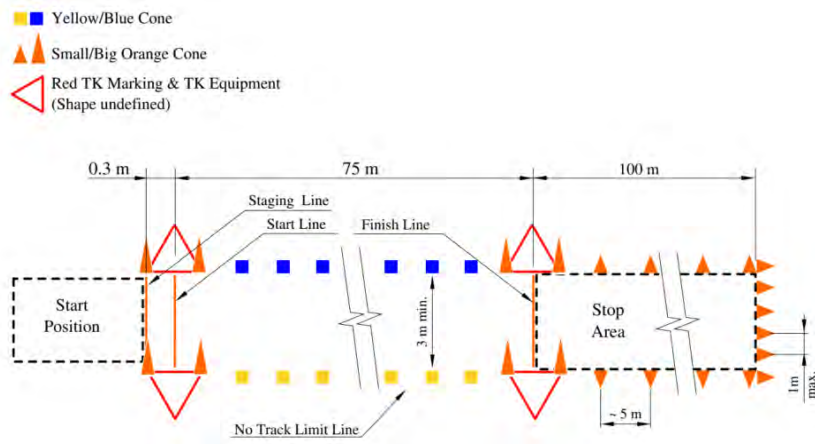
© 2018 Intel Corporation. All rights reserved. Intel, the Intel logo, Intel Inside, Intel Experience What's Inside logo, Intel Optane, Intel Optane memory vs. Intel Core™ i7-8750H processor + 1 TB HDD + 32 GB Intel® Optane™ memory vs. Intel Core™ i7-8750H processor + 1 TB HDD.

\*Other names and brands may be claimed as the property of others.

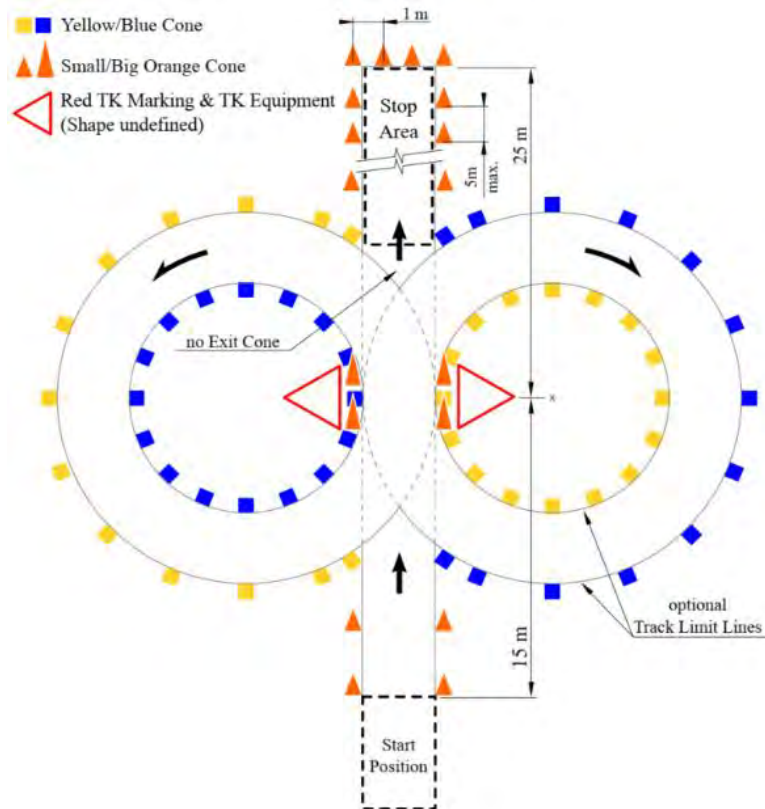




**A.3 Racetrack Acceleration Discipline [3, P.13]**

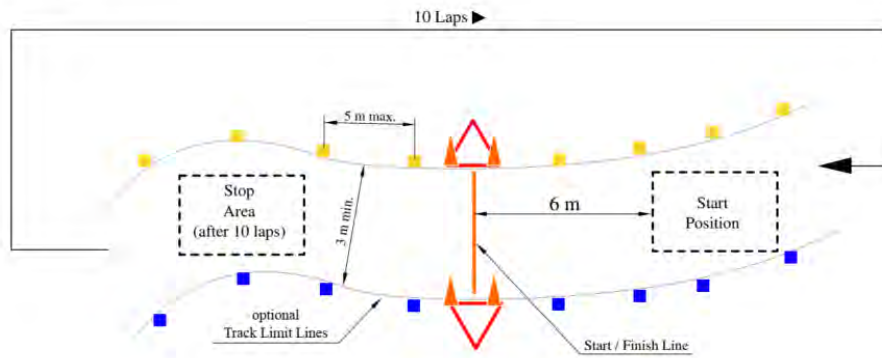


**A.4 Racetrack Skidpad Discipline [3, P.15]**



**A.5 Racetrack Autocross and Trackdrive Disciplines [3, P.14]**

- Yellow/Blue Cone
- ▲ Small/Big Orange Cone
- ◁ Red TK Marking & TK Equipment (Shape undefined)



## A.6 Electric Drive Motor Datasheet [8]



## PMS 120, ID 705b

## Datenblatt

08.03.2016

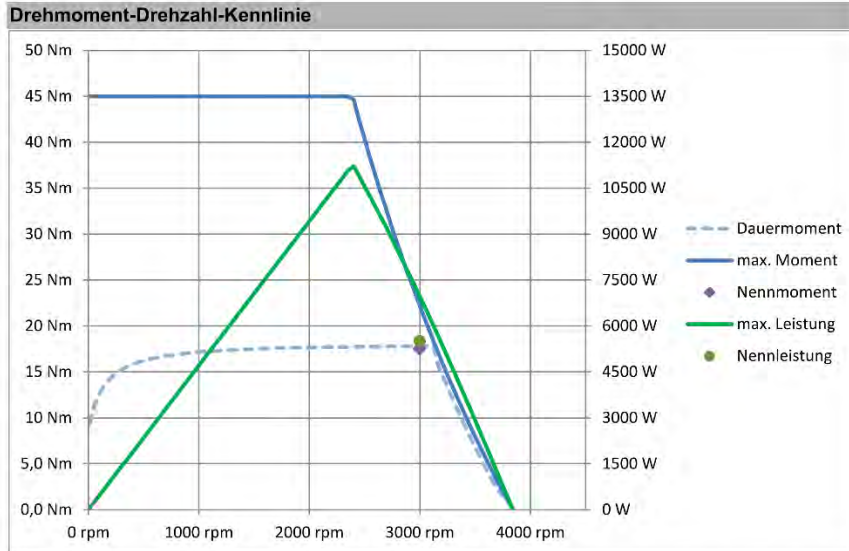
<b>Motortyp</b>			
Motor		PMS 120	
Motor ID		ID 705b	
Wickelblatt		WB 402b	
Polzahl	2p	8	
Kühlart		Luftkühlung mit mind. 5 m/s	
<b>Allg. Elektrische Daten</b>			
DC-Bus Spannung	$U_{dc}$	48	VDC
Spannungskonstante	$K_E$	8,84	V/1000rpm
Drehmomentkonstante	$K_M$	0,138	Nm/A
Wicklungswiderstand	$R_{2ph}$	11,7	m $\Omega$
Induktivität	$L_{2ph}$	46	$\mu$ H
<b>Betriebspunkt 1</b>			
Betriebsart		S1	
Motornennspannung	$U_n$	32,4	VAC
Nenndrehzahl	$n_n$	3000	rpm
Nennmoment	$M_n$	17,5	Nm
Nennleistung	$P$	5,5	kW
Motornennstrom	$I_n$	127	A
Frequenz	$f_n$	200	Hz
Wirkungsgrad	$\eta$	92,3	%
<b>Mechanische Daten</b>			
Maximaldrehzahl	$n_{max}$	3840	rpm
Spitzenleistung	$P_{max}$	11,1	kW
Spitzenmoment	$M_{max}$	45	Nm
Spitzenstrom	$I_{max}$	330	A
Trägheitsmoment	$J$	26,3	kg cm <sup>2</sup>
Gewicht	$m$	12,3	kg
Luftspalt	$s$	1,2	mm
Magnetmaterial		N35UH	
Spezielle Ausführung		...	
<b>Sensorik</b>			
Temperatursensor		...	

Heinzmann GmbH &amp; Co. KG

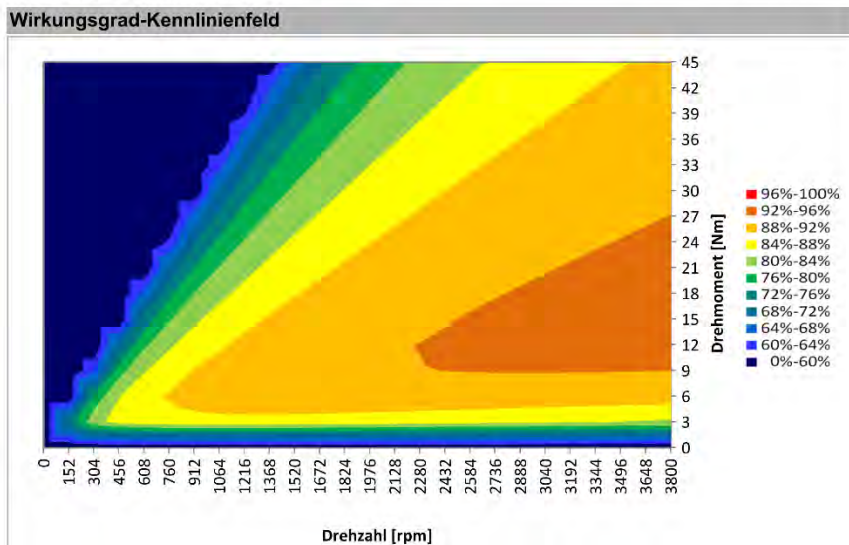
Electric &amp; Hybrid Drives

Am Haselbach 1 • D-79677 Schönau/Germany • Tel. +49 7673 8208 – 0 • Fax +49 7673 8208 - 799

E-Mail: info@heinzmann.de • www.heinzmann.com



**Bitte Beachten**  
 Betriebspunkte, deren Drehmoment-Anforderung oberhalb des Dauermoments liegen, können nicht für den Dauerbetrieb (S1) verwendet werden.



