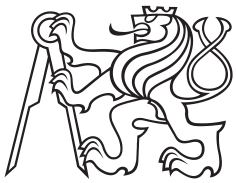
Bachelor's thesis



Czech Technical University in Prague

F3

Faculty of Electrical Engineering
Department of Control Engineering

Over-actuated vehicles path tracking algorithms

Jakub Macar

Supervisor: doc. Ing. Tomáš Haniš, Ph.D.
May 2023

BACHELOR‘S THESIS ASSIGNMENT

I. Personal and study details

Student's name: **Macar Jakub** Personal ID number: **492170**

Faculty / Institute: **Faculty of Electrical Engineering**

Department / Institute: **Department of Control Engineering**

Study program: **Cybernetics and Robotics**

Branch of study: **Common courses**

II. Bachelor’s thesis details

Bachelor’s thesis title in English:

Over-actuated vehicles path tracking algorithms

Bachelor’s thesis title in Czech:

Algoritmy vedení po trati pro vozidla s více stupni volnosti

Guidelines:

The goal of this thesis is to develop path tracking algorithms for vehicles with more degree of freedom for lateral dynamics actuation. The development of electric vehicles and corner modules opens new possibility for path tracking strategies namely, how to utilize extra degree of freedom in vehicle actuation.
1. Get familiar with vehicle dynamic mathematical models and implement suitable one.
2. Get familiar with path tracking algorithms.
3. Develop and implement path tracking strategy for over-actuated vehicle.
4. Evaluate develop algorithm.

Bibliography / sources:

[1] Dieter Schramm, Manfred Hiller, Roberto Bardini – Vehicle Dynamics – Duisburg 2014
[2] Hans B. Pacejka - Tire and Vehicle Dynamics – The Netherlands 2012
[3] Robert Bosch GmbH - Bosch automotive handbook - Plochingen, Germany : Robet Bosch GmbH ; Cambridge, Mass. : Bentley Publishers
[4] Rajamani R. (2012) Mean Value Modeling of SI and Diesel Engines. In: Vehicle Dynamics and Control. Mechanical Engineering Series. Springer, Boston, MA. https://doi.org/10.1007/978-1-4614-1433-9_9

Name and workplace of bachelor’s thesis supervisor:

doc. Ing. Tomáš Haniš, Ph.D. Department of Control Engineering FEE

Name and workplace of second bachelor’s thesis supervisor or consultant:

Date of bachelor’s thesis assignment: **31.01.2023** Deadline for bachelor thesis submission: **26.05.2023**

Assignment valid until:
by the end of summer semester 2023/2024

doc. Ing. Tomáš Haniš, Ph.D.	prof. Ing. Michael Šebek, DrSc.	prof. Mgr. Petr Páta, Ph.D.
Supervisor’s signature	Head of department’s signature	Dean’s signature

III. Assignment receipt

The student acknowledges that the bachelor’s thesis is an individual work. The student must produce his thesis without the assistance of others, with the exception of provided consultations. Within the bachelor’s thesis, the author must state the names of consultants and include a list of references.

. _____ Date of assignment receipt

_____ Student’s signature

Acknowledgements

First of all, I would like to express my gratitude to my supervisor, doc. Ing. Tomáš Haniš, Ph.D., for his expert guidance and patience while working on this thesis.

Second, I would like to thank Ing. Jan Švančar for sharing his knowledge and offering me his help with the validation platform.

Third, I would like to thank all my former tutors at the Faculty of Electrical Engineering of Czech Technical University in Prague. I appreciate the opportunity to study at such a prestigious university and the resources this university provides for students.

Lastly, a very special thanks go to my family and my girlfriend without whose support and encouragement, my studies would not have been possible.

Declaration

I declare that this work is all my own work and I have cited all sources I have used in the bibliography.

Prague, May 26, 2023

Abstract

The emergence of autonomous vehicles in the public transport sector has sparked increasing interest and recognition as a solution in the transportation industry. Consequently, the significance of path-tracking algorithms has escalated, accompanied by heightened demands on these control systems. This thesis aims to present a design of a path-tracking algorithm tailored explicitly for over-actuated vehicles, which deviate from conventional platforms.

To begin, the thesis provides a comprehensive overview of the kinematics and dynamics regarding the motion of over-actuated vehicles, incorporating the derivation of relevant mathematical models. These models serve as the foundation for designing and evaluating the proposed algorithms. Furthermore, the thesis explores the distinctions between reactive algorithms and those incorporating a look-ahead component of the reference path. The primary algorithm proposed is augmented with additional features and subsequently assessed across diverse driving scenarios. Performance and driving comfort comparisons are conducted between the developed algorithms and a Stanley-inspired controller.

Lastly, the thesis concludes with the integration of the algorithm into a validation platform, enabling real hardware platform validation. This integration serves as a pivotal step in assessing the algorithm's functionality.

Keywords: path tracking, over-actuated vehicle, control system design, simulation framework, autonomous vehicle

Supervisor: doc. Ing. Tomáš Haniš, Ph.D.
CTU in Prague
Faculty of Electrical Engineering
Department of Control Engineering,
Resslova 9, E-12, Prague 2

Abstrakt

Rozvoj autonomních vozidel ve veřejné dopravě upoutal pozornost široké veřejnosti a začal být považován za možné řešení v otázce dopravy. V důsledku toho vzrostl význam algoritmů vedení po trati a zároveň se zvýšily požadavky na tyto řídicí systémy. Cílem předkládané práce je návrh algoritmu vedení po trati přímo uzpůsobený pro vozidla s více stupni volnosti, která se odchylují od konvenčních platforem.

Na začátku práce uvádíme souhrnný přehled kinematiky a dynamiky pohybu vozidel s více stupni volnosti včetně odvození příslušných matematických modelů. Dané modely slouží jako podklad pro návrh a vyhodnocení navrhovaných algoritmů. Práce dále zkoumá rozdíly mezi reaktivními algoritmy a algoritmy zahrnujícími dopřednou složku referenční trati. Navržený primární algoritmus je rozšířen o další funkce a následně vyhodnocen v rámci různých jízdních scénářů. Výkon a jízdní komfort jednotlivých navržených algoritmů mezi sebou srovnáváme a následně výsledky. podrobujeme srovnání s regulátorem inspirovaným Stanley řídícím systémem.

V závěru práce algoritmus integrujeme do validační platformy, čímž ověříme algoritmus na reálné hardwarové platformě a podnikneme tak klíčový krok pro posuzování funkčnosti algoritmu.

Klíčová slova: vedení po trati, vozidla s více stupni volnosti, návrh řídícího systému, simulační prostředí, autonomní vozidlo

Překlad názvu: Algoritmy vedení po trati pro vozidla s více stupni volnosti

Contents

Figures

Tables

Chapter 1

Introduction

The topic of this thesis is a model-based design of a path-tracking controller for over-actuated autonomous vehicles. The main focus is to design a control system for several driving scenarios using a hierarchical architecture of the over-actuated vehicle configuration. A Stanley [1] inspired path-tracking controller is used as a reference controller to compare performance and driving comfort data. The designed control system is to be integrated into a real hardware platform used to validate the model-based design.

1.1 Motivation

Autonomous vehicles have emerged as a groundbreaking technology revolutionizing the transportation industry in recent years. As we strive to develop vehicles capable of safely navigating complex road networks, the component of path-tracking algorithms becomes critical. The precision and efficiency of these algorithms directly influence the overall performance and safety of autonomous vehicles.

The development of robust and reliable path-tracking algorithms holds immense potential for reshaping the future of transportation. By effectively addressing the challenges associated with autonomous vehicle control, we can unlock many benefits, including improved traffic flow and reduced accidents.

Many path-tracking algorithms are for vehicles with only front steering axle. The concept of over-actuated vehicles is the ability to steer both the front and rear axle. This ability significantly affects the fundamental dynamics of the vehicles. However, by understanding this modification, it might be possible to unlock new potential in driving autonomous vehicles.

1.2 State of the Art

This section summarizes the path-tracking process and well-known and tested path-tracking algorithms to this date.

1.2.1 Path-Tracking Architecture

The process of path tracking of an autonomous vehicle can be divided into several parts. Firstly, obtaining information about the position and movement of the controlled vehicle is necessary. The vehicle must therefore be equipped with a number of sensors that send their information to the control unit. Often this data must be further filtered and modified.

Secondly, if we want to control a vehicle on a reference path, this path needs to be generated. In robotics and control engineering, a distinction is made between the concept of trajectory and path. A path is a sequence of positions without time parameterization. A trajectory is parameterized by time. Most often, a trajectory is thus specified as a sequence of positions, where a reference velocity or acceleration is also specified for each of them. The design of the reference path is done by a path-planning unit. Many different algorithms are used for real-time planning. For example, algorithms based on tree searches, such as A* or Dijkstra's desribed in [2] and [3] or algorithms based on sampling methods, such as Probalilistic Road Maps (PRM), Rapidly-exploring Random Tress (RRT) [4] or RRT* [5].

The last part is the actual control. From the estimated position of the car and the reference path, the control system calculates a control action. To transfer this control action onto the controlled system, the control system must have a set of actuators available. For vehicle control, these are actuators controlling the steering angles of axles and actuators controlling longitudinal dynamics - speed, acceleration, torque, or the throttle and the brake.

1.2.2 Path-Tracking Algorithms

Stanley Controller

The Stanley controller published in [1] is a nonlinear control law for vehicles to track a reference trajectory. The controller was implemented in a rally car by the Stanford Racing Team. The team competed in the DARPA (Defense Advanced Research Projects Agency) Grand Challenge 2005, tracking a reference trajectory in a desert, so rapidly varying, off-road terrain. The Stanley controller showed a great path-tracking performance, with a typical RMS (root mean square) value of the cross-track error of under 0.1 m. The vehicle with the implemented Stanley controller was the only one that succeeded in the 132-mile desert obstacle course without any significant errors and in the fastest time of 7 hours. This tremendous performance secured the victory for the Stanford team. [6]

The Stanley controller algorithm is a trajectory-tracking algorithm for front wheel steering vehicles. As a trajectory controller, the algorithm has to control both the lateral and longitudinal dynamics of the vehicle. The lateral control can be viewed as a path-tracking control with the reference path determined by a path-planning algorithm. The only control action used by the lateral Stanley controller is the desired front steering angle $d_f(t)$. The algorithm combines the heading and cross-track errors with two damping

members, providing better stability in the vehicle's yaw angle and steering servo without significant performance limitations.

The formula for the desired front steering angle generated by the lateral Stanley controller is as follows:

$$\delta_f(t) = (\psi(t) - \psi_{ss}(t)) + \arctan\left(\frac{k\, e_\perp(t)}{k_{soft} + v(t)}\right)$$

$$+k_{d,yaw}(r_{meas} - r_{traj}) + k_{d,steer}(\delta_{meas}(i) - \delta_{meas}(i+1)) \tag{1.1}$$

Pure Pursuit

The pure pursuit algorithm was originally designed as a method for calculating the arc required for the robot to return to the reference path.

The algorithm falls into the category of algorithms using the look-ahead component of the reference trajectory. Figure 1.1 shows the operation of the algorithm. The vehicle tries to reach a target position TP that is some desired distance L_d ahead of it and lies on the reference path. This point is gradually moved along the reference path to maintain the desired distance between the vehicle and the target position. [7] [8]

The algorithm requires the calculation of the curvature of an arc starting at the center of the rear axle and ending at the target position. The arc has its center in the center of rotation K and radius R. From the curvature values, it is then possible to calculate the value of the control action. In the figure 1.1, the control action is the front steering angle δ_f. According to the sine theorem:

$$\frac{L_d}{\sin 2\alpha} = \frac{R}{\sin\left(\frac{\pi}{2} - \alpha\right)} \tag{1.2}$$

And from geometry:

$$R = \frac{L}{\tan\left(\delta_f\right)} \tag{1.3}$$

From 1.2, 1.3 it is possible to calculate the desired front steering angle for the Pure Pursuit algorithm:

$$\delta_f = \arctan\left(\frac{2L\sin\alpha}{L_d}\right) \tag{1.4}$$

1.3 Problem Definition

The main objective of this work is to design a controller to control the lateral dynamics of an over-actuated vehicle. The issue of over-actuated vehicles offers new opportunities to develop control algorithms. Solving the possibility of multiple actuators will be one of the main points of the design. A simple cruise control-based longitudinal dynamics controller will also be designed in this thesis.

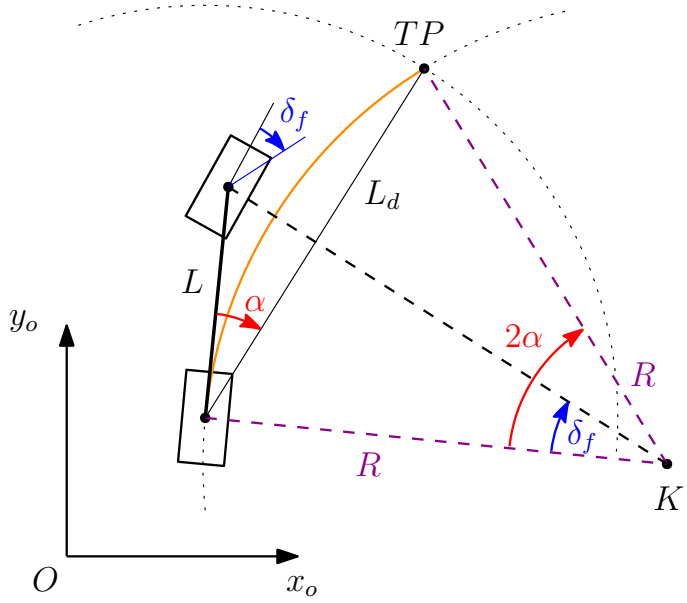


Figure 1.1: The Pure Pursuit algorithm demonstrated on a front wheel steering vehicle.

To design the controller and test it, it is necessary to implement a mathematical model of the vehicle that will sufficiently describe the vehicle dynamics. The mathematical model will be implemented in a simulation framework that will be used for the development and validation of the control algorithms and its creation will be part of this thesis. The commonly used principle of decoupling the lateral and longitudinal vehicle dynamics will be used for the development of the control system.