**Bitte Beachten**  
 Das Wirkungsgrad-Kennlinienfeld beruht auf Rechenwerten. Der tatsächlich erreichbare Wirkungsgrad kann durch Fertigungstoleranzen und Einflüsse durch den Motorregler leicht abweichen.

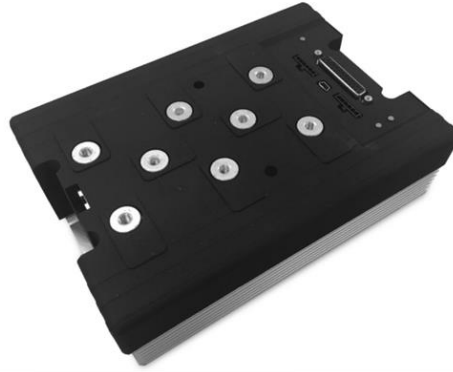
**Heinzmann GmbH & Co. KG**  
 Electric & Hybrid Drives

Am Haselbach 1 • D-79677 Schönau/Germany • Tel. +49 7673 8208 – 0 • Fax +49 7673 8208 - 799  
 E-Mail: info@heinzmann.de • www.heinzmann.com

## A.7 Electric Drive Motor Controller Datasheet [35]




## High Power 2 x 180A or 1 x 360A Brushless P Motor Controller with USB, CAN and Ethernet



Roboteq's GBL26xx is a feature-packed, high-current, dual or single channel controller for brushless Permanent Magnets AC or DC Synchronous motors. The controller also uses the Hall sensor and/or most types of Encoders to capture the Rotor position and measure traveled distance. The motors may be operated in open or closed loop speed mode, position mode or in torque mode. The GBL26xx features several Analog, Pulse and Digital I/Os which can be remapped as command or feedback inputs, limit switches, or many other functions. The GBL26xx accepts commands received from an RC radio, Analog Joystick, wireless modem, or microcomputer. For mobile robot applications, the controller's two motor channels can either be operated independently or mixed to move and steer a vehicle. Using CAN bus, up to 127 controllers can be networked at up to 1Mbit/s on a single twisted pair. An optional Ethernet port with PC allows the connection to PLCs and TCP/IP networks.

Numerous safety features are incorporated into the controller to ensure reliable and safe operation. The controller's operation can be extensively automated and customized using Basic Language scripts. The controller can be configured, monitored and tuned in real-time using a Roboteq's free PC utility. The controller can also be reprogrammed in the field with the latest features by downloading new operating software from Roboteq.

### Applications

- Automatic Guided Vehicles
- Small Electric Vehicles, Electric Bikes
- Terrestrial and Underwater Robotic Vehicles
- Police and Military Robots
- Hazardous Material Handling Robots
- Balancing Robots
- Telepresence Systems
- Animatronics

### Key Features

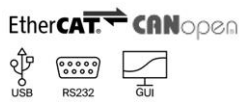
- USB, Serial, 0-5V Analog, or Pulse (RC radio) command modes
- One serial port
- CAN bus interface up to 1Mbit/s with multiple protocol support
- Optional RS485 interface
- Optional 10/100 Ethernet
- Auto switch between Serial, USB, CAN, Analog, or Pulse based on user-defined priority
- Built-in dual 3-phase high-power drivers for two brushless DC motor at up to 180A
- Output channels can be paralleled in order to drive a single motor at up to 360A
- Multiple Motor Operating mode
  - Trapezoidal with Hall Sensors
  - Sinusoidal with Incremental Encoder
  - Sinusoidal with SSI Encoder
  - Sinusoidal with Resolver
  - Sinusoidal with Hall Sensors
  - Sinusoidal with Sin/Cos Encoder
- Field Oriented Control in Sinusoidal modes
- Full forward & reverse motor control. Four quadrant operation. Supports regeneration
- Operates from a single 10V to 60V (120V optional) power source
- STO - Safe Torque Off support - Certification Pending
- Programmable current limit up to 180A (360A on single channel version) per motor for protecting controller, motor, wiring and battery.
- Separate connector for Hall Sensors
- Accurate speed and Odometry measurement using Hall Sensor or Encoder data



## A.9 Electric Steering Motor Controller Datasheet [34]

## EPOS4 Positioning Controllers Data

motor control

**EPOS4 Compact 50/8 EtherCAT**

Ready-to-install compact solution, designed for use with brushed DC motors with encoders or brushless EC motors with Hall sensors and encoders up to 400/1500 Watt.

**EPOS4 Compact 50/15 CAN**

Ready-to-install compact solution, designed for use with brushed DC motors with encoders or brushless EC motors with Hall sensors and encoders up to 750/1500 Watt.

Controller version	EtherCAT Slave	CANopen Slave
<b>Electrical data</b>		
Operating voltage $V_{CC}$	10 - 50 VDC	10 - 50 VDC
Logic supply voltage $V_C$ (optional)	10 - 50 VDC	10 - 50 VDC
Max. output voltage	$0.9 \times V_{CC}$	$0.9 \times V_{CC}$
Max. output current $I_{max}$	30 A (<5 s)	30 A (<60 s)
Continuous output current $I_{cont}$	8 A	15 A
Switching frequency of power stage	50 kHz	50 kHz
Sampling rate of PI current controller	25 kHz (40 $\mu$ s)	25 kHz (40 $\mu$ s)
Sampling rate of PI speed controller	2.5 kHz (400 $\mu$ s)	2.5 kHz (400 $\mu$ s)
Sampling rate of PID position controller	2.5 kHz (400 $\mu$ s)	2.5 kHz (400 $\mu$ s)
Max. speed (1 pole pair)	50 000 rpm (sinusoidal), 100 000 rpm (block)	50 000 rpm (sinusoidal), 100 000 rpm (block)
Built-in motor choke per phase	2.2 $\mu$ H / 15 A	2.2 $\mu$ H / 15 A
<b>Inputs</b>		
Hall sensor signals	H1, H2, H3	H1, H2, H3
Encoder signals	A, A', B, B', I, I' (max. 6.25 MHz)	A, A', B, B', I, I' (max. 6.25 MHz)
Sensor signals	A, A', B, B', I, I', Clock, Clock', Data, Data'	A, A', B, B', I, I', Clock, Clock', Data, Data'
Digital inputs	4 (level switchable: logic/PLC)	4 (level switchable: logic/PLC)
Digital inputs "High-speed"	4, differential	4, differential
Analog inputs	2 (12-bit resolution, -10...+10 V)	2 (12-bit resolution, -10...+10 V)
CAN ID / DEV ID	configurable with DIP switch 1...5	configurable with DIP switch 1...5
<b>Outputs</b>		
Digital outputs	2	2
Digital outputs "High-speed"	1, differential	1, differential
Analog outputs	2 (12-bit resolution, -4...+4 V, max. 1 mA)	2 (12-bit resolution, -4...+4 V, max. 1 mA)
Encoder voltage output	+5 VDC, max. 70 mA	+5 VDC, max. 70 mA
Hall sensor voltage output	+5 VDC, max. 30 mA	+5 VDC, max. 30 mA
Auxiliary voltage output	+5 VDC, max. 150 mA	+5 VDC, max. 150 mA
<b>Interfaces</b>		
RS232	-	RxD; TxD (max. 115 200 bit/s)
CAN	-	high; low (max. 1 Mbit/s)
USB 2.0/3.0	Data+; Data- (Full Speed)	Data+; Data- (Full Speed)
EtherCAT	100 Mbit/s (Full Duplex)	-
<b>Indicator</b>		
LED green = READY, red = ERROR	Green LED, red LED	Green LED, red LED
<b>Environmental conditions</b>		
Temperature - Operation	-30...+45 °C	-30...+25 °C
Temperature - Extended Range	+45...+77 °C; Derating: -0.250 A/°C	+25...+77 °C; Derating: -0.288 A/°C
Temperature - Storage	-40...+85 °C	-40...+85 °C
Humidity (condensation not permitted)	5...90%	5...90%
<b>Mechanical data</b>		
Weight	approx. 100 g	approx. 126 g
Dimensions (L x W x H)	59.5 x 79.5 x 35.7 mm	59.5 x 65.5 x 35.1 mm
Mounting	M2.5 screws	M3 screws
<b>Part numbers</b>		
	605298 EPOS4 Compact 50/8 EtherCAT	520886 EPOS4 Compact 50/15 CAN
<b>Accessories</b>		
	235811 DSR 70/30 Shunt regulator	235811 DSR 70/30 Shunt regulator
	Order accessories separately, see page 512	Order accessories separately, see page 512