The proposed algorithms will then be implemented in the validation model and their functionality will thus be tested on real hardware.

The mathematical models and experiments will be represented in the coordinate system O. This is a right-handed coordinate system with two axes x_o, y_o in the direction of two basis vectors of Euclidean space of dimension two. [9]

Chapter 2

Vehicle Modelling

Accurate vehicle modelling is a crucial foundation for developing effective control systems in the rapidly evolving field of autonomous vehicles. This chapter provides an overview of vehicle modelling, focusing on understanding the kinematic and dynamic behavior of autonomous vehicles.

By capturing the essential characteristics of vehicle motion, this modelling framework enables us to design and optimize control algorithms for achieving precise and safe autonomous driving. In this chapter, we explore the fundamental principles and mathematical representations used in vehicle modelling, laying the groundwork for subsequent chapters about control strategies and path-planning algorithms.

2.1 Bicycle Models

In this section, the idea of bicycle modelling will be explained. The models presented in the following sections are in the bicycle model category, so the same characteristics will apply. From a kinematic point of view, a vehicle with four wheels and front wheel steering has a condition between the inner and outer wheel steering angle for turning slip-free. As shown in figure 2.1, the steering angle of the wheels differs because the radius to the center of rotation K is different.

This can be described by the Ackermann formula

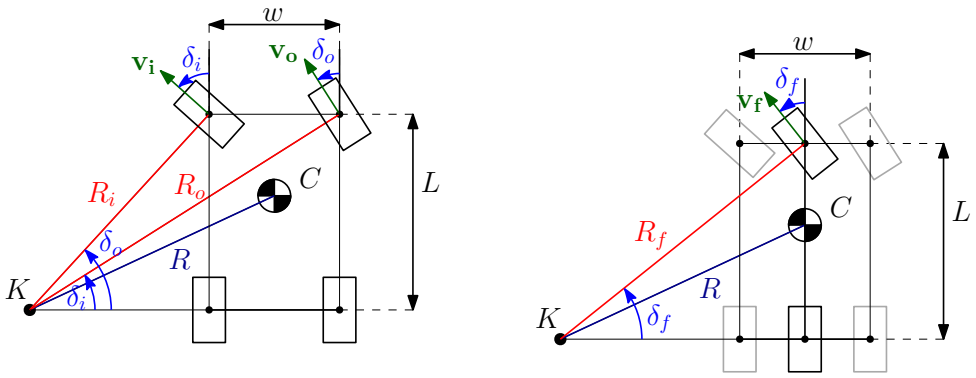
$$\cos \delta_o - \cos \delta_i = \frac{w}{L}, \tag{2.1}$$

where δ_o is the steering angle of the outer wheel, δ_i is the steering angle of the inner wheel, w is track width and L is the wheelbase. [10]

The bicycle model neglects this fact; the axle is represented by a single wheel, thus does not allow explicit modelling of Ackerman steering.

The model, however, can be extended to include Ackerman steering by the following procedure. The vehicle's center of gravity (c.g.) C must be turned on a circle with the same radius R. From geometry, the following formula can be obtained for equivalent front steering angle δ_f:

$$\cot \delta_f = \frac{\cot \delta_o + \cot \delta_i}{2}, \tag{2.2}$$



(a) : A front wheel steeering vehicle and steering angles of the inner and outer wheel.

(b) : Equivalent bicycle model for a front wheel steering vehicle.

Figure 2.1: Ackermann geometry.

where δ_f is the equivalent steering angle of a bicycle model having the same radius of rotation R for the same wheelbase L. From the equations 2.1, 2.2 the formulas for the Ackermann steering can be computed:

$$\delta_o = \cot^{-1}(\cot \delta_f + \frac{w}{2L}) \tag{2.3}$$

$$\delta_i = \cot^{-1}(\cot \delta_f - \frac{w}{2L}) \tag{2.4}$$

An approximation method of the equation 2.2 can be used. [11] The parameter δ_f is calculated as the arithmetic average of the steering angles δ_i and δ_o of the two wheels:

$$\delta_f \approx \frac{\delta_o + \delta_i}{2} \tag{2.5}$$

This small angle approximation will be regarded as a linearization of the Ackermann steering geometry. For δ_f values in the range of 0 and 30 degrees, we can observe the difference between the two calculation methods on the graph 2.2. We can observe that the linearized method deviates from the Ackermann geometry with increasing steering angle. Still, the difference between the two methods does not exceed 2 degrees in the chosen steering range for any of the used wheelbase-to-track ratios. Five different wheelbase-to-track ratios were selected, as these ratios represent commonly used geometry in road cars. [12] [13] [14]

The Ackermann steering is commonly used at lower speeds. With increasing speed, parallel or reverse steering is used, especially in racing vehicles. [10] Research from [15] that studied the steering geometry on the performance of a race car concluded, however, that there is a very minimal difference in performance between the parallel and Ackermann steering methods.

With bicycle models, we do not have the possibility to observe the interaction between the two wheels on the same axle. For stability and handling analysis, it is appropriate to use the twin-track model presented in [16]. However, for the purposes of this thesis, the twin-track model will not be relevant and thus will not be further discussed.

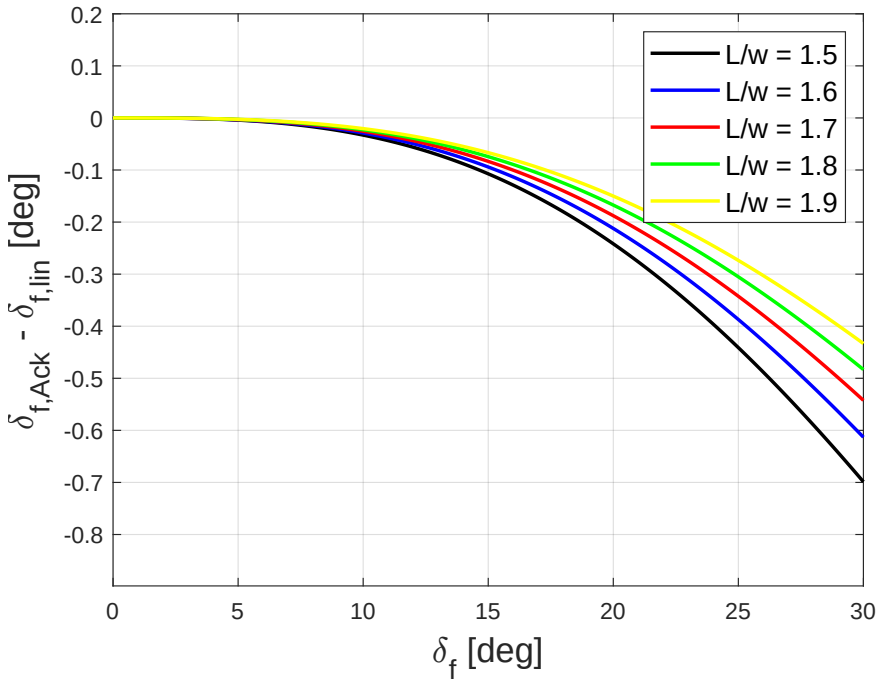


Figure 2.2: The difference between the Ackermann geometry and linearized version for $\delta_f \in \langle 0; 30 \rangle$ [deg] for different wheelbase-to-track ratios.

2.2 Kinematic Model

In this section, the Kinematic Model of a vehicle will be presented. This model provides mathematical equations to describe the motion of a vehicle without considering the forces that generate the motion. The equations are derived only from the geometry of the vehicle. The main advantage of using a kinematic model to model vehicle motion is its relative simplicity.

The kinematic model of a vehicle is a simplified representation of a vehicle's motion, which assumes certain constraints and idealizations to simplify the equations of motion. It is a bicycle model (the characteristics are discussed in the section 2.1), so the model represents the two left and right wheels as one wheel at the center of the axle.

The critical assumption is that the velocity vectors $\mathbf{v_f}$, $\mathbf{v_r}$ are aligned with the orientation of the front and rear wheels, respectively, as shown in figure 2.3. This assumption states that there is zero lateral slip on both wheels as the vehicle moves. The total lateral force generated by the two tires when a vehicle of mass m moves along a circular path of radius R is

$$F_y = \frac{mv_c^2}{R}. \tag{2.6}$$

As we can see, this assumption is reasonable for low speeds and smaller steering angles. The lateral force increases quadratically with increasing speed of motion. On the contrary, it is inversely proportional to the vehicle's turning radius. For faster and more aggressive maneuvers, this model is no

longer accurate. However, it can serve as a good validation tool or a model for parking control design.

The vehicle is assumed rigid and has no deformations or vibrations. The vehicle's mass is concentrated to one point - the c.g.

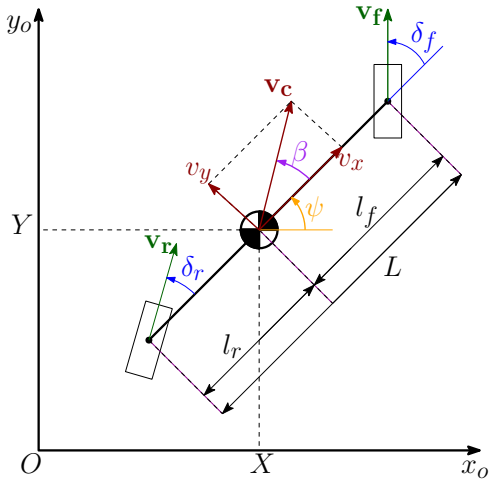


Figure 2.3: Kinematic model in coordinate system O.

The motion of the vehicle is modeled in the plane; the third-height coordinate is neglected. The vehicle model is equivalent to a model of a line segment moving in the plane, hence a model with three degrees of freedom. The vehicle is, therefore, sufficiently described by three coordinates X,Y and ψ in the global coordinate system O. Coordinates X and Y are the position of the center of gravity and ψ the vehicle's orientation - the yaw angle.

The equations of the kinematic model are as follows:

$$\dot{X} = v_c \cos(\psi + \beta), \tag{2.7}$$

$$\dot{Y} = v_c \sin(\psi + \beta), \tag{2.8}$$

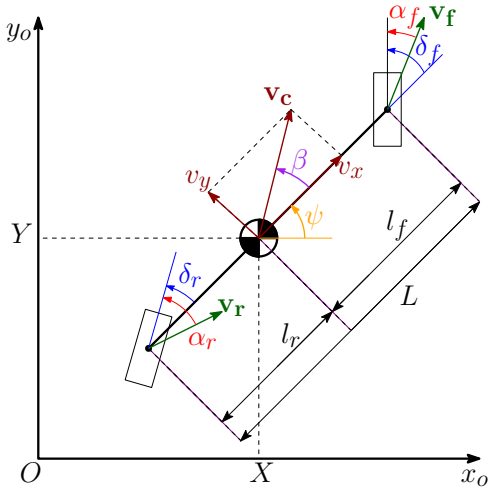
$$\dot{\psi} = \frac{v_c \cos(\beta)}{l_f + l_r} \left(\tan(\delta_f) - \tan(\delta_r)\right), \tag{2.9}$$

$$\beta = \tan^{-1}\left(\frac{l_f \tan(\delta_r) + l_r \tan(\delta_f)}{l_f + l_r}\right), \tag{2.10}$$

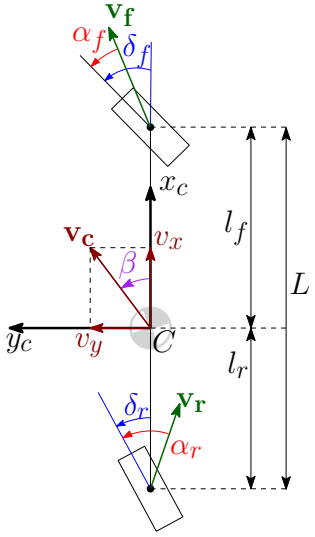
where X and Y are x and y-coordinate in the global coordinate system O, v_c is the magnitude of the velocity vector in the c.g. of the vehicle, ψ is the yaw angle of the vehicle in the global coordinate system O, $\dot{\psi}$ vehicle's yaw rate, β is the vehicle side slip angle, l_f and l_r are the distance from the c.g. to the center of the front and rear axle, respectively, $L = l_f + l_r$ is the wheelbase of the vehicle and δ_f and δ_f are the steering angles of the front and rear wheels, respectively. [11]

2.3 Linear Single-Track Model

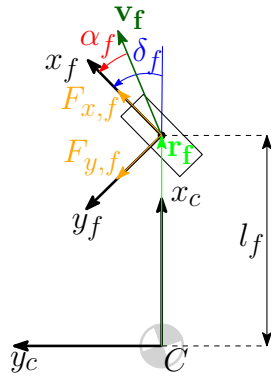
In this section, the linear single-track model of a vehicle will be presented. For higher velocities, it cannot be assumed that the velocity vectors on each wheel point in the same direction as the wheels are turned. It is necessary to model the slips on the tires. The single-track model is a dynamic model, so unlike the kinematic model, it studies the forces acting on the vehicle that cause the motion.



(a) : Single-track model model in coordinate system O.



(b) : Single-track model body fixed.



(c) : Single-track model tire coordinates.

Figure 2.4: Single-track Model model in in different coordinate systems.

This thesis will not focus on the nonlinear single-track model; for more information, please refer to [16]. The linear model provides a relatively good approximation of the lateral dynamics of the vehicle. However, a certain number of simplifications are applied:

- The speed at the c.g. of a vehicle is considered constant throughout the vehicle's movement, and the longitudinal forces on the tires are not considered.
- It is a planar model; the model does not consider the height coordinate, and all lifting, rolling and pitching motions are not considered.
- It is a bicycle model, so the model represents the two left and right wheels as one wheel at the center of the axle.
- We do not consider the pneumatic trail and the aligning torque resulting from the slip angle.
- The load on the vehicle is evenly distributed between the front and rear axles.

[16]

2.3.1 Model of Tires

The dynamic model considers the forces acting on the vehicle motion; therefore, a tire model is necessary, as the tires are the only point of contact between the vehicle and the road and thus transmit the forces that cause the vehicle to move. In a nonlinear single-track model, Pacejka's *Magic Formula* Tire Model is used. [17] The formula provides a relationship between the forces acting on the tires and the slip ratio. In the linear single-track model, the longitudinal forces on tires are not considered. For small tire angles, the relationship between lateral forces acting on the tires and the slip ratio can be assumed linear:

$$F_y, f = C_f \alpha_f, \tag{2.11}$$

$$F_y, r = C_r \alpha_r, \tag{2.12}$$

where the coefficients C_f and C_r represent the cornering stiffness of the front and rear tire, respectively. [11] [16]

2.3.2 Center of Gravity to Center of the Axle Transformation

To determine the tire slip angles, the transformation of the velocity vector from the c.g. to the center of the axle is needed.

$$\mathbf{v_f} = \mathbf{v_c} + \omega \times \mathbf{r} = \begin{bmatrix} \|\mathbf{v_c}\| \cos(\beta) \\ \|\mathbf{v_c}\| \sin(\beta) \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ \dot{\psi} \end{bmatrix} \times \begin{bmatrix} l_f \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} \|\mathbf{v_c}\| \cos(\beta) \\ \|\mathbf{v_c}\| \sin(\beta) + l_f \dot{\psi} \\ 0 \end{bmatrix}, \tag{2.13}$$

$$\mathbf{v_r} = \mathbf{v_c} + \omega \times \mathbf{r} = \begin{bmatrix} \|\mathbf{v_c}\| \cos(\beta) \\ \|\mathbf{v_c}\| \sin(\beta) \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ \dot{\psi} \end{bmatrix} \times \begin{bmatrix} -l_r \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} \|\mathbf{v_c}\| \cos(\beta) \\ \|\mathbf{v_c}\| \sin(\beta) - l_r \dot{\psi} \\ 0 \end{bmatrix}, \tag{2.14}$$

where $\mathbf{v_c}$ is the velocity vector of the c.g. in the body-fixed coordinates, ω is the vector of angular velocities in the body-fixed coordinates, $\mathbf{r}$ is the position vector of the center of the axle in respect of the position of the c.g. and $\mathbf{v_f}$ and $\mathbf{v_r}$ are the velocity vectors in the tire contact point.

From the figure 2.4a, vector $\mathbf{v_f}$ can be also written as follows:

$$\mathbf{v_f} = \begin{bmatrix} \|\mathbf{v_f}\| \cos(\delta_f - \alpha_f) \\ \|\mathbf{v_f}\| \sin(\delta_f - \alpha_f) \\ 0 \end{bmatrix}. \tag{2.15}$$

From the equations 2.13, 2.14, 2.15 and with the use of small angle approximation:

$$\frac{\sin(\delta_f - \alpha_f)}{\cos(\delta_f - \alpha_f)} = \frac{\|\mathbf{v_c}\| \sin(\beta) + l_f \dot{\psi}}{\|\mathbf{v_c}\| \cos(\beta)} \approx \beta + l_f \frac{\dot{\psi}}{\|\mathbf{v_c}\|} \tag{2.16}$$

$$\rightarrow \alpha_f = \delta_f - \beta - l_f \frac{\dot{\psi}}{\|\mathbf{v_c}\|}. \tag{2.17}$$

With an analogous procedure, it is possible to obtain the relationship for the rear axle:

$$\alpha_r = \delta_r - \beta + l_r \frac{\dot{\psi}}{\|\mathbf{v_c}\|}. \tag{2.18}$$

[16]

2.3.3 State-Space Representation of the Linear Single-Track Model

Due to the many assumptions described in section 2.3, the motion of the vehicle can be described by the yaw rate $\dot{\psi}$ and side slip angle β. As a result, these are the two state variables of the state-space model of the linear single-track model.

The state-space model of the linear single-track model is as follows:

$$\dot{\beta} = -\frac{C_f + C_r}{m v_c}\beta + \left(\frac{l_r C_r - l_f C_f}{m v_c^2} - 1\right)\dot{\psi} + \frac{C_f}{m v_c}\delta_f + \frac{C_r}{m v_c}\delta_r, \tag{2.19}$$

$$\ddot{\psi} = \frac{l_r C_r - l_f C_f}{I_z}\beta - \frac{l_f^2 C_f + l_r^2 C_r}{v_c I_z}\dot{\psi} + \frac{l_f C_f}{I_z}\delta_f - \frac{l_r C_r}{I_z}\delta_r, \tag{2.20}$$

where v_c is the magnitude of the velocity vector of the c.g., i.e., $v_c = \|\mathbf{v_c}\|$, C_f [C_r] is the cornering stiffnesses of the front [rear] tire, l_f [l_r] is the distance from the c.g. to the center of the front [rear] axle, I_z is the moment of inertia of the vehicle, m is the mass of the vehicle, δ_f and δ_r are the steering angles of the wheels and inputs of this state-space model. [16]

2.4 Model Selection and Identification

For the simulation experiments and lateral controller design, the linear single-track model was selected. The model captures the essential characteristics of a vehicle's lateral dynamics.

Due to its relative simplicity and computational efficiency, the model is a good simulation tool. The linear single-track model has a limited number of parameters that can be identified from experimental data. This simplifies the process of parameter identification, which is essential for designing control systems and conducting vehicle simulations. Due to its relative simplicity and linearity, the model provides good properties for simulation and control system design.

As mentioned before, the model has several parameters whose value needs to be identified. The parameters of the model serve as a specification for the model to resemble the real hardware validation platform as much as possible, as this platform is the final evaluation of the designed control system. The identification process will not be discussed in this thesis. For the identification experiments and results, please refer to [18].

The parameters of the linear single-track model and their identified values are listed in the table 2.1.

Parameters of linear single-track model		
C_f [-]	53.3964	Cornering stiffness of the front tire
C_r [-]	68.8640	Cornering stiffness of the rear tire
l_f [m]	0.3	Distance between c.g. and the center of the front axle
l_r [m]	0.3	Distance between c.g. and the center of the rear axle
m [kg]	21	Mass of the vehicle
I_z [kg $\cdot$ m^2]	1.2562	Moment of inertia of the vehicle

Table 2.1: Parameters of linear single-track model

Chapter 3

Path-Tracking Architecture Design

This chapter will present the path-tracking architecture. For the control system design, lateral and longitudinal dynamics are decoupled. In section 3.1, this decoupling concept is summarized.

For the purposes of this thesis, a simulation framework was created as a primary validation tool for the designed control systems. In section 3.2, the structure of the framework and its components will be described.

3.1 Decoupling of Lateral and Longitudinal Dynamics

This concept involves separating the modelling and control of lateral (side-to-side) and longitudinal (forward-backward) motion, allowing for independent design and performance optimization. The decoupling of lateral and longitudinal concept is commonly used for the modelling and control design of vehicle. [11] [1] [16] [19]

This separation of dynamics can be done due to several assumptions. The extraction of lateral dynamics was shown in section 2.3; the linear single-track model covers the lateral dynamics of a vehicle. Since the topic of this thesis is path-tracking algorithms, not trajectory-tracking algorithms, there is no reference time for each section of the reference path. This means there is no information about the reference velocity; thus, the velocity can be regulated externally. The primary focus of the control design is the design of the lateral dynamics controller. The longitudinal dynamics controller is designed separately, while the information about reference velocity is externally given and not connected to the reference path. By using this method, the feasibility of the reference maneuver, however, is not guaranteed.

3.2 Simulation Framework

This chapter will introduce the simulation framework used for performing the simulation experiments and data evaluation. This framework was developed in Matlab version R2022b, a software from MathWorks company, and its toolbox Simulink.

3.2.1 Framework Architecture

In this section, the simulation framework architecture is presented. The scheme of architecture is pictured in figure 3.1. The architecture uses the decoupling concept from section 3.1. There are the following function blocks:

- Model of lateral vehicle's dynamics; the block output is a vector $\mathbf{p}$ which containts the vehicle position in the coordinate system O and other variables mentioned in section 2.3.
- Model of longitudinal vehicle's dynamics. 4.2
- Designed lateral controller; the block's output is the desired front and rear steering angles, δ_f and δ_r. 5
- Designed longitudinal controller; the block's output is the desired torque τ. 4
- Reference generator block; the block's output is the reference vector $\mathbf{r}$, the definition of this vector is in section 3.2.3.
- Error calculation block; the block's output is the path-tracking error vector $\mathbf{e}$.

The vector $\mathbf{r_0}$ describes the initial conditions of the vehicle:

$$\mathbf{r_0} = \begin{bmatrix} x_0 \\ y_0 \\ \psi_0 \\ v_0 \end{bmatrix},$$

where x_0, y_0, ψ_0 are the initial coordinates in the coordinate system O and v_0 is the initial value of the vehicle's velocity in the c.g. The vector $\mathbf{e}$ contains the path-tracking errors:

$$\mathbf{e} = \begin{bmatrix} e_\perp \\ e_{\psi,C} \\ e_{\psi,L} \end{bmatrix},$$

where $e_\perp$ is the cross-track error and $e_{\psi,C}$ and $e_{\psi,L}$ are the current and look-ahead heading error. For the definition of these path-tracking errors, please refer to section 5.1.

3.2.2 Path Generation

This section will describe generating a vehicle's reference path in the simulation framework. The reference path $\mathbf{P}$ has the structure of a matrix of dimension $2 \times N$, where N is the number of points in the path. Each column j of this matrix consists of a pair (x_j, y_j), i.e., one point in the plane. The sequence of these pairs forms the reference path. The sequence of points is not parameterized by time, so it is not a reference trajectory. However, the

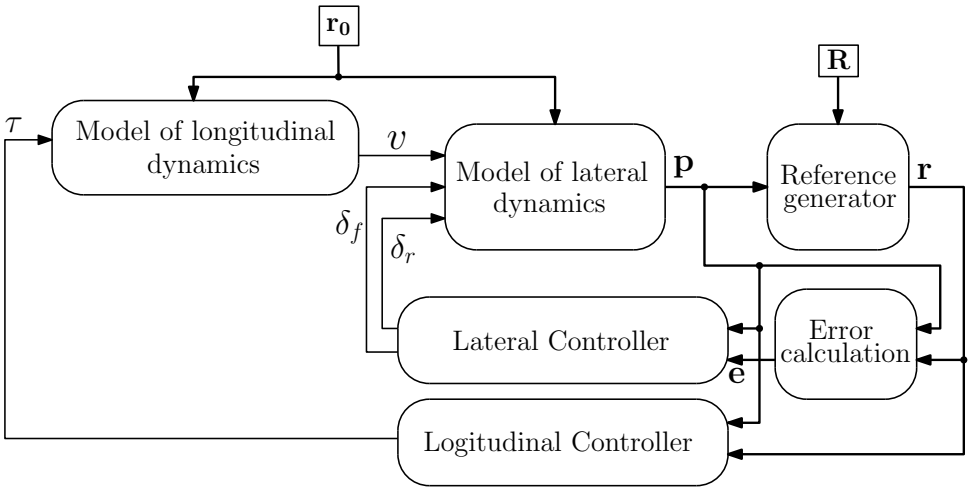
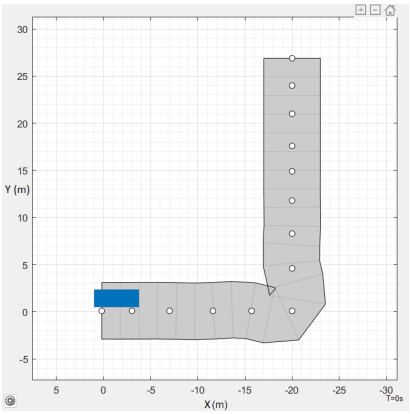


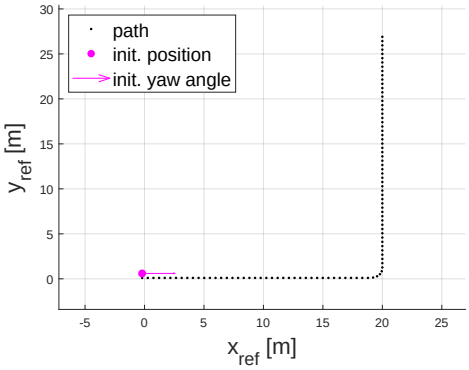
Figure 3.1: FrameworkArch.

order of the points corresponds to the order in which the points should be passed.

Matlab's application Driving Scenario Designer was used to design the given sequence of points. The application provides a convenient user interface and path visualization, as shown in figure 3.2a. A path is created by inserting several path centers, which can then be exported to a *.mat* file. The subsequent sequence of path centers was linearly interpolated to increase the density of points to at least three points per metre of the path.



(a) : Driving Scenario desginer.



(b) : Linear interpolation of the road centers.

Figure 3.2: Path Generation.

3.2.3 Reference Generation

This section will describe how to generate a reference in the Simulation Framework. The reference vector $\mathbf{r}$ will have the following form:

$$\mathbf{r} = \begin{bmatrix} x_{ref} \\ y_{ref} \\ \psi_{ref} \\ v_{ref} \end{bmatrix},$$

where x_{ref}, y_{ref}, ψ_{ref} are the reference coordinates in the coordinate system O and v_{ref} is the value of the reference velocity of the vehicle.