## Appendix

### A.10 Total Mass and Moment of Inertia Calculation Table H14DV

Part	Masse [kg]	CoG X [mm]	CoG Y [mm]	CoG Z [mm]	IoZG [kg*m <sup>2</sup> ]	d [CoG]	Izz [kg*m <sup>2</sup> ]	Weighted (W)/ Estimated (E)
LiDAR	2	-844	0	-43	0	1,6	5,12	W
Camera & TSAL	0,6	1051	0	720	0	0,3	0,05	W
DCU	8	1251	0	519	0	0,5	1,96	W
A-04 FWMounting Plates	0,696	-458	0	-54	0	1,2	1,03	W
A-04 FWMounting Left	0,125	-409	200	2	0	1,2	0,18	W
A-04 FWMounting Right	0,125	-409	-200	2	0	1,2	0,17	W
B-01 Monocoque	24	450	0	22	14,557	0,3	2,25	W
B-05 AIP Crashbox	1,079	-640	0	69,5	0	1,4	2,10	W
B-07 Main Hoop	8	1037,9	0	440	0	0,3	0,64	W
B-08 Neck Protect	0,2	1131,5	0	356	0	0,4	0,03	W
B-08 Shoulder Harness	0,8	1000	0	258	0	0,2	0,05	W
B-09 Head Restraint	0,9	1139	0	474	0	0,4	0,13	W
B-10 Rearframe + Spring Rear	4,035	1522	0	160	0	0,8	2,37	W
B-12 Belt system	4,020	800	0	100	0	0,0	0,01	W
B-13 Seat	1,385	685,5	0	21	0	0,1	0,01	W
B-14 Fire shield	0,7	900	0	-7	0	0,1	0,01	W
B-15 Exterior	1,1	-400	0	190	0	1,2	1,47	W
C-02 Spring Front	3,5	-18,25	0	352,8	0	0,8	2,10	W
C-03 Steering	1,09	180,5	0	51,4	0	0,6	0,36	W
C-04 Brake FL	0,2	20	-611	-113	0	1,0	0,61	W
C-04 Brake FR	0,2	20	611	-113	0	1,0	0,62	W
C-04 Brake RL	0,2	1469	-612	-87	0	0,9	0,17	W
C-04 Brake RR	0,2	1469	612	-87	0	0,9	0,18	W
C-04 Brake Lines	1	690	0	90	0	0,1	0,00	W
C-04-200 Brake Disc FL	0,69	0	-614,5	0	0	1,0	0,65	W
C-04-200 Brake Disc FR	0,69	0	614,5	0	0	1,0	0,66	W
C-04-200 Brake Disc RL	0,42	1550	-614,5	0	0	1,0	0,42	W
C-04-200 Brake Disc RR	0,42	1550	614,5	0	0	1,0	0,43	W
C-05 Wheel Assembly FL	1,1	2	-593	6,4	0	1,0	1,00	W
C-05 Wheel Assembly FR	1,1	2	593	6,4	0	1,0	1,02	W
C-05 Wheel Assembly RL	1,4	1549	-593	1,43	0	1,0	1,36	W



## Appendix

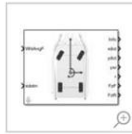
C-05 Wheel Assembly RR	1,4	1549	593	1,43	0	1,0	1,39	W
C-06 Pedal box	1,4	-359,2	-14	3,7	0	1,1	1,74	W
C-07 Steering Wheel	0,831	425	0	245	0	0,3	0,09	W
C-08 Tire FL	3,875	0	-600	0	0	1,0	3,57	W
C-08 Tire FR	3,875	0	600	0	0	1,0	3,65	W
C-08 Tire RL	3,875	1550	-600	0	0	1,0	3,80	W
C-08 Tire RR	3,875	1550	600	0	0	1,0	3,88	W
C-09 Wishbones FL	0,75	-20	-356	11	0	0,9	0,54	W
C-09 Wishbones FR	0,75	-20	-356	11	0	0,9	0,54	W
C-09 Wishbones RL	0,75	1532	-369	-49	0	0,9	0,55	W
C-09 Wishbones RR	0,75	1532	369	-49	0	0,9	0,56	W
C-69 Rim FL	3,63	0	-623	0	0	1,0	3,45	W
C-69 Rim FR	3,63	0	623	0	0	1,0	3,52	W
C-69 Rim RL	3,63	1550	-623	0	0	1,0	3,66	W
C-69 Rim RR	3,63	1550	623	0	0	1,0	3,73	W
F-01 Wire Harness	15	650	-50	-100	0	0,1	0,12	E
Accumulator Case	6,1	1124,7	-69	-68	0	0,4	0,85	W
G-01 Drivetrain	5	1524	-28	-3,3	0	0,8	2,95	W
Motor	12	1345	-33	-26	1,283	0,6	4,17	W
Motor Mounting	3	1378	-1,5	-97	0	0,6	1,16	E
Rear Plate	1	1697	0,7	-39	0	0,9	0,89	W
Diff Mount	1,6	1582,2	-55	-7	0	0,8	1,10	W
Correxit	0,25				0	0,8	0,00	W
EBS Pneumatic tanks	1,25	1456	112,3	13	0	0,7	0,63	W
EBS	10	-380	-8,8	3	0	1,1	6,45	E
Tractive System Box	10,64	939	-6,5	53,265	0	0,2	0,36	E
TS Cables	4	1310	-147,5	-93	0	0,6	1,30	W
HVD	0,7	1198,4	-250	185,7	0	0,5	0,18	E
Accumulator	20	1140	69,3	-56	0	0,4	1,53	E
Miscellaneous	10	611	0	120	0	0,1	1,52	E
<b>Total Sum Complete Vehicle</b>	<b>207,16</b>	<b>756,1</b>	<b>-8,4</b>	<b>69,3</b>			<b>85,0</b>	
<b>Axle load distribution</b>		<b>51,22</b>	<b>% front</b>	297,4	mm	From ground		
		<b>48,78</b>	<b>% rear</b>					

## A.11 Simulink “Vehicle-Body-3DOF-Block” Documentation [38]

### Vehicle Body 3DOF

3DOF rigid vehicle body to calculate longitudinal, lateral, and yaw motion

**Library:** Vehicle Dynamics Blockset / Vehicle Body



#### Description

The Vehicle Body 3DOF block implements a rigid two-axle vehicle body model to calculate longitudinal, lateral, and yaw motion. The block accounts for body mass and aerodynamic drag between the axles due to acceleration and steering.

Use this block in vehicle dynamics and automated driving studies to model nonholonomic vehicle motion when vehicle pitch, roll, and vertical motion are not significant.

In the Vehicle Dynamics Blockset™ library, there are two types of Vehicle Body 3DOF blocks that model longitudinal, lateral, and yaw motion.

Block	Vehicle Track Setting	Implementation
	Single (bicycle)	<ul style="list-style-type: none"> <li>Forces act along the center line at the front and rear axles.</li> <li>No lateral load transfer.</li> </ul>
	Dual	Forces act at the four vehicle corners or <i>hard points</i> .

Use the **Axle forces** parameter to specify the type of force.

Axle Forces Setting	Implementation

Axle Forces Setting	Implementation
External longitudinal velocity	<ul style="list-style-type: none"> <li>The block assumes that the external longitudinal velocity is in a quasi-steady state, so the longitudinal acceleration is approximately zero.</li> <li>Because the motion is quasi-steady, the block calculates lateral forces using the tire slip angles and linear cornering stiffness.</li> <li>Consider this setting when you want to:                             <ul style="list-style-type: none"> <li>Generate virtual sensor signal data.</li> <li>Conduct high-level software studies that are not impacted by driveline or nonlinear tire responses.</li> </ul> </li> </ul>
External longitudinal forces	<ul style="list-style-type: none"> <li>The block uses the external longitudinal force to accelerate or brake the vehicle.</li> <li>The block calculates lateral forces using the tire slip angles and linear cornering stiffness.</li> <li>Consider this setting when you want to:                             <ul style="list-style-type: none"> <li>Account for changes in the longitudinal velocity on the lateral and yaw motion.</li> <li>Specify the external longitudinal motion through a force instead of an external longitudinal velocity.</li> <li>Connect the block to tractive actuators, wheels, brakes, and hitches.</li> </ul> </li> </ul>
External forces	<ul style="list-style-type: none"> <li>The block uses the external lateral and longitudinal forces to steer, accelerate, or brake the vehicle.</li> <li>The block does not use the steering input to calculate vehicle motion.</li> <li>Consider this setting when you need tire models with more accurate nonlinear combined lateral and longitudinal slip.</li> </ul>

You can use these block parameters to create additional input ports. This table summarizes the settings.

Input Signals Pane Parameter	Input Port	Description
Front wheel steering	WhlAngF	Front wheel angle, $\delta_F$
External wind	WindXYZ	Wind speed, $W_x$ , $W_y$ , $W_z$ , in the inertial reference frame
External forces	FExt	External force on vehicle center of gravity (CG), $F_x$ , $F_y$ , $F_z$ , in the vehicle-fixed frame
Rear wheel steering	WhlAngR	Rear wheel angle, $\delta_R$
External friction	Mu	Friction coefficient
External moments	MExt	External moment about vehicle CG, $M_x$ , $M_y$ , $M_z$ , in vehicle-fixed frame
Hitch forces	Fh	Hitch force applied to the body at the hitch location, $Fh_x$ , $Fh_y$ , and $Fh_z$ , in the vehicle-fixed frame
Hitch moments	Mh	Hitch moment at the hitch location, $Mh_x$ , $Mh_y$ , and $Mh_z$ , about the vehicle-fixed frame
Initial longitudinal position	X_o	Initial vehicle CG displacement along the earth-fixed X-axis, in m
Initial lateral position	Y_o	Initial vehicle CG displacement along the earth-fixed Y-axis, in m
Initial longitudinal velocity	xdot_o	Initial vehicle CG velocity along the vehicle-fixed x-axis, in m/s
Initial lateral velocity	ydot_o	Initial vehicle CG velocity along the vehicle-fixed y-axis, in m/s
Initial yaw angle	psi_o	Initial rotation of the vehicle-fixed frame about the earth-fixed Z-axis (yaw), in rad

Input Signals Pane Parameter	Input Port	Description
Initial yaw rate	r_o	Initial vehicle angular velocity about the vehicle-fixed z-axis (yaw rate), in rad/s
Air temperature	AirTemp	Ambient air temperature. Considering this option if you want to vary the temperature during run-time.

**Theory**

The Vehicle Body 3DOF block implements a rigid two-axle vehicle body model to calculate longitudinal, lateral, and yaw motion. The block accounts for body mass, aerodynamic drag, and weight distribution between the axles due to acceleration and steering. To determine the vehicle motion, the block implements these equations for the single track, dual track, and drag calculations.