First, it is necessary to extend the reference path $\mathbf{P}$ to a reference matrix $\mathbf{R}$, which for each pair (x_j, y_j) will also contain the reference yaw angle and the reference velocity. The generation of the reference path is described in section 3.2.2. The reference yaw angle can be obtain from the (x_j, y_j) pairs by following formula:

$$\psi_j = \arctan\left(\frac{y_{j+1} - y_j}{x_{j+1} - x_j}\right). \tag{3.1}$$

The reference vector $\mathbf{r}$ is the column of this matrix that best corresponds to the position of the vehicle center, i.e., $\mathbf{r}$ corresponds to the jth-column of the matrix $\mathbf{R}$ whose components (x_j, y_j) have the smallest Euclidean distance [9] of all the matrix's components from the position of the center of mass of the vehicle (x_C, y_C):

$$\arg\min_j \sqrt{(x_C - x_j)^2 + (y_C - y_j)^2}, \quad j \in \langle 1, N \rangle. \tag{3.2}$$

For additional look-ahead components, the look-ahead yaw angle $\psi_{ref,L}$ can be included in the reference matrix and reference vector. Then the look-ahead yaw angle corresponds $(j + L)$th-column of the matrix $\mathbf{R}$, where L is the look-ahead parameter. The parameter represents how many points forward in the reference path is the look-ahead component taken. L depends on the current vehicle speed and the density of points in the reference path:

$$L = \mathrm{RNI}\left(\frac{C\, v_x}{d}\right),$$

where v_x is the longitudinal velocity of the vehicle, d is the distance between two consecutive points in metres, C is a constant heuristically set to $C = 1.3$ and RNI stands for round to nearest integer function.

With the look-ahead component, the reference vector has following form:

$$\mathbf{r} = \begin{bmatrix} x_{ref} \\ y_{ref} \\ \psi_{ref,C} \\ \psi_{ref,L} \\ v_{ref} \end{bmatrix},$$

where $\psi_{ref,C}$ is the current reference yaw angle and $\psi_{ref,L}$ is the look-ahead reference yaw angle.

Chapter 4

Longitudinal Controller

A longitudinal controller is a system that controls the longitudinal dynamics of a vehicle. There can be a variety of controlled variables, for example, velocity or acceleration. The control inputs in the vehicle are the throttle and the brakes, which indicate the torque on the wheels.

The purpose of this longitudinal controller is to control the control input to achieve the reference velocity. That is the standard cruise control system. [11] There is, however, also the request to control the maximal acceleration of the vehicle. In the simulation framework, the control input is the torque on the wheels.

4.1 Controller Objectives

The controller is used to generate sufficient torque to allow the vehicle to reach a reference velocity. During abrupt changes in reference velocity, the controller must provide a gradual increase in the vehicle's velocity. It is required that the step response of the vehicle's velocity is to be without overshoot.

4.2 Model of Logitudinal Dynamics

A very simplified model based on Newton's second law for rotation was used to model the longitudinal dynamics of the vehicle

$$\tau = I_w \dot{\omega_w}, \tag{4.1}$$

where τ is a external torque, I_w is the moment of interia of the wheels and ω_w is the angular velocity of the wheels. [20]

From the equation 4.1, it is possible to obtain the dependence between the torque and the vehicle's acceleration a.

$$a = C_w \tau, \tag{4.2}$$

where the constant C_w depends on the radius of the wheels and the moment of inertia of the wheels. The model does not consider longitudinal tire slip, tire rolling resistance, or wheel servo dynamics.

4.3 Controller Architecture

The architecture of the longitudinal controller is shown in figure 4.1. It is a hierarchical architecture with two feedback loops and a velocity and acceleration controller. [11]

The upper-level velocity controller C_v determines the desired longitudinal acceleration of the vehicle from the deviation of the vehicle's velocity from the reference velocity. The maximal level of reference acceleration a_{ref} is limited by the saturation block in the controller's output, with the values $\pm a_{max}$.

The lower-level acceleration controller C_a determines the level of torque input to achieve the desired reference acceleration. The mechanical saturation of the torque input is simulated by the saturation block in the controller's output, with the values $\pm \tau_{max}$.

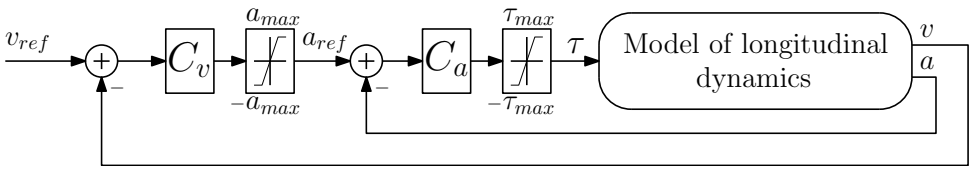


Figure 4.1: Architecture of the longitudinal controller.

Chapter 5

Lateral Controller

A lateral controller is a system that controls the lateral dynamics of a vehicle. The controller design is a model-based design based on the linear single-track model from section 2.3.

The main objective is to control the path-tracking errors throughout the vehicle's motion. The performance aspect is judged by the maximal and root mean square values of these errors. However, driving comfort will be included in the overall evaluation of the designed control system. Therefore, the following procedure will be used in the design:

- Determination of the Lateral Controller architecture.
- Damping of unwanted modes of the system.
- Determination of controller performance evaluation.
- Design of the controller for the selected constant vehicle velocity.
- Testing and adjustment of controllers for velocity in the working range $v_c \in \langle 1, 10 \rangle$ $[\mathrm{m} \cdot \mathrm{s}^{-1}]$.

5.1 Path-Tracking Errors

The path-tracking errors are the fundamental indicators of the driving performance of the designed control system. This section will describe how the positional and orientational errors with respect to the reference path are computed. These errors are a key part of the lateral dynamics control system design, as they serve as reference signals for the lateral controller.

5.1.1 Cross-Track Error

The cross-track error $e_\perp$ represents the signed perpendicular distance of the c.g. of the vehicle from the reference path. The sign represents the side (left or right) on which the vehicle is located relative to the reference path. Its computation is pictured in figure 5.1a. Waypoints $\mathbf{P_1}$,$\mathbf{P_2}$ are the points in the reference path with the smallest Euclidean distance from the vehicle's

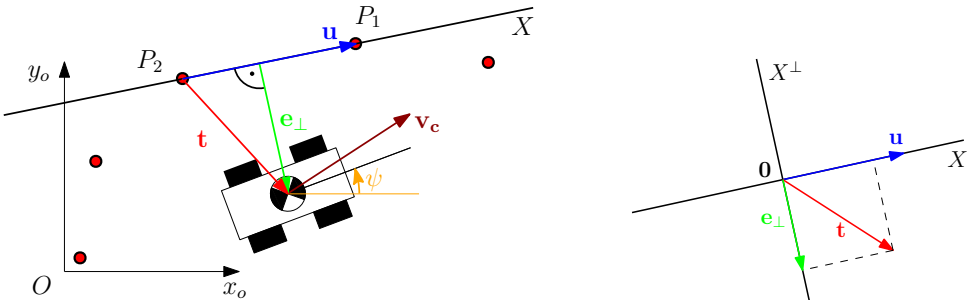
c.g. If we intersect these two points with a line, the cross-track error is the distance of c.g. from this line.

We can compute the vectors $\mathbf{u}$, $\mathbf{t}$ and $\mathbf{e}_\perp$ as shown in the figure. In figure 5.1b, these vectors are moved to the origin.

The vector $\mathbf{u}$ is the base of the linear subspace X, i.e., $X = \text{rng}(\mathbf{u})$. The vector $\mathbf{e}_\perp$ is the ortogonal projection of vector $\mathbf{t}$ on the linear subspace $X^\perp$, where $X^\perp$ is the complement of the subspace X. The orthonormal base of the subspace $X^\perp$ is the vector $\frac{\mathbf{u}^\perp}{\|\mathbf{u}^\perp\|}$, where $\mathbf{u}^\perp = (u_2, -u_1)$. Then the cross-track error $e_\perp$ is the coefficient of the vector $\mathbf{e}_\perp$ in this orthonormal base. [9]

With that, the [9] presents the following formula for the calculation of the cross-track error:

$$e_\perp = \frac{\left(\mathbf{u}^\perp\right)^T}{\|\mathbf{u}^\perp\|}\mathbf{t}. \tag{5.1}$$



(a) : Geometrical meaning of the cross-track error.

(b) : Vector translation to origin.

Figure 5.1: The cross-track error.

5.1.2 Heading Error

The heading error is a deviation between the vehicle's yaw angle and a reference yaw angle. Two heading errors are used in the architecture of the lateral controller. The difference between them is the different references that are being used for calculation. The current heading error $e_{\psi,C}$ describes the current deviation in the yaw angle of the vehicle from the reference path. Its computation is pictured in figure 5.2a. The formula for the current heading error is as follows:

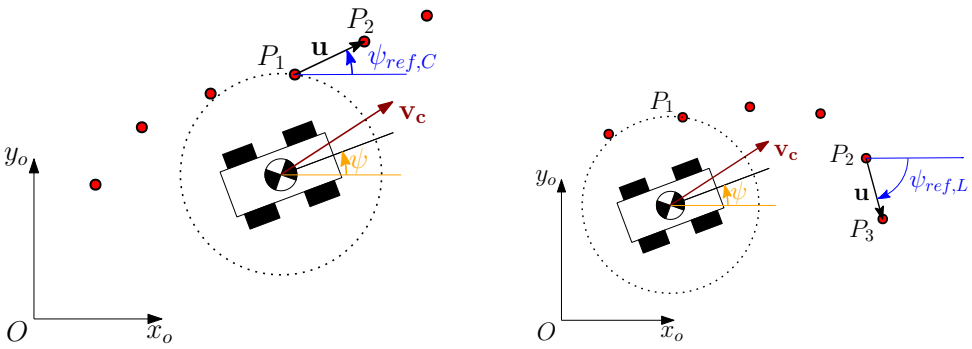
$$e_{\psi,C} = \psi_{ref,C} - \psi, \tag{5.2}$$

where the ψ_{ref} is the current reference yaw angle and ψ is the yaw angle of the vehicle.

The look-ahead heading error $e_{\psi,L}$ describes the deviation in the yaw angle of the vehicle from the look-ahead reference yaw angle. Its computation is pictured in the figure 5.2b for the look-ahead parameter $L = 3$. The formula for look-ahead heading error is as follows:

$$e_{\psi,L} = \psi_{ref,L} - \psi, \tag{5.3}$$

where the $\psi_{ref,L}$ is the look-ahead reference yaw angle and ψ is the yaw angle of the vehicle.



(a) : Current heading error. **(b) :** Look-ahead heading error.

Figure 5.2: Heading errors.

For detailed information on how to obtain the current and look-ahead reference yaw angle, please refer to section 3.2.3.

5.2 Decoupling of Control Actions

In this section, the decoupling of control actions will be presented. Two main control action members are used in the lateral controller architecture. Virtual control action member δ_ψ represents a control system requirement for turning the vehicle, i.e., the change in the vehicle's yaw angle to minimize the heading error. Virtual control action member $\delta_\perp$ represents a control system requirement for a change of vehicle c.g. position to minimize the cross-track error.

The actual control actions of the control system, which controls the vehicle, are the steering angles of the front and rear wheels: δ_f and δ_r. They can be computed in many ways with the use of the virtual control action members δ_ψ and $\delta_\perp$. In this thesis, the principle pictured in figure 5.3 is used. The principle is described by the following formula:

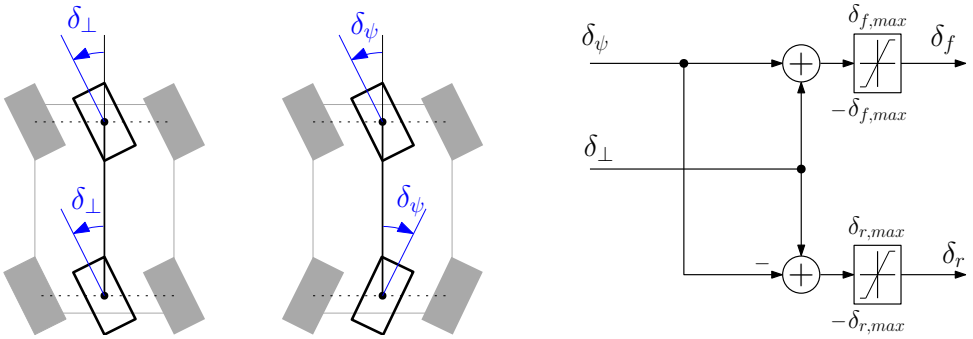
$$\delta_f = \delta_\psi + \delta_\perp, \tag{5.4}$$

$$\delta_r = -\delta_\psi + \delta_\perp. \tag{5.5}$$

Two saturation blocks are applied as a representation of mechanical saturation of the steering angles of the wheels $\delta_{f,max}$ and $\delta_{r,max}$.

5.3 Controller Architecture

The first part of the control system design is the design of the architecture. The architecture of the lateral controller will be presented in this section. The architecture is based on the decoupling concept, discussed in section 5.2. The



(a) : Vehicle view. **(b) :** Block diagram.

Figure 5.3: Decoupling of control action members.

two virual control action members $\delta_\perp$ and δ_ψ, are used for the computation of actual control actions δ_f and δ_r.

The block diagram of the controller architecture is shown in figure 5.4. The scheme shows the principle of calculation of the virtual control action members. The architecture includes two main controllers, the Cross-track controller and the Yaw controller. Each of these controllers generates one of the virtual action members depending on their inputs, which are the path-tracking errors. A saturation block is used to limit the maximal value of $\delta_\perp$.

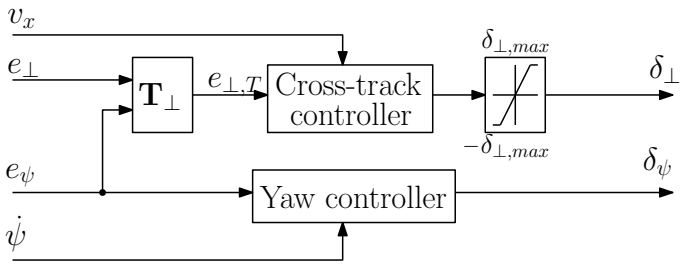


Figure 5.4: Laretal Controller Structure.

The Cross-track controller and the Yaw controller will be designed separately. For certain scenarios, it is possible to assume that one of the virtual control actions will be zero. For others, however, both will be non-zero and both will control the motion of the vehicle. Therefore, additional requirements for the design of the control system arise. It is required that the step response of the Yaw controller be approximately twice as fast as the step response of the Cross-track controller. In addition, a system for the priority of the Yaw controller is presented in section 5.6.

The design process of the Cross-track controller 5.5 and the Yaw controller 5.4 will follow the procedure already mentioned at the beginning of this chapter.

5.4 Yaw Controller Design

The purpose of the Yaw controller is to create sufficient virtual control action δ_ψ to minimize the heading error.

5.4.1 Transfer Function Derivation

For model-based controller design, the transfer function from the input steering angle to the vehicle's yaw angle is needed. Assume that the virtual control action $\delta_\perp$ is zero throughout the vehicle's motion. The wheels are turned only by the δ_ψ signal. It is then possible to create a state-space model with only one input. The equations for this model, which are based on the equations 2.19, 2.20, are as follows:

$$\begin{bmatrix} \dot{\beta} \\ \ddot{\psi} \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} -\frac{C_f+C_r}{mv_c} & \left(\frac{l_rC_r-l_fC_f}{mv_c^2}-1\right) & 0 \\ \frac{l_rC_r-l_fC_f}{I_z} & -\frac{C_fl_f^2+C_rl_r^2}{I_zv_c} & 0 \\ 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} \beta \\ \dot{\psi} \\ \psi \end{bmatrix} + \begin{bmatrix} \frac{C_f-C_r}{mv_c} \\ \frac{C_fl_f+C_rl_r}{I_z} \\ 0 \end{bmatrix} \delta_\psi \tag{5.6}$$

The transfer function $G_\psi(s)$ in respect to the input δ_ψ and the output ψ depends on the parameter v_c. The goal is to build a feedback control system according to the diagram in figure 5.5.

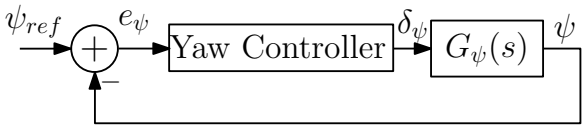
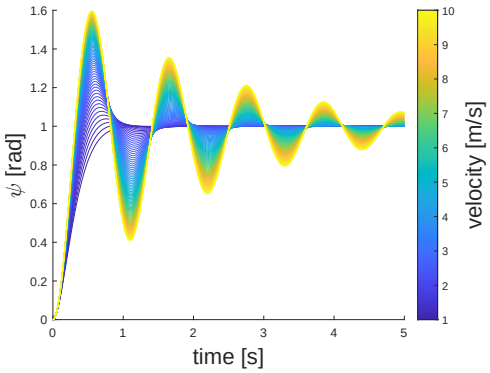


Figure 5.5: Feedback control system for yaw angle.

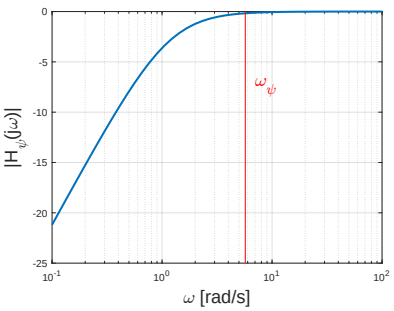
5.4.2 Yaw Damper

The first part of the controller design is the damping of the unwanted modes of the system. Closed-loop step response of the controlled system $G_\psi(s)$ is presented in figure 5.6a. With increasing velocity of the vehicle, it is possible to observe a significant increase in the oscillations of the vehicle's yaw angle. The angular frequency of these oscillations is $\omega_\psi = 5.71$ rad/s.

The addition of the yaw damper should lead to the elimination of the oscillations. As the vehicle's oscillations increase, so does the vehicle's yaw rate as the derivative of the vehicle's yaw angle. The virtual control action δ_ψ is then reduced if high values of yaw rate occur. The yaw controller structure with the addition of the yaw damper is presented in figure 5.7. The W/O block stands for washout filter. The washout filter is a stable high-pass filter with zero static gain. Oscillations at lower frequencies than ω_ψ are not considered as unwanted as they can be caused by changes in the reference signal. It is possible to model the filter to filter out frequencies lower than the frequencies of the unwanted oscillations. The transfer function $H_\psi(s)$ of



(a) : Close loop step response of $G_\psi(s)$.



(b) : Magnitude frequency response of washout filter.

Figure 5.6: The W/O filter designed for damping the unwanted oscillations in the vehile's yaw angle.

the washout filter is as follows:

$$H_\psi(s) = \frac{s}{s + \frac{\omega_\psi}{5}}. \tag{5.7}$$

Its magnitude frequency response is shown in figure 5.6b. With the addition of W/O filter, only the oscillations with frequencies higher and around the ω_ψ are damped.

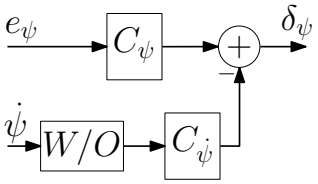


Figure 5.7: Yaw controller structure.

5.4.3 Controller Performance

Two PID controllers are used in the Yaw Controller block, as shown in figure 5.7. The coefficients of the controllers C_ψ and $C_{\dot\psi}$ were tuned in respect of the following performance criteria for the vehicle's velocity range $v_c \in \langle 1, 10\rangle$ $[\mathrm{m}\cdot\mathrm{s}^{-1}]$:

- Keep the overshoot below 20 %.
- Achieve a settling time of under 3 s.
- Achieve oscillation-free response, i.e., the step response must leave the $\pm 5\,\%$ band at most once.
- Achieve a zero steady-state error.

The results of the tuning of the yaw controller can be seen in figure 5.8.

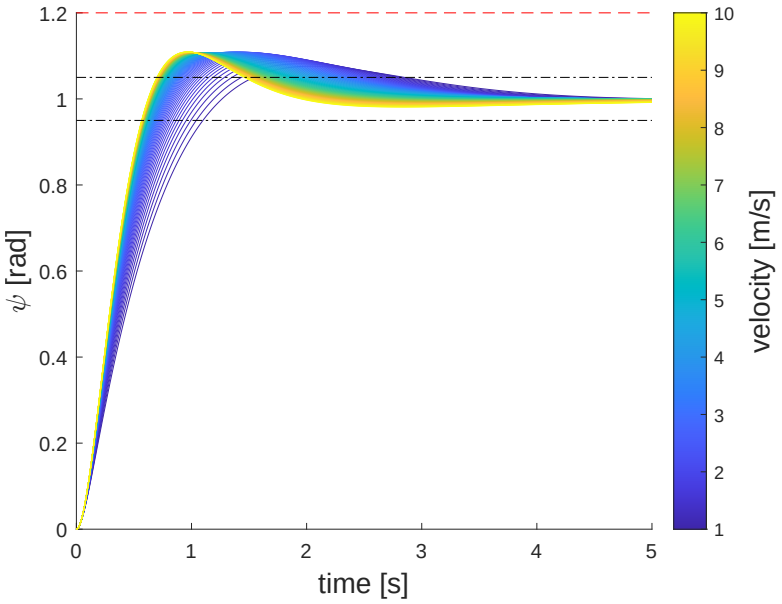


Figure 5.8: Step response of $G_\psi(s)$ with the designed yaw controller.

5.4.4 Path Preview

If the control algorithm uses only the current heading and cross-track error components, it is called a reactive algorithm. The vehicle has deviated from the reference path and the control algorithm reacts to this deviation. If a certain section of the reference path ahead of the vehicle is known, it is possible to work with this information as well. This allows the vehicle to react to changes in the reference path that have not yet occurred. In practice, this can lead to an improvement of the driving line, especially when the reference path changes abruptly.

In section 5.1, the current and look-ahead heading errors were presented. Figure 5.9 shows the use of both of these errors. The resulting heading error e_ψ is given by the convex combination of the current heading error $e_{\psi,C}$ and the look-ahead heading $e_{\psi,L}$ with the coefficients k_C and k_L.

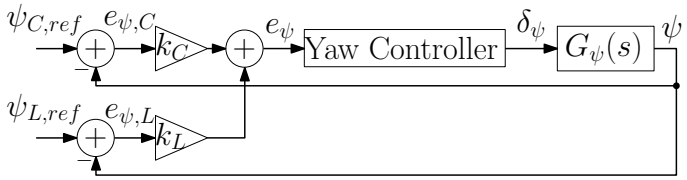


Figure 5.9: The use of the look-ahead heading component.

5.4.5 Saturation of the Cross-Track Error

As the vehicle moves, the cross-track error value may rise above the specified limit $e_{\perp,max}$. If the designed control algorithm is correct, this situation should only occur during a path planning failure or real-time reference path

rescheduling when there is no feasible way for the vehicle to stay close to the reference path.

In this scenario, it is desirable to reduce the value of the cross-track error $e_{\perp,T}$ also by rotating the vehicle. In the figure 5.10 the architecture for dealing with the saturation of the cross-track error is presented.

If the cross-track error value exceeds $e_{\perp,max}$ (limit value of the first saturation block), then the error $e_{\perp\to\psi}$ is non-zero. The second saturation block is to limit the error $e_{\perp\to\psi}$. Large $e_{\perp\to\psi}$ values could cause the vehicle to oversteer, so it is advisable to limit the error value systematically, as shown. Saturation limit values of $\pm\frac{pi}{2}$ are suggested to keep the vehicle perpendicular to the reference road at the maximum error value.

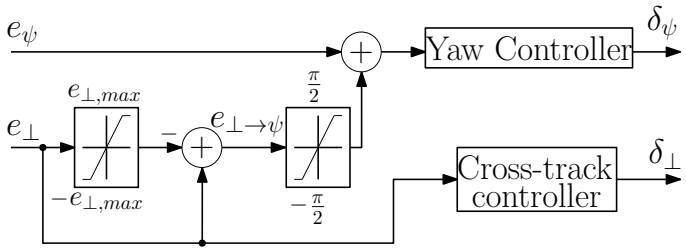


Figure 5.10: Controller architecture for the cross-track error saturation.

5.5 Cross-Track Controller

The purpose of the Cross-track controller is to create sufficient virtual control action $\delta_\perp$ to minimize the cross-track error.

5.5.1 Cross-Track Error Transformation

The cross-track error represents the signed perpendicular distance of the c.g. of the vehicle from the reference path. Using the decoupling mentioned in section 5.2, however, we need to transform this error so that the control action $\delta_\perp$, which is generated by this error, helps to reduce this error. Only if the vehicle is aligned with the reference path the maximum steering angle of both wheels is the optimal way to reduce cross-track error.

Figure 5.11 shows the transformation of the cross-track error so that the new transformed cross-track error $e_{\perp,T}$ represents the need to rotate the wheels to reduce the cross-track error $e_\perp$. The transformation formula is as follows:

$$e_{\perp,T} = e_\perp |\cos(e_\psi)|. \tag{5.8}$$

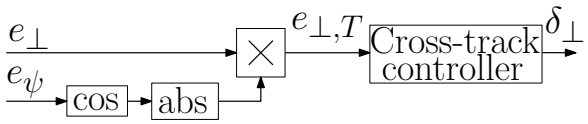


Figure 5.11: The cross-track error transformation.

5.5.2 Transfer Function Derivation

For model-based controller design, the transfer function from the input steering angle to the cross-track error is needed. In [11] the single-track model in terms of error with respect to road is presented. This model is based on the equations of the linear single-track model 2.19, 2.20.

The principle described in section 5.2 is used, so that the $\delta_\perp$ is the only input of the system. Assume that the virtual control action δ_ψ is zero throughout the vehicle's motion. The wheels are turned only by the $\delta_\perp$ signal. The equations of the single-track error model are as follows:

$$\begin{bmatrix} \dot{e_\perp} \\ \ddot{e_\perp} \\ \dot{e_\psi} \\ \ddot{e_\psi} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & -\frac{C_f+C_r}{mv_x} & \frac{C_f+C_r}{m} & \frac{-C_f l_f+C_r l_r}{mv_x} \\ 0 & 0 & 0 & 1 \\ 0 & -\frac{C_f l_f-C_r l_r}{I_z v_x} & \frac{C_f l_f-C_r l_r}{I_z} & -\frac{C_f l_f^2+C_r l_r^2}{I_z v_x} \end{bmatrix} \begin{bmatrix} e_\perp \\ \dot{e_\perp} \\ e_\psi \\ \dot{e_\psi} \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{C_f+C_r}{m} \\ 0 \\ \frac{C_f l_f-C_r l_r}{I_z} \end{bmatrix} \delta_\perp \tag{5.9}$$

The transfer function $G_\perp(s)$ in respect to the input $\delta_\perp$ and the output $e_\perp$ depends on the parameter v_c. The goal is to build a feedback control system according to the diagram in figure 5.12. For a non-zero initial value of the cross-track error $e_{\perp,0}$, the goal is to control the cross-track error $e_\perp$ to follow zero asymptotically.

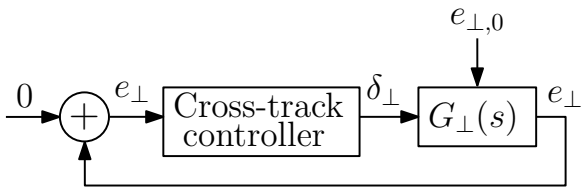


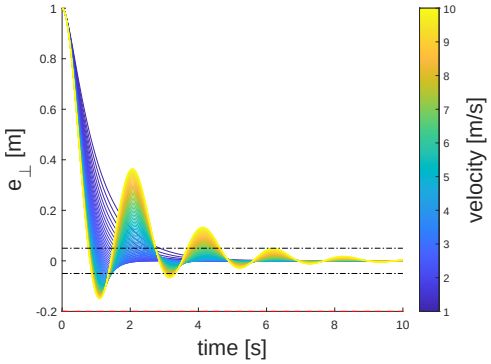
Figure 5.12: Feedback control system for the cross-track error.