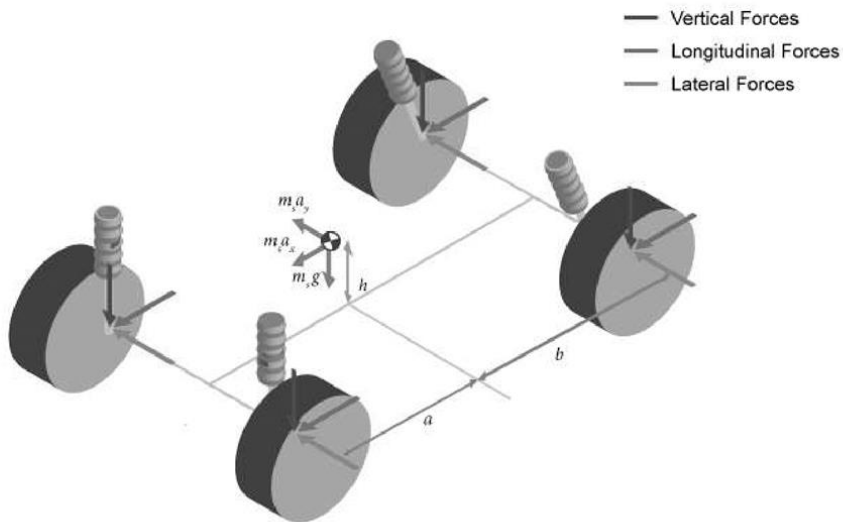
**Single Track**

Calculation	Description
Dynamics	<p>The block uses these equations to calculate the rigid body planar dynamics.</p> $\ddot{y} = -\dot{x}r + \frac{F_{yf} + F_{yr} + F_{yext}}{m}$ $\dot{r} = \frac{aF_{yf} - bF_{yr} + M_{zext}}{I_{zz}}$ $r = \dot{\psi}$ <p>If you set <b>Axle forces</b> to either External longitudinal forces or External forces, the block uses this equation for the longitudinal acceleration.</p> $\ddot{x} = \dot{y}r + \frac{F_{xf} + F_{xr} + F_{xext}}{m}$ <p>If you set <b>Axle forces</b> to External longitudinal velocity, the block assumes a quasi-steady state for the longitudinal acceleration.</p> $\ddot{x} = 0$
External forces	<p>External forces include both drag and external force inputs. The forces act on the vehicle CG.</p> $F_{x,y,z\ ext} = F_{d\ x,y,z} + F_{x,y,z\ input}$ $M_{x,y,z\ ext} = M_{d\ x,y,z} + M_{x,y,z\ input}$ <p>If you set <b>Axle forces</b> to External longitudinal forces, the block uses these equations.</p> $F_{xft} = F_{xfinput}$ $F_{yft} = -C_{yf}\alpha_f\mu_f \frac{F_{zf}}{F_{znom}}$ $F_{xft} = F_{xrinput}$ $F_{yft} = -C_{yr}\alpha_r\mu_r \frac{F_{zr}}{F_{znom}}$ <p>If you set <b>Axle forces</b> to External longitudinal velocity, the block uses these equations.</p> $F_{xft} = 0$ $F_{yft} = -C_{yf}\alpha_f\mu_f \frac{F_{zf}}{F_{znom}}$ $F_{xft} = 0$ $F_{yft} = -C_{yr}\alpha_r\mu_r \frac{F_{zr}}{F_{znom}}$ <p>The block divides the normal forces by the nominal normal load to vary the effective friction parameters during weight and load transfer. The block uses these equations to maintain pitch and roll equilibrium.</p> $F_{zf} = \frac{bmg - (\ddot{x} - \dot{y}r)mh + hF_{xext} + bF_{zext} - M_{yext}}{a + b}$ $F_{zr} = \frac{amg + (\ddot{x} - \dot{y}r)mh - hF_{xext} + aF_{zext} + M_{yext}}{a + b}$

Calculation	Description
Dynamics	<p>The block uses these equations to calculate the rigid body planar dynamics.</p> $\ddot{x} = \dot{y}r + \frac{F_{yfl} + F_{yfr} + F_{xrl} + F_{xrr} + F_{xext}}{m}$ $\ddot{y} = -\dot{x}r + \frac{F_{yfl} + F_{yfr} + F_{yrl} + F_{yrr} + F_{yext}}{m}$ $\dot{r} = \frac{a(F_{yfl} + F_{yfr}) - b(F_{yrl} + F_{yrr}) + \frac{w_f(F_{xfl} - F_{xfr})}{2} + \frac{w_r(F_{xrl} - F_{xrr})}{2} + M_{zext}}{I_{zz}}$ $r = \dot{\psi}$ <p>If you set <b>Axle forces</b> to External longitudinal velocity, the block assumes a quasi-steady state for the longitudinal acceleration.</p> $\ddot{x} = 0$
External forces	<p>External forces include both drag and external force inputs. The forces act on the vehicle CG.</p> $F_{x,y,z \ ext} = F_{d \ x,y,z} + F_{x,y,z \ input}$ $M_{x,y,z \ ext} = M_{d \ x,y,z} + M_{x,y,z \ input}$ <p>If you set <b>Axle forces</b> to External longitudinal forces, the block uses these equations.</p> $F_{xfl} = F_{xflinput}$ $F_{yfl} = -C_{yfl}\alpha_{fl}\mu_{fl} \frac{F_{zfl}}{2F_{znom}}$ $F_{xfr} = F_{xfrinput}$ $F_{yfr} = -C_{yfr}\alpha_{fr}\mu_{fr} \frac{F_{zfr}}{2F_{znom}}$ $F_{xrl} = F_{xrlinput}$ $F_{yrl} = -C_{yrl}\alpha_{rl}\mu_{rl} \frac{F_{zrl}}{2F_{znom}}$ $F_{xrr} = F_{xrrinput}$ $F_{yrr} = -C_{yrr}\alpha_{rr}\mu_{rr} \frac{F_{zrr}}{2F_{znom}}$ <p>If you set <b>Axle forces</b> to External longitudinal velocity, the block uses these equations.</p> $F_{xfl} = 0$ $F_{yfl} = -C_{yfl}\alpha_{fl}\mu_{fl} \frac{F_{zfl}}{2F_{znom}}$ $F_{xfr} = 0$ $F_{yfr} = -C_{yfr}\alpha_{fr}\mu_{fr} \frac{F_{zfr}}{2F_{znom}}$ $F_{xrl} = 0$ $F_{yrl} = -C_{yrl}\alpha_{rl}\mu_{rl} \frac{F_{zrl}}{2F_{znom}}$ $F_{xrr} = 0$ $F_{yrr} = -C_{yrr}\alpha_{rr}\mu_{rr} \frac{F_{zrr}}{2F_{znom}}$ <p>The block divides the normal forces by the nominal normal load to vary the effective friction parameters during weight and load transfer. The block uses these equations to maintain pitch and roll equilibrium.</p>

Calculation	Description
Tire forces	<p>The block uses the ratio of the local and longitudinal and lateral velocities to determine the slip angles.</p> $\alpha_f = \text{atan}\left(\frac{\dot{y} + ar}{\dot{x}}\right) - \delta_f$ $\alpha_r = \text{atan}\left(\frac{\dot{y} - br}{\dot{x}}\right) - \delta_r$ <p>To determine the tire forces, the block uses the slip angles.</p> $F_{xf} = F_{xft} \cos(\delta_f) - F_{yft} \sin(\delta_f)$ $F_{yf} = -F_{xft} \sin(\delta_f) + F_{yft} \cos(\delta_f)$ $F_{xr} = F_{xrt} \cos(\delta_r) - F_{yrt} \sin(\delta_r)$ $F_{yr} = -F_{xrt} \sin(\delta_r) + F_{yrt} \cos(\delta_r)$ <p>If you set <b>Axle forces</b> to External forces, the block sets the tire forces equal to the external input force.</p> $F_{xf} = F_{xft} = F_{xfinput}$ $F_{yf} = F_{yft} = F_{yfinput}$ $F_{xr} = F_{xrt} = F_{xrinput}$ $F_{yr} = F_{yrt} = F_{yrinput}$

Dual Track



Calculation	Description
-------------	-------------

Calculation	Description
	$F_{zf} = \frac{bmg - (\ddot{x} - \dot{y}r)mh + hF_{xext} + bF_{zext} - M_{yext}}{a + b}$ $F_{zr} = \frac{amg + (\ddot{x} - \dot{y}r)mh - hF_{xext} + aF_{zext} + M_{yext}}{(a + b)}$ $F_{zfl} = F_{zf} + (mh(\ddot{y} + \dot{x}r) - hF_{yext} - M_{xext}) \frac{2}{w_f}$ $F_{zfr} = F_{zf} + (-mh(\ddot{y} + \dot{x}r) + hF_{yext} + M_{xext}) \frac{2}{w_f}$ $F_{zrl} = F_{zr} + (mh(\ddot{y} + \dot{x}r) - hF_{yext} - M_{xext}) \frac{2}{w_r}$ $F_{zrr} = F_{zr} + (-mh(\ddot{y} + \dot{x}r) + hF_{yext} + M_{xext}) \frac{2}{w_r}$
Tire forces	<p>The block uses the ratio of the local and longitudinal and lateral velocities to determine the slip angles.</p> $\alpha_{fl} = \operatorname{atan}\left(\frac{\dot{y} + ar}{\dot{x} + r \frac{w_f}{2}}\right) - \delta_{fl}$ $\alpha_{fr} = \operatorname{atan}\left(\frac{\dot{y} + ar}{\dot{x} - r \frac{w_f}{2}}\right) - \delta_{fr}$ $\alpha_{rl} = \operatorname{atan}\left(\frac{\dot{y} - ar}{\dot{x} + r \frac{w_r}{2}}\right) - \delta_{rl}$ $\alpha_{rr} = \operatorname{atan}\left(\frac{\dot{y} - ar}{\dot{x} - r \frac{w_r}{2}}\right) - \delta_{rr}$ <p>The block uses the steering angles to transform the tire forces to the vehicle-fixed frame.</p> $F_{xf} = F_{xfl} \cos(\delta_f) - F_{yfl} \sin(\delta_f)$ $F_{yf} = -F_{xfl} \sin(\delta_f) + F_{yfl} \cos(\delta_f)$ $F_{xr} = F_{xrl} \cos(\delta_r) - F_{yrl} \sin(\delta_r)$ $F_{yr} = -F_{xrl} \sin(\delta_r) + F_{yrl} \cos(\delta_r)$ <p>If you set <b>Axle forces</b> to External forces, the block uses these equations. The blocks assumes that the externally provided forces are in the vehicle-fixed frame at the axle-wheel location.</p> $F_{xf} = F_{xfl} = F_{xfinput}$ $F_{yf} = F_{yfl} = F_{yfinput}$ $F_{xr} = F_{xrl} = F_{xrinput}$ $F_{yr} = F_{yrl} = F_{yrinput}$