5.5.3 Cross-Track Error Damper

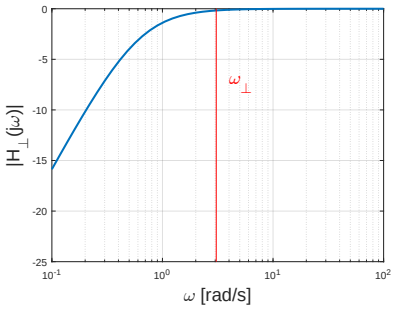
A similar approach, as in section 5.4.2, will be taken. The first part of the controller design is the damping of the unwanted modes of the system. Closed-loop response on the initial value $e_{\perp,0} = 1$ of the controlled system $G_\perp(s)$ is presented in figure 5.13a. With an increasing velocity of the vehicle, it is possible to observe a significant increase in the oscillations of the cross-track error. The angular frequency of these oscillations is $\omega_\perp = 3.07$ rad/s.

The addition of the cross-track damper should lead to the elimination of the oscillations. As the oscillations in the cross-track error increase, so does the derivative of the cross-track error. The virtual control action $\delta_\perp$ is then reduced if high values of $\dot{e}_{\perp,T}$ occur. The Cross-track controller structure with the addition of the cross-track error damper is presented in figure 5.14. Again, the W/O filter is used for filtering out the lower frequencies than $\omega_\perp$. The transfer function $H_\perp(s)$ of the washout filter is as follows:

$$H_\perp(s) = \frac{s}{s + \frac{\omega_\perp}{5}}. \tag{5.10}$$



(a) : Close loop step response of $G_{\perp}(s)$.



(b) : Magnitude frequency response of washout filter.

Its magnitude frequency response is shown in figure 5.13b.

5.5.4 Velocity Scaling

A similar approach, as in [1], is used for scaling the cross-track error. For higher velocities, it is desirable to limit the interventions to reduce the cross-track error.

$$\frac{k\, e_{\perp,T}}{k_{soft} + v_x} \tag{5.11}$$

The controller structure with the cross-track damper presented in section 5.5.3 and with the velocity scaling is shown in figure 5.14.

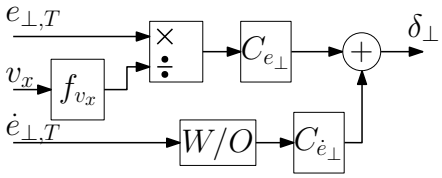


Figure 5.14: Cross-track controller structure.

5.5.5 Controller Performance

Two PID controllers are used in the Cross-track controller block, as shown in figure 5.14. The coefficients of the controllers $C_{e_\perp}$ and $C_{\dot{e}_\perp}$ were tuned in respect of the following performance criteria for the vehicle's velocity range $v_c \in \langle 1, 10 \rangle$ $[\mathrm{m} \cdot \mathrm{s}^{-1}]$:

- Keep the overshoot below 2 %.
- Achieve a settling time of over 2 s.
- Achieve a settling time of under 6 s.
- Achieve a slower settling time for higher velocities to keep higher driving-comfort levels.

- Achieve a zero steady-state error.

The results of the tuning of the cross-track controller can be seen in figure 5.15.

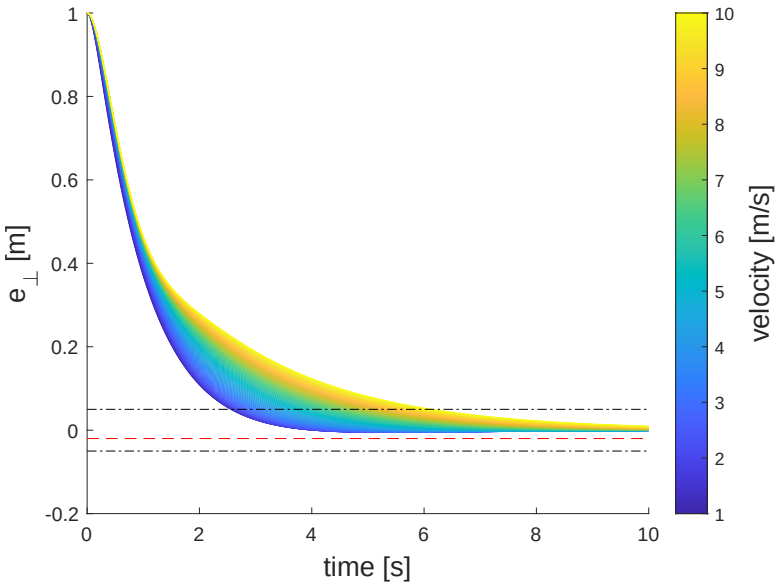


Figure 5.15: Step response of $G_{\perp}(s)$ with tuned cross-track controller.

5.6 Yaw Controller Priority

Due to the decoupling of control actions described in section 5.2, there are two separate controllers and two separate virtual control actions δ_{ψ} and $\delta_{\perp}$ that generate two actual control actions δ_f and δ_r. With the mechanical saturation of the wheels $\delta_{f,max}$ and $\delta_{r,max}$, the two virtual control actions can be compromised. It is then appropriate to further limit one of the two virtual control actions so that the other one can be performed optimaly. In this thesis, the designed control system has a yaw controller preference. When a saturation of the wheels is detected, the output of the cross-track controller $\delta_{\perp}$ is limited. The block diagram describing the detection of saturation and resulting $\delta_{\perp}$ limitation is shown in figure 5.16.

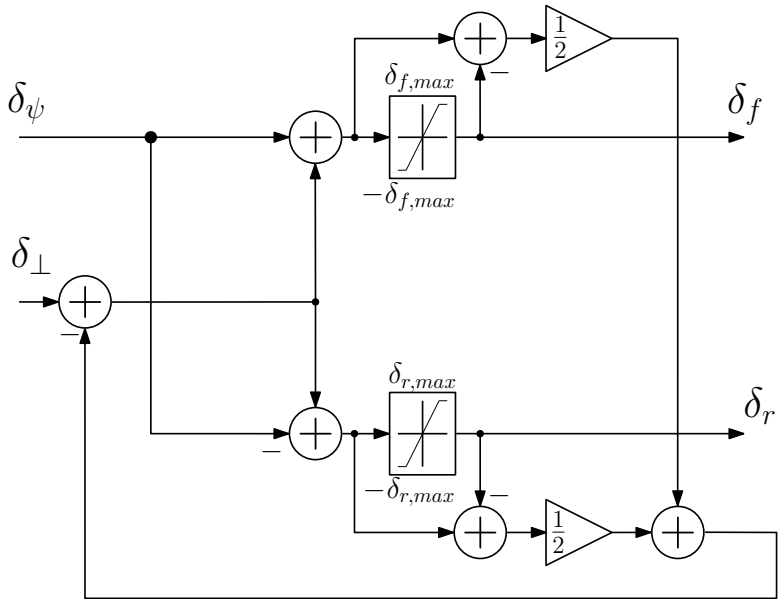


Figure 5.16: Limitation of $\delta_\perp$ when wheels saturation is detected.

Chapter 6

Validation Platform

This chapter will give a brief summary of the vehicle platform on which the designed control system is tested. It is a modified version of the RC car Losi Desert Buggy to execute more computationally demanding algorithms. Neither the vehicle hardware nor the communication protocol will be described in this thesis. For more information about this please refer to [18].

Figure 6.1: A photo of the validation platform.

6.1 Local Tangent Plane Coordinate System

The local tangent plane system is commonly used in robotics and autonomous driving. The origin of this coordinate system is in a point that lies on the Earth's surface, which is modeled as an ellipsoid. The normal of this tangent plane points downwards, as shown in figure 6.2. The x-axis points in the direction to the north, the y-axis points in the direction to the east.

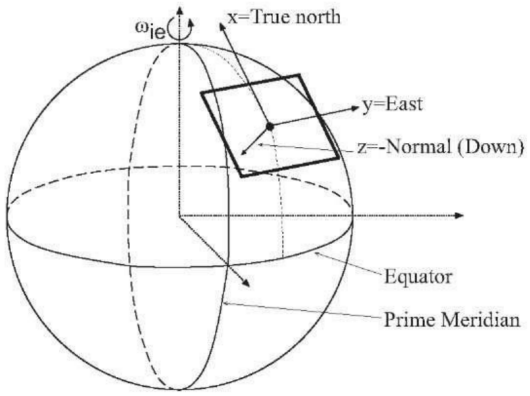


Figure 6.2: The scenario for the test of both lateral and longitudinal controllers. [21]

In order to have all models and experiments in one coordinate system, we need transformations of the NED coordinates to the O coordinate system.

$$x_o = y_{NED} \tag{6.1}$$

$$y_o = x_{NED} \tag{6.2}$$

6.2 Inputs

As an over-actuated vehicle, there is the possibility to control both steering angles of the front and rear axle. The wheels are turned by servo motors. An Ackermann steering geometry is implemented into this platform. The Ackermann steering geometry is designed mechanically; the inputs of the vehicle are the front and rear steering angles δ_f, δr, not the steering angles of each right and left wheel.

The vehicle can be controlled by a remote control system with steering angles and throttle inputs or an autonomous control system, with the longitudinal dynamics still being controlled by the throttle input.

6.3 Measurements

Several physical variables are measured throughout the vehicle's motion thanks to sensors integrated into the vehicle. A necessary condition for the

execution of the control algorithm is the knowledge of the position of the vehicle in the global coordinate system, the orientation of the vehicle, the lateral and longitudinal velocity of the vehicle and the yaw rate of the vehicle.

The measurement of the vehicle coordinates in the global coordinate system is performed by dual GPS (Global Positioning System) system attached to the vehicle and a base GPS station. The position of the base is denoted as the origin of the global coordinate system in NED coordinates. This GPS measurement also provides the measurement of the vehicle's yaw angle in the global coordinate system and the orientation of the velocity vector of the vehicle.

The velocity is measured by Hall sensors attached to the front axle of the vehicle. Both the angular frequency of the rotational speed of the front wheels and the tangential speed of the front wheels are measured. The speed measurement is performed only on the front wheels of the vehicle. This is because of the rear drive shaft of the vehicle. Since the vehicle's engine does not generate any torque on the front wheels, the longitudinal slip of the front wheels can be considered zero.

The yaw rate measurements are done with the use of a gyroscope.

The steering angles of the vehicle are not measured, only the reference signals of the servos.

6.4 Path-Planning

The planning of the reference route is done in the vehicle by acquiring the image from the camera that is attached to the vehicle. The data from the camera is processed by a neural network that distinguishes the types of surfaces that are in front of the vehicle. From this data, a planning algorithm is then used to design the optimal path for the vehicle.

It is also possible to use already created driving scenarios to upload as a reference path to the vehicle. The reference path is a sequence of x,y and yaw angle positions in the global coordinate system. This principle thus follows the principle described in section 3.2.3.

Chapter 7

Experiments

In this section, experiments with the designed control system will be presented. The experiments are simulation-based and validation-based. For the simulation experiments, the Simulation Framework described in section 3.2 is used. The validation experiments test the validity of the control system on the validation platform, described in chapter 6.

7.1 Testing Scenarios

Due to the method of decoupling the lateral and longitudinal vehicle dynamics, two controllers (lateral and longitudinal) were designed. The controllers will be tested both separately and simultaneously.

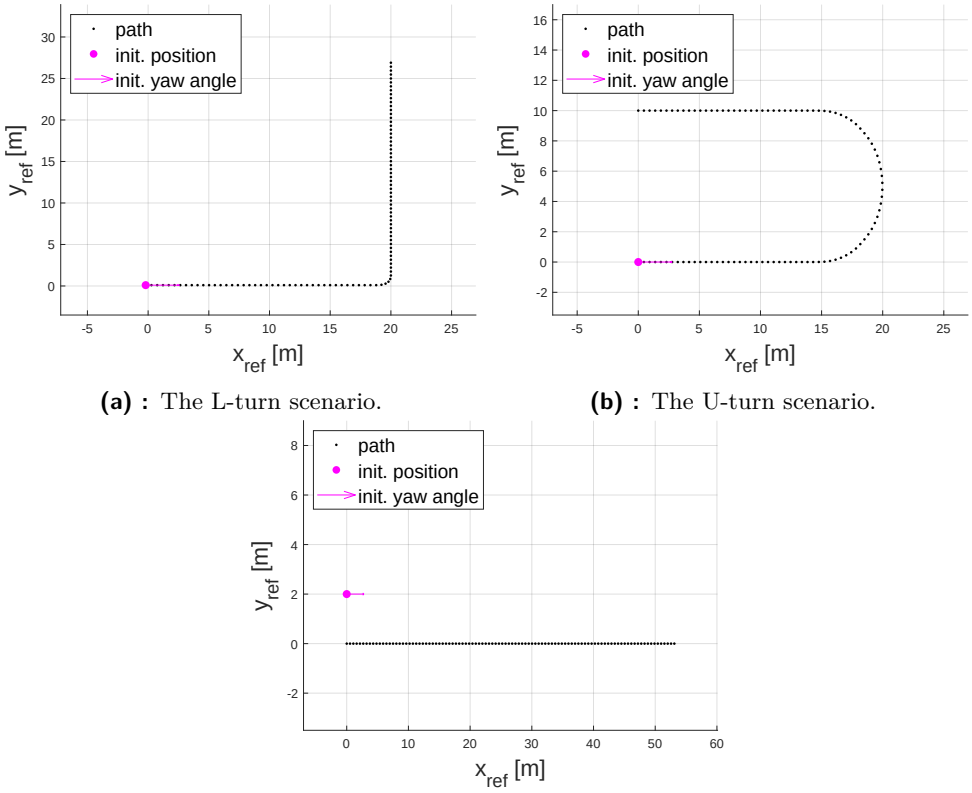
7.1.1 Lateral Dynamics

Three different testing scenarios, shown in figure 7.1, were prepared for testing the lateral controller. These experiments will be conducted at constant vehicle velocity $v_c = 3\ \mathrm{m} \cdot \mathrm{s}^{-1}$. This assumption allows us to neglect the longitudinal dynamics.

The L-turn scenario is not a feasible reference path for the vehicle. However, it can provide a good comparison of the tested algorithms in terms of overshoot and path-tracking errors. The U-turn scenario was chosen as a feasible scenario for the selected velocity. The straight-line scenario should test the cross-track controller, as the main focus is minimizing the overshoot in cross-track error.

Specifically, the following data will be recorded for the following analysis of performance and driving comfort:

- The maximal absolute value of the cross-track error $e_{\perp,max}$ in metres for the L-turn and U-turn scenario.
- The RMS (root mean square) value of the cross-track error $e_{\perp,rms}$ in metres for the L-turn and U-turn scenario.
- The maximal absolute value of the current heading error e_{ψ} in degrees for the L-turn and U-turn scenarios.



(a) : The L-turn scenario.

(b) : The U-turn scenario.

(c) : The straight-line scenario.

Figure 7.1: Testing scenarios for lateral controller.

- The RMS value of the current heading error e_ψ in degrees for the L-turn and U-turn scenarios.
- The overshoot of the cross-track error $OS_\perp$ in metres for the straight-line scenario. The overshoot is measured as the highest absolute value of the cross-track error after the error reaches zero for the first time.
- The settling time τ_s in seconds for the straight line scenario. The settling time is measured as the time from the start of the simulation to the time when the absolute value of the cross-track error first drops below 0.1 metre.
- The RMS comfort coefficient $C_{comf,rms}$. This indicator includes the vehicle's yaw rate $\dot{\psi}$, lateral acceleration a_y and lateral jerk j_y values, that are weighted by the coefficient w_1, w_2, w_3. This comfort function is a subjective assessment of driving comfort and thus has limited informational value. (Proposed coefficient values: $w_1 = 0.4$, $w_2 = 0.3$, $w_3 = 0.3$)

$$C_{comf,rms} = \mathrm{RMS}(w_1 \left|\dot{\psi}\right| + w_2 \left|a_y\right| + w_3 \left|j_y\right|) \tag{7.1}$$

The abruptness of steering inputs can also be included in the driving comfort section, especially if there is a link between the steering wheel and the steering

angles of the wheels. In commercial vehicles, autonomous driving functionality is still very limited, mostly to parking assistance or line-keeping steering on the highway. So the vehicle must also have a steering wheel that is controlled by the driver. However, the steering wheel moves when the autonomous driving functionality is turned on. If there are abrupt interventions of the steering angles of the wheels, there are also abrupt movements of the steering wheel, which can cause concern for the driver despite the correct function.

7.1.2 Logitudinal Dynamics

To test the longitudinal dynamics, the straight-line scenario shown in figure 7.1c will be used. The experiments will include different initial velocities v_0 and different functions of reference velocities v_R, such as step, ramp and their combinations.

7.1.3 Combination of Dynamics

The final simulation experiment will test both lateral and longitudinal controllers simultaneously. The scenario is shown in figure 7.2. It is a double 45 degrees turn. The reference velocity is reduced to $v_R = 1\ \mathrm{m} \cdot \mathrm{s}^{-1}$ in and before the apex of the first corner. The reference velocity in the second corner is constant at $v_R = 5\ \mathrm{m} \cdot \mathrm{s}^{-1}$, so a significant overshoot is expected.

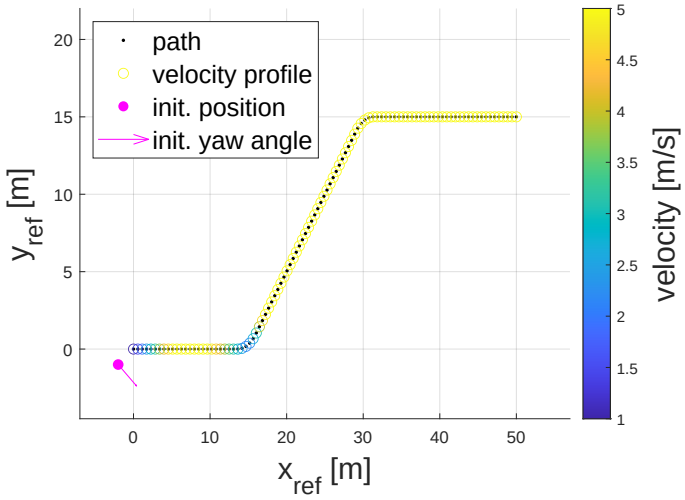


Figure 7.2: The scenario for the test of both lateral and longitudinal controllers.

7.2 Tested Algorithms

Multiple path-tracking algorithms were presented in section 5 to control the lateral dynamics of a vehicle. The control system designed in this thesis will be referred to as HA (Hierarchical Architecture) for the following experiments. It is a reactive feedback control system that does not use the look-ahead

component described in section 5.4.4. The control system using the look-ahead component will be referred to as LAHA (Hierarchical Architecture with the Look-Ahead component). The control system using the cross-track error saturation reduction method described in section 5.4.5 will be referred to as SRHA (Hierarchical Architecture with the cross-track error Saturation Reduction method).

A Stanley-inspired controller will be used as a reference controller for the performance and driving comfort comparison. This algorithm uses only the front steering angle δ_f as a control action. This Stanley-inspired controller, which does not use the look-ahead component, will be referred to as Stanley. The Stanley-inspired controller with the addition of the look-ahead component will be referred to as LAStanley.

Longitudinal controller, presented in section 4, will be referred to as HAcc (Hierarchical Architecture Cruise-Control).

7.3 Simulation Experiments

7.3.1 Stanley and HA comparison

Performance Evaluation

Table 7.1 compares the performance and driving comfort levels between the Stanley and HA path-tracking algorithms.

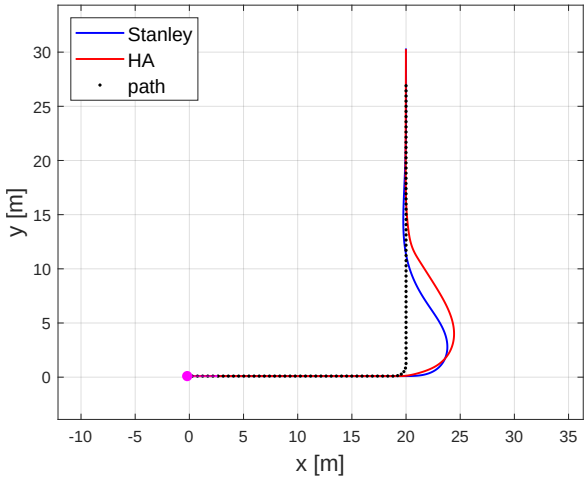
The values of maximal and RMS cross-track error are lower for the Stanley algorithm in both the L-turn and U-turn maneuvers pictured in figures 7.3 and 7.4. Althouth the value of maximal heading error is lower for the Stanley in the L-turn, it is possible to observe the significant overshoot in vehicle's yaw angle, shown in figure figure 7.3b. In the straight-line scenario, shown in 7.5, the overshoot of the HA algorithm is almost zero, which was one of the main focuses of the cross-track controller design.

The comfort levels are significantly lower for the Stanley algorithm. The values suggest a more aggressive approach of the HA algorithm.

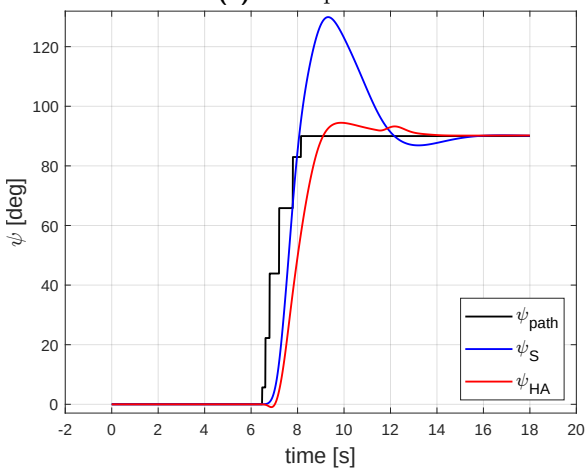
Performance and comfort - Stanley and HA						
Manuever:	L turn		U turn		straight	
Algorithm:	S	HA	S	HA	S	HA
$e_{\perp,max}$ [m]	**3.79**	4.41	**2.64**	3.77	-	-
$e_{\perp,rms}$ [m]	**1.26**	1.63	**1.54**	2.09	-	-
$e_{\psi,max}$ [deg]	**51.38**	64.35	**19.78**	32.24	-	-
$e_{\psi,rms}$ [deg]	14.93	**13.82**	**11.13**	12.41	-	-
OS_y [m]	-	-	-	-	0.17	**0.01**
τ_s [s]	-	-	-	-	**2.69**	2.85
$C_{comf,rms}$ [-]	**0.53**	0.98	**0.28**	0.55	**0.64**	1.74

Table 7.1: Performance and comfort comparison between Stanley and HA

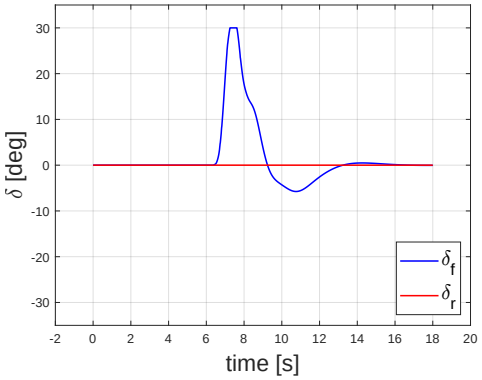
Simulated Scenarios



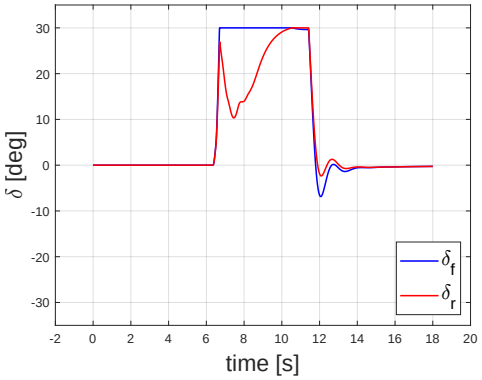
(a) : XY plot.



(b) : Plot of the yaw angles.

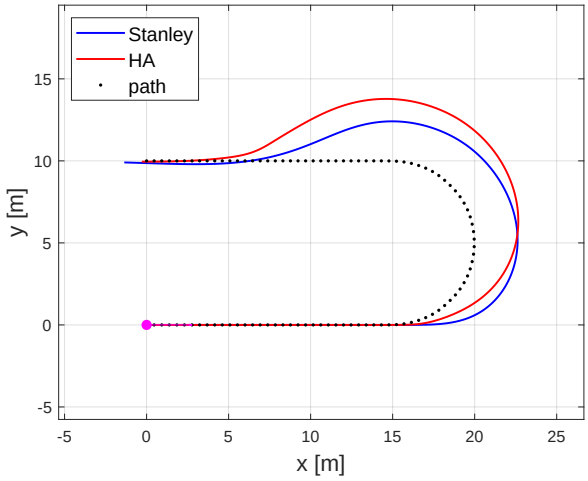


(c) : Plot of the steering angles - Stanley controller.

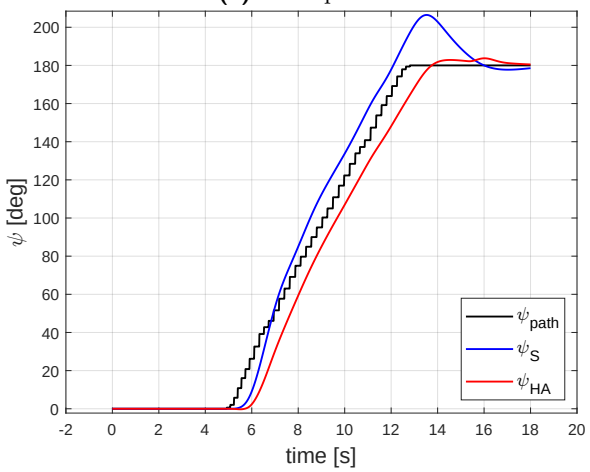


(d) : Plot of the steering angles - HA controller.

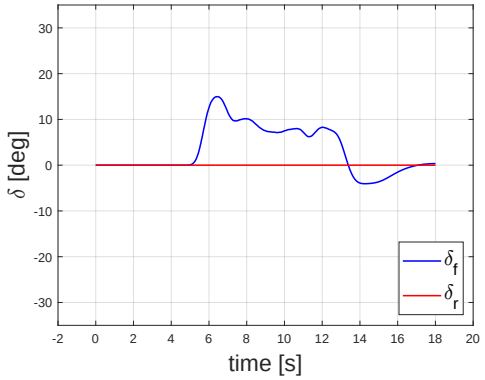
Figure 7.3: The L-Turn scenario.



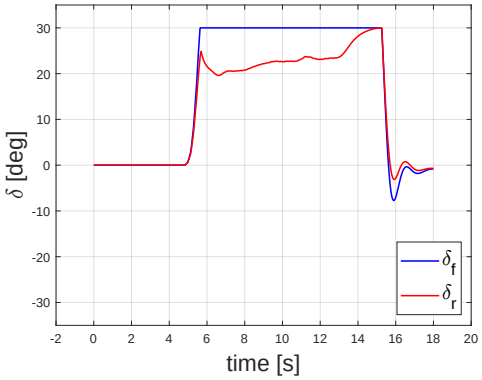
(a) : XY plot.



(b) : Plot of the yaw angles.

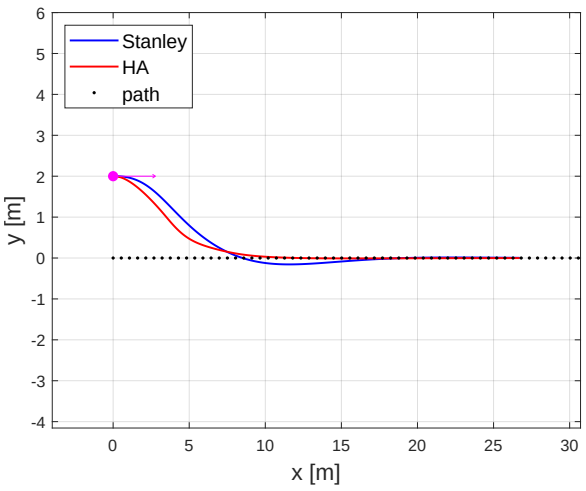


(c) : Plot of the steering angles - Stanley controller.