**Drag**

Calculation	Description
Coordinate transformation	<p>The block transforms the wind speeds from the inertial frame to the vehicle-fixed frame.</p> $w_x = W_x \cos(\psi) + W_y \sin(\psi)$ $w_y = W_y \cos(\psi) - W_x \sin(\psi)$ $w_z = W_z$

Calculation	Description
Drag forces	<p>To determine a relative airspeed, the block subtracts the wind speed from the CG vehicle velocity. Using the relative airspeed, the block determines the drag forces.</p> $\bar{w} = \sqrt{(\dot{x}_b - w_x)^2 + (\dot{y}_b - w_y)^2 + (w_z)^2}$ $F_{dx} = -\frac{1}{2TR} C_d A_f P_{abs}(\bar{w})^2$ $F_{dy} = -\frac{1}{2TR} C_s A_f P_{abs}(\bar{w})^2$ $F_{dz} = -\frac{1}{2TR} C_l A_f P_{abs}(\bar{w})^2$
Drag moments	<p>Using the relative airspeed, the block determines the drag moments.</p> $M_{dr} = -\frac{1}{2TR} C_{rm} A_f P_{abs}(\bar{w})^2 (a + b)$ $M_{dp} = -\frac{1}{2TR} C_{pm} A_f P_{abs}(\bar{w})^2 (a + b)$ $M_{dy} = -\frac{1}{2TR} C_{ym} A_f P_{abs}(\bar{w})^2 (a + b)$

**Lateral Corner Stiffness and Relaxation Dynamics**

Description	Implementation
Constant values.	The block uses constant stiffness values for $C_{yf}$ and $C_{yr}$ .
Lookup tables as a function of corner stiffness data and slip angles.	<p>The block uses lookup tables that are functions of the corner stiffness data and slip angles.</p> $C_{yf} = f(\alpha_f, C_{yfdata})$ $C_{yr} = f(\alpha_r, C_{yrdata})$
Lookup tables as a function of corner stiffness data and slip angles. Slip angles include the relaxation length dynamic settings.	<p>The block uses lookup tables that are functions of the corner stiffness data and slip angles. The slip angles include the relaxation length dynamic settings. The relaxation length approximates an effective corner stiffness force that is a function of wheel travel.</p> $C_{yf} = f(\alpha_{f\sigma}, C_{yfdata})$ $C_{yr} = f(\alpha_{r\sigma}, C_{yrdata})$ $\alpha_{f\sigma} = \frac{1}{s} \left[ \frac{(\alpha_f - \alpha_{f\sigma}) v_{wf}}{\alpha_f} \right]$ $\alpha_{r\sigma} = \frac{1}{s} \left[ \frac{(\alpha_r - \alpha_{r\sigma}) v_{wr}}{\alpha_r} \right]$

The equations use these variables.

$x, \dot{x}, \ddot{x}$	Vehicle CG displacement, velocity, and acceleration, along the vehicle-fixed x-axis
$y, \dot{y}, \ddot{y}$	Vehicle CG displacement, velocity, and acceleration, along the vehicle-fixed y-axis
$\psi$	Rotation of the vehicle-fixed frame about the earth-fixed Z-axis (yaw)
$r, \dot{\psi}$	Vehicle angular velocity, about the vehicle-fixed z-axis (yaw rate)
$F_{xf}, F_{xr}$	Longitudinal forces applied to front and rear wheels, along the vehicle-fixed x-axis
$F_{yf}, F_{yr}$	Lateral forces applied to front and rear wheels, along vehicle-fixed y-axis
$F_{xext}, F_{yext}, F_{zext}$	External forces applied to vehicle CG, along the vehicle-fixed x-, y-, and z-axes
$F_{dx}, F_{dy}, F_{dz}$	Drag forces applied to vehicle CG, along the vehicle-fixed x-, y-, and z-axes
$F_{xinput}, F_{yinput}, F_{zinput}$	Input forces applied to vehicle CG, along the vehicle-fixed x-, y-, and z-axes
$M_{xext}, M_{yext}, M_{zext}$	External moment about vehicle CG, about the vehicle-fixed x-, y-, and z-axes



## Appendix

$M_{dx}, M_{dy}, M_{dz}$	Drag moment about vehicle CG, about the vehicle-fixed x-, y-, and z-axes
$M_{xinput}, M_{yinput}, M_{zinput}$	Input moment about vehicle CG, about the vehicle-fixed x-, y-, and z-axes
$I_{zz}$	Vehicle body moment of inertia about the vehicle-fixed z-axis
$F_{xfl}, F_{xrt}$	Longitudinal tire force applied to front and rear wheels, along the vehicle-fixed x-axis
$F_{yfl}, F_{yfr}$	Lateral tire force applied to front and rear wheels, along vehicle-fixed y-axis
$F_{xfl}, F_{xfr}$	Longitudinal force applied to front left and front right wheels, along the vehicle-fixed x-axis
$F_{yfl}, F_{yfr}$	Lateral force applied to front left and front right wheels, along the vehicle-fixed y-axis
$F_{xrl}, F_{xrr}$	Longitudinal force applied to rear left and rear right wheels, along the vehicle-fixed x-axis
$F_{yrl}, F_{yrr}$	Lateral force applied to rear left and rear right wheels, along the vehicle-fixed y-axis
$F_{xfls}, F_{xfrt}$	Longitudinal tire force applied to front left and front right wheels, along the vehicle-fixed x-axis
$F_{yfls}, F_{yfrt}$	Lateral force tire applied to front left and front right wheels, along the vehicle-fixed y-axis
$F_{xrls}, F_{xrtr}$	Longitudinal tire force applied to rear left and rear right wheels, along the vehicle-fixed x-axis
$F_{yrls}, F_{yrrt}$	Lateral force applied to rear left and rear right wheels, along the vehicle-fixed y-axis
$F_{zf}, F_{zr}$	Normal force applied to front and rear wheels, along vehicle-fixed z-axis
$F_{znom}$	Nominal normal force applied to axles, along the vehicle-fixed z-axis
$F_{zfl}, F_{zfr}$	Normal force applied to front left and right wheels, along vehicle-fixed z-axis
$F_{zrl}, F_{zrr}$	Normal force applied to rear left and right wheels, along vehicle-fixed z-axis
$m$	Vehicle body mass
$a, b$	Distance of front and rear wheels, respectively, from the normal projection point of vehicle CG onto the common axle plane
$h$	Height of vehicle CG above the axle plane
$d$	Lateral distance from the geometric centerline to the center of mass along the vehicle-fixed y-axis
$hh$	Height of the hitch above the axle plane along the vehicle-fixed z-axis
$dh$	Longitudinal distance of the hitch from the normal projection point of tractor CG onto the common axle plane
$hl$	Lateral distance from center of mass to hitch along the vehicle-fixed y-axis.
$\alpha_f, \alpha_r$	Front and rear wheel slip angles
$\alpha_{fl}, \alpha_{fr}$	Front left and right wheel slip angles
$\alpha_{rl}, \alpha_{rr}$	Rear left and right wheel slip angles
$\delta_f, \delta_r$	Front and rear wheel steering angles
$\delta_{fl}, \delta_{fr}$	Front left and right wheel steering angles
$\delta_{rl}, \delta_{rr}$	Rear left and right wheel steering angles
$w_f, w_r$	Front and rear track widths
$C_{yf}, C_{yr}$	Front and rear wheel cornering stiffness
$C_{yldata}, C_{yrd}$	Front and rear wheel cornering stiffness data
$\sigma_f, \sigma_r$	Front and rear wheel relaxation length
$\alpha_{fo}, \alpha_{ro}$	Front and rear wheel slip angles that include relaxation length
$v_{wf}, v_{wr}$	Magnitude of front and rear wheel hardpoint velocity
$\mu_f, \mu_r$	Front and rear wheel friction coefficient
$\mu_{fl}, \mu_{fr}$	Front left and right wheel friction coefficient
$\mu_{rl}, \mu_{rr}$	Rear left and right wheel friction coefficient
$C_d$	Air drag coefficient acting along vehicle-fixed x-axis
$C_s$	Air drag coefficient acting along vehicle-fixed y-axis

## Appendix

---

$C_l$	Air drag coefficient acting along vehicle-fixed z-axis
$C_{rm}$	Air drag roll moment acting about the vehicle-fixed x-axis
$C_{pm}$	Air drag pitch moment acting about the vehicle-fixed y-axis
$C_{ym}$	Air drag yaw moment acting about the vehicle-fixed z-axis
$A_f$	Frontal area
$R$	Atmospheric specific gas constant
$T$	Environmental air temperature
$P_{abs}$	Environmental absolute pressure
$w_x, w_y, w_z$	Wind speed, along the vehicle-fixed x-, y-, and z-axes
$W_x, W_y, W_z$	Wind speed, along inertial X-, Y-, and Z-axes

### Ports

#### Input

[expand all](#)

> **whlAngF — Front wheel steering angles**  
scalar | array

> **whlAngR — Rear wheel steering angles**  
scalar | array

> **xdotin — Longitudinal velocity**  
scalar

> **FwF — Total force on the front wheels**  
scalar | array

> **FwR — Total force on the rear wheels**  
scalar | array

> **FExt — External force on vehicle CG**  
array

> **MExt — External moment about vehicle CG**  
array

> **Fh — Hitch force on the body**  
array

> **Mh — Hitch moment about body**  
array

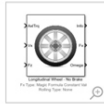
> **windXYZ — Wind speed**  
array

## A.12 Simulink “Longitudinal-Wheel-Block” Documentation [40]

### Longitudinal Wheel

Longitudinal wheel with disc, drum, or mapped brake

Library: Powertrain Blockset / Drivetrain / Wheels  
 Vehicle Dynamics Blockset / Wheels and Tires



#### Description

The Longitudinal Wheel block implements the longitudinal behavior of an ideal wheel. You can specify the longitudinal force and rolling resistance calculation method, and brake type. Use the block in driveline and longitudinal vehicle simulations where low frequency tire-road and braking forces are required to determine vehicle acceleration, braking, and wheel-rolling resistance. For example, you can use the block to determine the torque and power requirements for a specified drive cycle or braking event. The block is not suitable for applications that require combined lateral slip.

There are four types of Longitudinal Wheel blocks. Each block implements a different brake type.

Block Name	Brake Type Setting	Brake Implementation
Longitudinal Wheel - No Brake	None	None
Longitudinal Wheel - Disc Brake	Disc	Brake that converts the brake cylinder pressure into a braking force.
Longitudinal Wheel - Drum Brake	Drum	Simplex drum brake that converts the applied force and brake geometry into a net braking torque.
Longitudinal Wheel - Mapped Brake	Mapped	Lookup table that is a function of the wheel speed and applied brake pressure.

The block models longitudinal force as a function of wheel slip relative to the road surface. To calculate the longitudinal force, specify one of these **Longitudinal Force** parameters.

Setting	Block Implementation
Magic Formula constant value	Magic Formula with constant coefficient for stiffness, shape, peak, and curvature.
Magic Formula pure longitudinal slip	Magic Formula with load-dependent coefficients that implement equations 4.E9 through 4.E18 in <i>Tire and Vehicle Dynamics</i> .
Mapped force	Lookup table that is a function of the normal force and wheel slip ratio.

To calculate the rolling resistance torque, specify one of these **Rolling Resistance** parameters.

Setting	Block Implementation
None	None
Pressure and velocity	Method in <i>Stepwise Coastdown Methodology for Measuring Tire Rolling Resistance</i> . The rolling resistance is a function of tire pressure, normal force, and velocity.
ISO 28580	Method specified in ISO 28580:2018, <i>Passenger car, truck and bus tyre rolling resistance measurement method — Single point test and correlation of measurement results</i> .
Magic Formula	Magic formula equations from 4.E70 in <i>Tire and Vehicle Dynamics</i> . The magic formula is an empirical equation based on fitting coefficients.
Mapped torque	Lookup table that is a function of the normal force and spin axis longitudinal velocity.

To calculate vertical motion, specify one of these **Vertical Motion** parameters.

Setting	Block Implementation
None	Block passes the applied chassis forces directly through to the rolling resistance and longitudinal force calculations.
Mapped stiffness and damping	Vertical motion depends on wheel stiffness and damping. Stiffness is a function of tire sidewall displacement and pressure. Damping is a function of tire sidewall velocity and pressure.

#### Rotational Wheel Dynamics

The block calculates the inertial response of the wheel subject to:

- Axle losses
- Brake and drive torque
- Tire rolling resistance

## Appendix

- Ground contact through the tire-road interface

The input torque is the summation of the applied axle torque, braking torque, and moment arising from the combined tire torque.

$$T_i = T_a - T_b + T_d$$

For the moment arising from the combined tire torque, the block implements tractive wheel forces and rolling resistance with first order dynamics. The rolling resistance has a time constant parameterized in terms of a relaxation length.

$$T_d(s) = \frac{1}{\frac{|\omega| R_e}{L_e} s + 1} (F_x R_e + M_y)$$

To calculate the rolling resistance torque, you can specify one of these **Rolling Resistance** parameters.

Setting	Block Implementation
None	Block sets rolling resistance, $M_y$ , to zero.
Pressure and velocity	Block uses the method in SAE <i>Stepwise Coastdown Methodology for Measuring Tire Rolling Resistance</i> . The rolling resistance is a function of tire pressure, normal force, and velocity. Specifically, $M_y = R_e \{ a + b  V_x  + c V_x^2 \} \{ F_z^\beta p^\alpha \} \tanh(4V_x)$
ISO 28580	Block uses the method specified in ISO 28580:2018, <i>Passenger car, truck and bus tyre rolling resistance measurement method — Single point test and correlation of measurement results</i> . The method accounts for normal load, parasitic loss, and thermal corrections from test conditions. Specifically, $M_y = R_e \left( \frac{F_c C_r}{1 + K_f (T_{amb} - T_{mcos})} - F_{pl} \right) \tanh(\omega)$
Magic Formula	Block calculates the rolling resistance, $M_y$ , using the Magic Formula equations from 4.E70 in <i>Tire and Vehicle Dynamics</i> . The magic formula is an empirical equation based on fitting coefficients.
Mapped torque	For the rolling resistance, $M_y$ , the block uses a lookup table that is a function of the normal force and spin axis longitudinal velocity.

If the brakes are enabled, the block determines the braking locked or unlocked condition based on an idealized dry clutch friction model. Based on the lock-up condition, the block implements these friction and dynamic models.

If	Lock-Up Condition	Friction Model	Dynamic Model
$\omega \neq 0$ or $T_s <  T_i + T_f - \omega b $	Unlocked	$T_f = T_k$ where, $T_k = F_c R_{eff} \mu_k \tanh[4(-\omega_d)]$ $T_s = F_c R_{eff} \mu_s$ $R_{eff} = \frac{2(R_o^3 - R_i^3)}{3(R_o^2 - R_i^2)}$	$\dot{\omega} J = -\omega b + T_i + T_o$
$\omega = 0$ and $T_s \geq  T_i + T_f - \omega b $	Locked	$T_f = T_s$	$\omega = 0$

The equations use these variables.

$\omega$	Wheel angular velocity
$a$	Velocity-independent force component
$b$	Linear velocity force component
$c$	Quadratic velocity force component
$L_e$	Tire relaxation length
$J$	Moment of inertia
$M_y$	Rolling resistance torque
$T_a$	Applied axle torque
$T_b$	Braking torque
$T_d$	Combined tire torque
$T_f$	Frictional torque
$T_i$	Net input torque
$T_k$	Kinetic frictional torque
$T_o$	Net output torque
$T_s$	Static frictional torque
$F_c$	Applied clutch force
$F_x$	Longitudinal force developed by the tire road interface due to slip
$R_{eff}$	Effective clutch radius
$R_o$	Annular disk outer radius

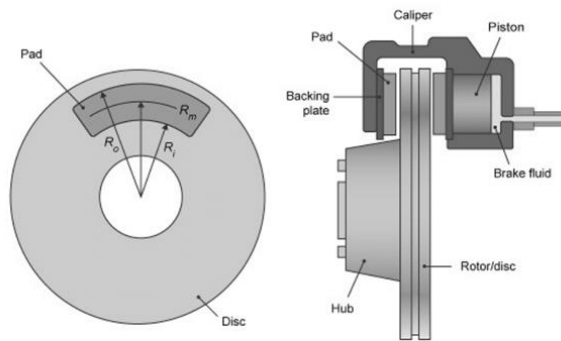
## Appendix

$R_i$	Annular disk inner radius
$R_e$	Effective tire radius while under load and for a given pressure
$V_x$	Longitudinal axle velocity
$F_z$	Vehicle normal force
$C_r$	Rolling resistance constant
$T_{amb}$	Ambient temperature
$T_{meas}$	Measured temperature for rolling resistance constant
$F_{pl}$	Parasitic force loss
$K_t$	Thermal correction factor
$a$	Tire pressure exponent
$\beta$	Normal force exponent
$p_i$	Tire pressure
$\mu_s$	Coefficient of static friction
$\mu_k$	Coefficient of kinetic friction

### Brakes

#### Disc

If you specify the **Brake Type** parameter Disc, the block implements a disc brake. This figure shows the side and front views of a disc brake.



A disc brake converts brake cylinder pressure from the brake cylinder into force. The disc brake applies the force at the brake pad mean radius.

The block uses these equations to calculate brake torque for the disc brake.

$$T = \begin{cases} \frac{\mu P \pi B_a^2 R_m N_{pads}}{4} & \text{when } N \neq 0 \\ \frac{\mu_{static} P \pi B_a^2 R_m N_{pads}}{4} & \text{when } N = 0 \end{cases}$$

$$R_m = \frac{R_o + R_i}{2}$$

The equations use these variables.

$T$	Brake torque
$P$	Applied brake pressure
$N$	Wheel speed
$N_{pads}$	Number of brake pads in disc brake assembly
$\mu_{static}$	Disc pad-rotor coefficient of static friction
$\mu$	Disc pad-rotor coefficient of kinetic friction
$B_a$	Brake actuator bore diameter
$R_m$	Mean radius of brake pad force application on brake rotor
$R_o$	Outer radius of brake pad
$R_i$	Inner radius of brake pad

## Appendix

### Drum

If you specify the **Brake Type** parameter Drum, the block implements a static (steady-state) simplex drum brake. A simplex drum brake consists of a single two-sided hydraulic actuator and two brake shoes. The brake shoes do not share a common hinge pin.

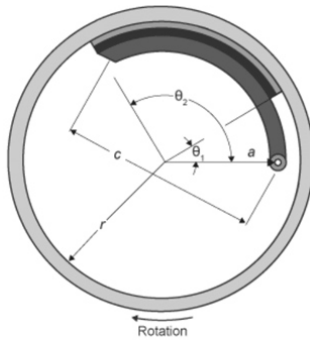
The simplex drum brake model uses the applied force and brake geometry to calculate a net torque for each brake shoe. The drum model assumes that the actuators and shoe geometry are symmetrical for both sides, allowing a single set of geometry and friction parameters to be used for both shoes.

The block implements equations that are derived from these equations in *Fundamentals of Machine Elements*.

$$T_{rshoe} = \left( \frac{\pi \mu c r (\cos \theta_2 - \cos \theta_1) B_a^2}{2\mu(2r(\cos \theta_2 - \cos \theta_1) + a(\cos^2 \theta_2 - \cos^2 \theta_1)) + ar(2\theta_1 - 2\theta_2 + \sin 2\theta_2 - \sin 2\theta_1)} \right) P$$

$$T_{lshoe} = \left( \frac{\pi \mu c r (\cos \theta_2 - \cos \theta_1) B_a^2}{-2\mu(2r(\cos \theta_2 - \cos \theta_1) + a(\cos^2 \theta_2 - \cos^2 \theta_1)) + ar(2\theta_1 - 2\theta_2 + \sin 2\theta_2 - \sin 2\theta_1)} \right) P$$

$$T = \begin{cases} T_{rshoe} + T_{lshoe} & \text{when } N \neq 0 \\ (T_{rshoe} + T_{lshoe}) \frac{\mu_{static}}{\mu} & \text{when } N = 0 \end{cases}$$



The equations use these variables.

$T$	Brake torque
$P$	Applied brake pressure
$N$	Wheel speed
$\mu_{static}$	Disc pad-rotor coefficient of static friction
$\mu$	Disc pad-rotor coefficient of kinetic friction
$T_{rshoe}$	Right shoe brake torque
$T_{lshoe}$	Left shoe brake torque
$a$	Distance from drum center to shoe hinge pin center
$c$	Distance from shoe hinge pin center to brake actuator connection on brake shoe
$r$	Drum internal radius
$B_a$	Brake actuator bore diameter
$\theta_1$	Angle from shoe hinge pin center to start of brake pad material on shoe
$\theta_2$	Angle from shoe hinge pin center to end of brake pad material on shoe

### Mapped

If you specify the **Brake Type** parameter Mapped, the block uses a lookup table to determine the brake torque.

$$T = \begin{cases} f_{brake}(P, N) & \text{when } N \neq 0 \\ \left( \frac{\mu_{static}}{\mu} \right) f_{brake}(P, N) & \text{when } N = 0 \end{cases}$$

The equations use these variables.

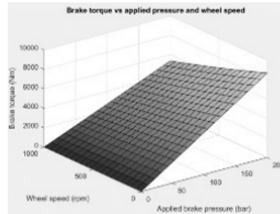
$T$	Brake torque
$f_{brake}(P, N)$	Brake torque lookup table
$P$	Applied brake pressure
$N$	Wheel speed
$\mu_{static}$	Friction coefficient of drum pad-face interface under static conditions

## Appendix

$\mu$	Friction coefficient of disc pad-rotor interface
-------	--

The lookup table for the brake torque,  $f_{brake}(P, N)$ , is a function of applied brake pressure and wheel speed, where:

- $T$  is brake torque, in N-m.
- $P$  is applied brake pressure, in bar.
- $N$  is wheel speed, in rpm.



### Longitudinal Force

To model the Longitudinal Wheel block longitudinal forces, you can use the Magic Formula. The model provides a steady-state *tire characteristic function*  $F_x = f(\kappa, F_z)$ , the longitudinal force  $F_x$  on the tire, based on:

- Vertical load  $F_z$
- Wheel slip  $\kappa$



The Magic Formula model uses these variables.

$\Omega$	Wheel angular velocity
$r_w$	Wheel radius
$V_x$	Wheel hub longitudinal velocity
$r_w \Omega$	Tire tread longitudinal velocity
$V_{sk} = r_w \Omega - V_x$	Wheel slip velocity
$\kappa = V_{sk} /  V_x $	Wheel slip
$F_z, F_{z0}$	Vertical load and nominal vertical load on tire
$F_x = f(\kappa, F_z)$	Longitudinal force exerted on the tire at the contact point. Also a characteristic function $f$ of the tire.

### Magic Formula Constant Value

If you set **Longitudinal Force** to Magic Formula constant value, the block implements the Magic Formula as a specific form of the tire characteristic function, characterized by four dimensionless coefficients ( $B, C, D, E$ ), or stiffness, shape, peak, and curvature:

$$F_x = f(\kappa, F_z) = F_z D \sin(C \tan^{-1} \{ B \kappa - E [ B \kappa - \tan^{-1}(B \kappa) ] \} )$$

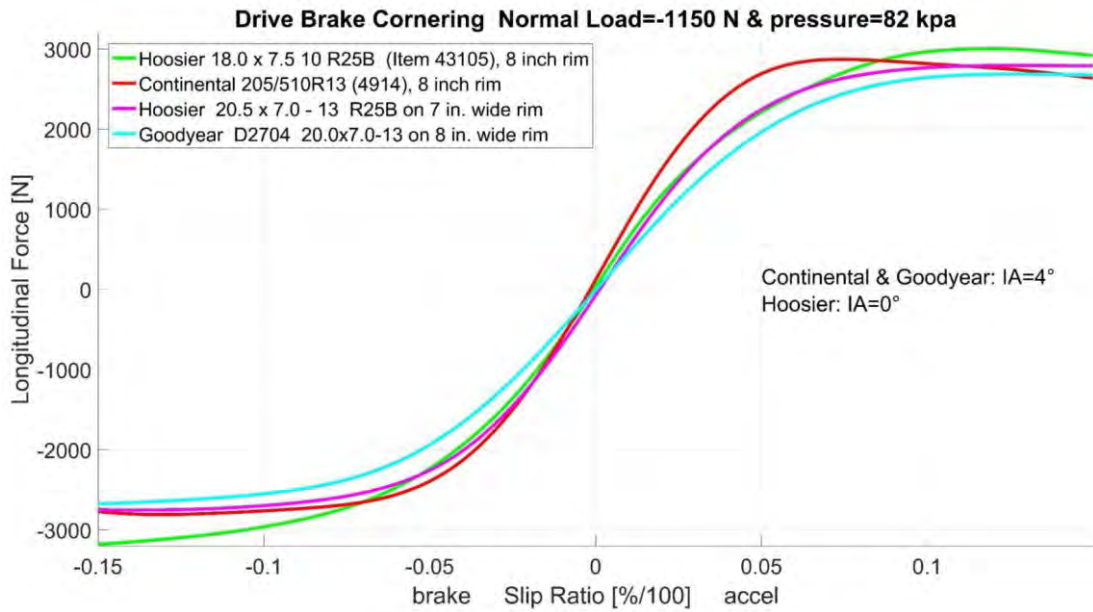
The slope of  $f$  at  $\kappa = 0$  is  $BCD F_z$ .

The coefficients are based on empirical tire data. These values are typical sets of constant Magic Formula coefficients for common road conditions.

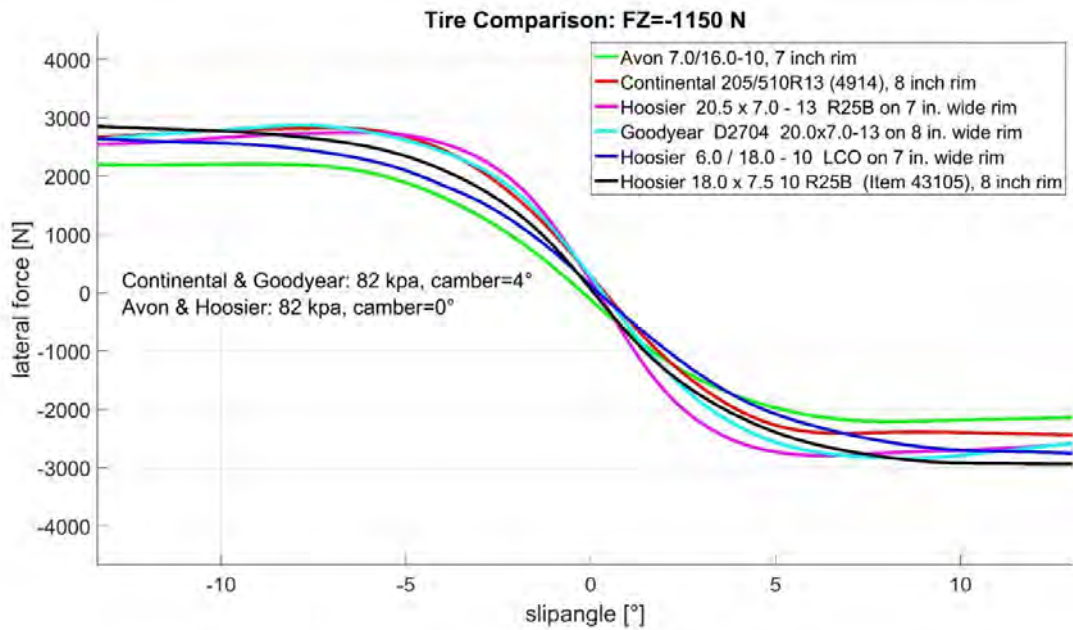
Surface	B	C	D	E
Dry tarmac	10	1.9	1	0.97
Wet tarmac	12	2.3	0.82	1
Snow	5	2	0.3	1
Ice	4	2	0.1	1

### Magic Formula Pure Longitudinal Slip

**A.13 Longitudinal tire data [40]**



**A.14 Lateral tire data [40]**





## A.15 Source code of the function blocks used in the data analyzer

```
Vehicle Control/Data Analyzer/getRearSetpoint* x +
1 function currPoseF = fcn(currPoseC)
2
3 currPoseF = [0 0 0];
4
5 currPoseF(1) = currPoseC(1);
6 currPoseF(2) = (currPoseC(2) + 0.750);
7 currPoseF(3) = round(currPoseC(3));
8
9 end
```

```
Vehicle Control/Data Analyzer/getRefPoint* x +
1 function PointID = fcn(currPoseF, xRef, yRef, numPoints)
2
3 minDist = 100.0;
4 d = 0.0;
5 PointID_temp = 0;
6
7 % Find closest point to vehicle
8 for i = 1:numPoints
9
10 %Xveh Xref Yveh Yref
11 d = sqrt( (currPoseF(1) - xRef(i))^2 + (currPoseF(2) - yRef(i))^2 );
12
13 if ( minDist > d )
14
15 minDist = d;
16 PointID_temp = i;
17 end
18 end
19
20 PointID = PointID_temp;
21
22 end
23
```

```
Vehicle Control/Data Analyzer/getRefPointVel x +
1 function refPointVel = fcn(refVel, pointID)
2
3 refPointVel = refVel(pointID);
4
5 end
6
```

## A.16 Simulation MATLAB Setup Script

```
setupSimulation.m  x +
1  % Initialization
2  clear all;
3  clear;
4  clc;
5  addpath(genpath('Images'));
6
7  % Track
8  load('Acceleration.mat');
9  %load('skidpad.mat');
10 %load('AutoX.mat');
11 %load('AutoCross2019.mat');
12
13 % Define reference points
14 refPose = data.ActorSpecifications.Waypoints;
15 xRef = refPose(:,1);
16 yRef = -refPose(:,2);
17 yawRef = -refPose(:,3);
18 refVel = data.ActorSpecifications.Speed;
19
20 % Define parameters
21 Ts = 12; % simulation time
22 s = size(xRef);
23 tRef = (linspace(0,Ts,s(1)))'; % time variable to plot reference path
24 L = 1.550; % bicycle length
25 X_o = refPose(1,1); % initial vehicleS x-position
26 Y_o = -refPose(1,2); % initial vehicle y-position
27 psi_o = 0; % initial yaw angle
28 numPoints = length(refPose); % Total number of waypoints
29
30 % Controller Parameters
31 Kp = 10;
32 Ki = 0;
33 Kld = 0.25; % lookaheadS distance factor
34
35 % Start Simulation
36 sim('Vehicle_Control_Simulation_H14DV');
```

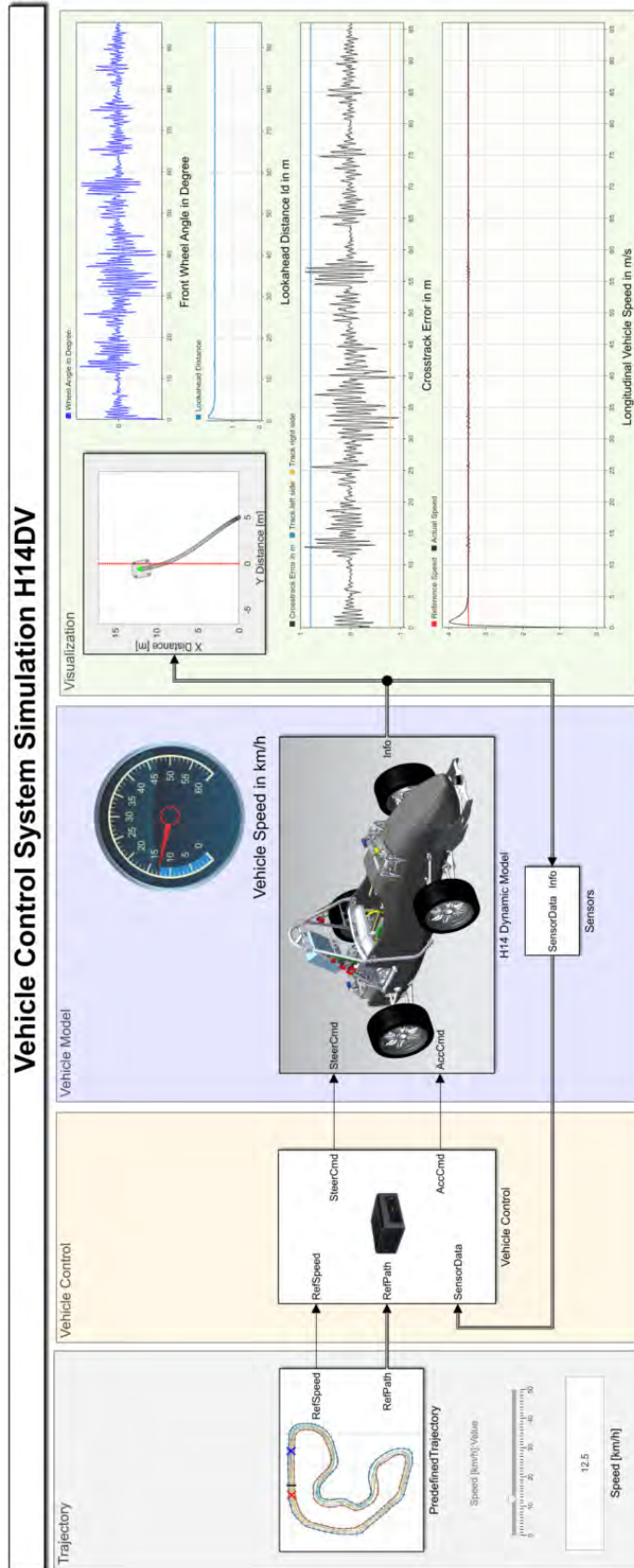
## A.17 MATLAB code used for fitting the longitudinal tire data to the magic formula

```
FittingTireData.m x +
1 % Parameters
2 B = 16;
3 C = 1.65;
4 D = 3.5;
5 E = 0.01;
6 mu = 0.72; %
7 Fz = 1150; %N
8 SR = [-0.1:0.01:.1];
9
10 % Tire Data
11 Data = [-2800 ...% -0.10
12         -2790 ...% -0.09
13         -2770 ...% -0.08
14         -2740 ...% -0.07
15         -2650 ...% -0.06
16         -2470 ...% -0.05
17         -2200 ...% -0.04
18         -1750 ...% -0.03
19         -1190 ...% -0.02
20         -0450 ...% -0.01
21         +0300 ...% +0.00
22         +1020 ...% +0.01
23         +1700 ...% +0.02
24         +2230 ...% +0.03
25         +2550 ...% +0.04
26         +2760 ...% +0.05
27         +2870 ...% +0.06
28         +2900 ...% +0.07
29         +2885 ...% +0.08
30         +2870 ...% +0.09
31         +2845]; % +0.10
32
33 % Magic Formula
34 Fx = D*sin(C*atan(B*SR - E*(B*SR - atan(B*SR))))*mu*Fz;
35
36 % Plotting
37 plot(SR, Fx, 'b', SR, Data(), 'r-o', 'LineWidth',2);
38 legend('Fitted Magic Formula', 'Continental 205/510R13');
39 xlabel('Slip Ratio [%]'), ylabel('Longitudinal Force [N]'), title('Magic Formula Curve Fitting');
40 grid on
```

## A.18 MATLAB code used for fitting the lateral tire data to the linear function

```
LateralTireDataFitting.m x +
1 % Lateral tire forces
2
3 % Parameters
4 Cy = 625;
5 SA = [-10:1:10]; %#ok<NBRAK>
6 b = 150;
7
8 % Tire Data
9 Data = [+2600 ...% -10
10        +2600 ...% -9
11        +2600 ...% -8
12        +2600 ...% -7
13        +2600 ...% -6
14        +2600 ...% -5
15        +2400 ...% -4
16        +2180 ...% -3
17        +1580 ...% -2
18        +1000 ...% -1
19        +0000 ...% +0
20        -0550 ...% +1
21        -1000 ...% +2
22        -1650 ...% +3
23        -2020 ...% +4
24        -2200 ...% +5
25        -2450 ...% +6
26        -2450 ...% +7
27        -2450 ...% +8
28        -2450 ...% +9
29        -2450]; % +10
30
31 % Linear Function
32 Fy = -Cy * SA + b;
33
34 % Plotting
35 plot(SA, Fy, 'b', SA, Data(), 'r-o', 'LineWidth',2);
36 legend('Linear function', 'Continental 205/510R13');
37 xlabel('Slip Angle [°]'), ylabel('Lateral Force [N]'), title('Linear Curve Fitting');
38 axis([-10 10 -3000 3000]);
39 grid on
```

A.19 Simulation Setup



**A.20 Compact disc containing Bachelor thesis as PDF-Format**

# Declaration



Hochschule für Angewandte Wissenschaften Hamburg  
Hamburg University of Applied Sciences

## Erklärung zur selbstständigen Bearbeitung einer Abschlussarbeit

Gemäß der Allgemeinen Prüfungs- und Studienordnung ist zusammen mit der Abschlussarbeit eine schriftliche Erklärung abzugeben, in der der Studierende bestätigt, dass die Abschlussarbeit „– bei einer Gruppenarbeit die entsprechend gekennzeichneten Teile der Arbeit [(§ 18 Abs. 1 APSO-TI-BM bzw. § 21 Abs. 1 APSO-INGI)] – ohne fremde Hilfe selbstständig verfasst und nur die angegebenen Quellen und Hilfsmittel benutzt wurden. Wörtlich oder dem Sinn nach aus anderen Werken entnommene Stellen sind unter Angabe der Quellen kenntlich zu machen.“

Quelle: § 16 Abs. 5 APSO-TI-BM bzw. § 15 Abs. 6 APSO-INGI

Dieses Blatt, mit der folgenden Erklärung, ist nach Fertigstellung der Abschlussarbeit durch den Studierenden auszufüllen und jeweils mit Originalunterschrift als letztes Blatt in das Prüfungsexemplar der Abschlussarbeit einzubinden.

Eine unrichtig abgegebene Erklärung kann -auch nachträglich- zur Ungültigkeit des Studienabschlusses führen.

### Erklärung zur selbstständigen Bearbeitung der Arbeit

Hiermit versichere ich,

Name: Wegner

Vorname: Michel

dass ich die vorliegende Bachelorarbeit bzw. bei einer Gruppenarbeit die entsprechend gekennzeichneten Teile der Arbeit – mit dem Thema:

Entwicklung einer Fahrdynamikregelung für einen fahrerlosen Rennwagen

ohne fremde Hilfe selbstständig verfasst und nur die angegebenen Quellen und Hilfsmittel benutzt habe. Wörtlich oder dem Sinn nach aus anderen Werken entnommene Stellen sind unter Angabe der Quellen kenntlich gemacht.

- die folgende Aussage ist bei Gruppenarbeiten auszufüllen und entfällt bei Einzelarbeiten -

Die Kennzeichnung der von mir erstellten und verantworteten Teile der -bitte auswählen- ist erfolgt durch:

Hamburg

Ort

05.03.2021

Datum

Unterschrift im Original