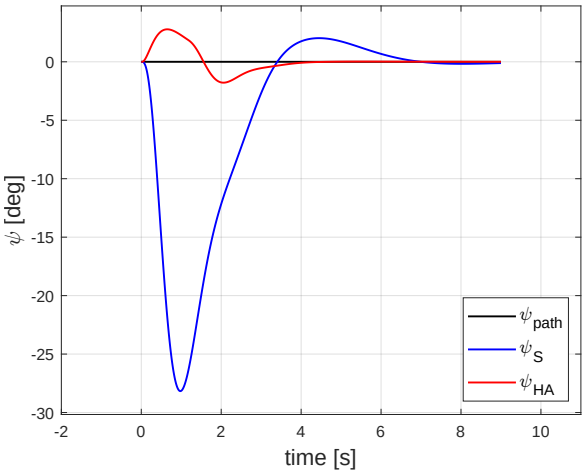


(d) : Plot of the steering angles - HA controller.

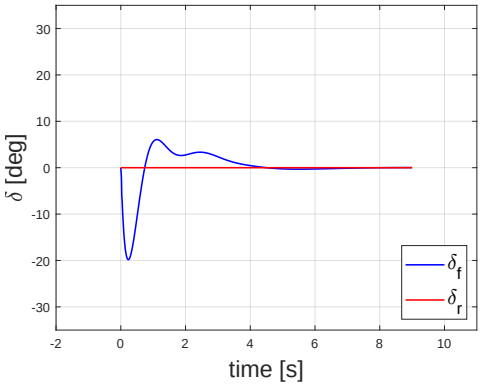
Figure 7.4: The U-Turn scenario.



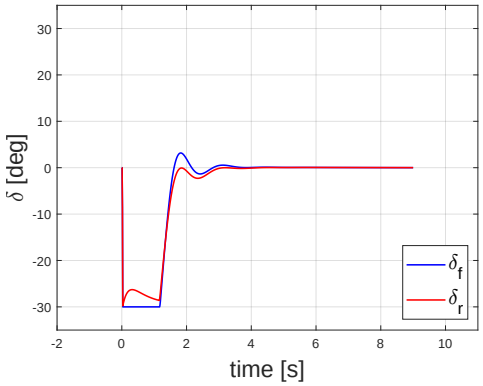
(a) : XY plot.



(b) : Plot of the yaw angles.



(c) : Plot of the steering angles - Stanley controller.



(d) : Plot of the steering angles - HA controller.

Figure 7.5: The straight-line scenario.

7.3.2 LAHA and HA comparison

Performance Evaluation

Table 7.2 compares the performance and driving comfort levels between the reactive HA algorithm and the look-ahead LAHA algorithm.

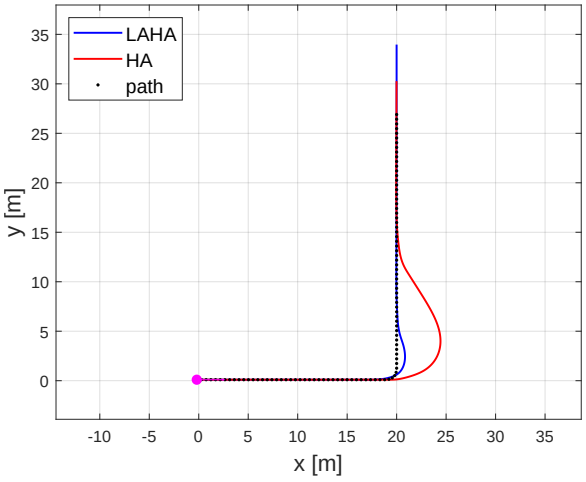
Significant improvements in performance are made by using the look-ahead heading error in performance. Both the maximal and RMS values of the path-tracking errors are much lower for LAHA. Especially with the U-turn, shown in 7.7, being a feasible maneuver, the cross-track error value does not exceed 0.15 metres. The comfort levels are very similar between the two algorithms.

In Conclusion, the look-ahead algorithm provides much better performance with similar comfort levels.

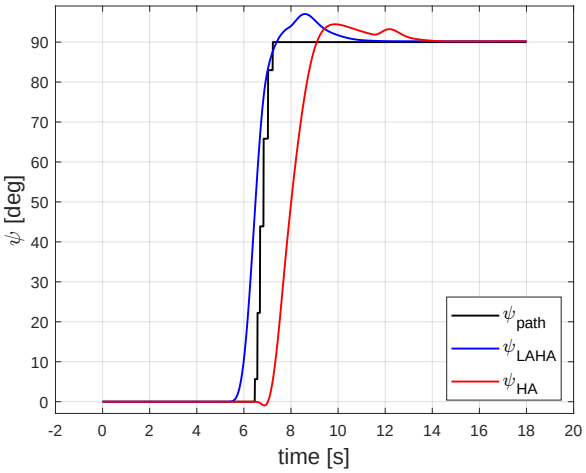
Performance and comfort - LAHA and HA				
Manuever:	L turn		U turn	
Algorithm:	LAHA	HA	LAHA	HA
$e_{\perp,max}$ [m]	**0.82**	4.41	**0.14**	3.77
$e_{\perp,rms}$ [m]	**0.19**	1.63	**0.05**	2.09
$e_{\psi,max}$ [deg]	**51.32**	64.35	**23.89**	32.24
$e_{\psi,rms}$ [deg]	**7.61**	13.82	**9.68**	12.41
OS_y [m]	-	-	-	-
τ_s [s]	-	-	-	-
$C_{comf,rms}$ [-]	1.09	**0.98**	**0.42**	0.55

Table 7.2: Performance and comfort comparison between LAHA and HA

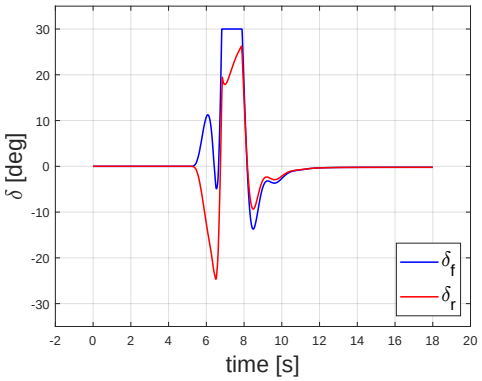
Simulated Scenarios



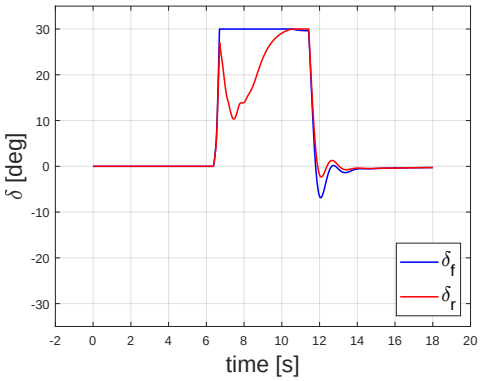
(a) : XY plot.



(b) : Plot of the yaw angles.

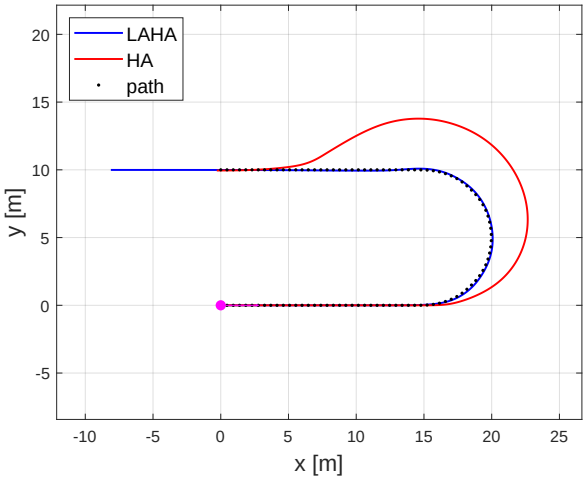


(c) : Plot of the steering angles - LAHA controller.

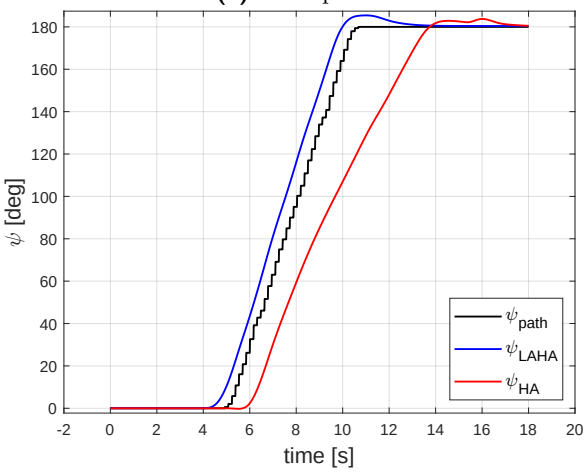


(d) : Plot of the steering angles - HA controller.

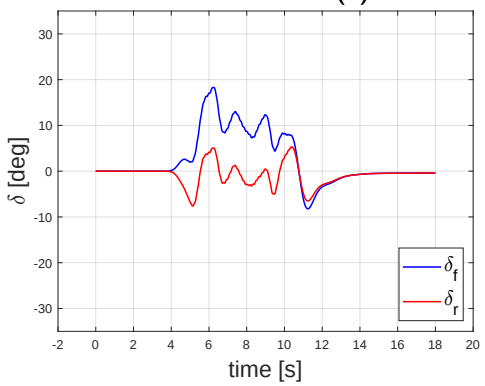
Figure 7.6: The L-turn scenario.



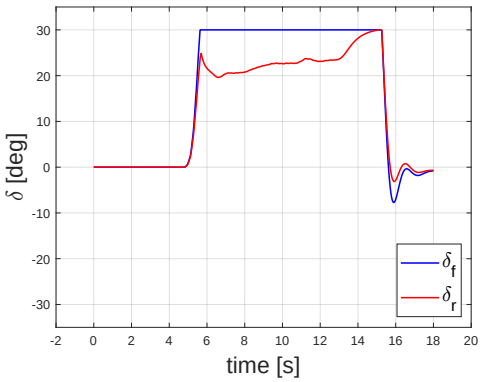
(a) : XY plot.



(b) : Plot of the yaw angles.



(c) : Plot of the steering angles - LAHA controller.



(d) : Plot of the steering angles - HA controller.

Figure 7.7: The U-turn scenario.

7.3.3 LAStanley and LAHA comparison

Performance Evaluation

Table 7.3 compares the performance and driving comfort levels between the LAHA and LAS algorithms, so two algorithms using the look-ahead component.

Significant improvements are made to both algorithms by using the look-ahead heading error in performance compared to the reactive versions. In the L-turn maneuver, shown in figure 7.8, the LAHA algorithm is more aggressive. Although similar values of maximal and RMS cross-track error, from the figure 7.8a, a vehicle using LAHA algorithm is back on the reference path in twice as less distance.

In figure 7.9, the U-turn scenario is presented. In figures 7.9c and 7.9d, the steering angles from the U-turn scenario are shown. Both algorithms generate oscillating control action caused by the oscillations in the cross-track error. Both algorithms do not possess the damping of the control action, as this feature was removed from the original Stanley controller.

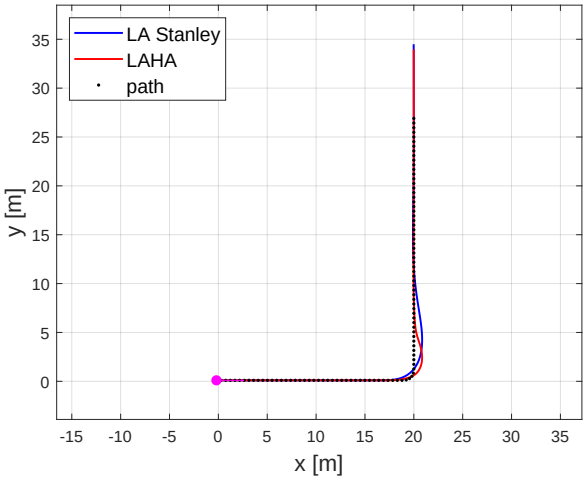
A possible solution could be to further slow down the cross-track controller (achieve a slower settling time of the step response with lower values of the control action) or add another control loop to restrain the control action.

In Conclusion, the look-ahead algorithms provide much better performance with similar comfort levels to their basic versions. The LAStanley provides a better comfort level than the LAHA, mostly due to lower values of lateral acceleration and jerk and overall less aggressive control approach.

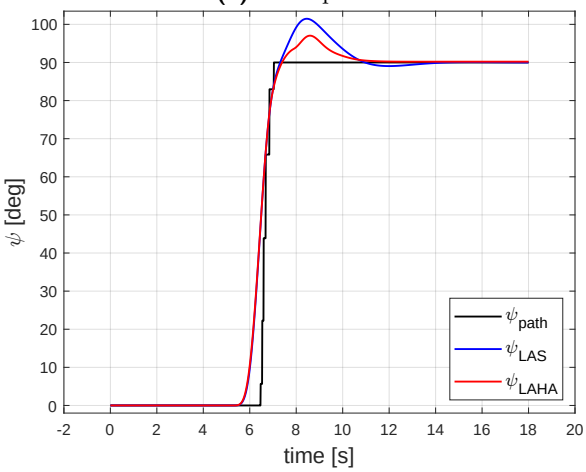
Performance and comfort - LAStanley and LAHA				
Manuever:	L turn		U turn	
Algorithm:	LAS	LAHA	LAS	LAHA
$e_{\perp,max}$ [m]	0.84	**0.82**	0.29	**0.14**
$e_{\perp,rms}$ [m]	0.25	**0.19**	0.12	**0.05**
$e_{\psi,max}$ [deg]	**47.58**	51.32	**22.28**	23.89
$e_{\psi,rms}$ [deg]	**6.80**	7.61	**8.65**	9.68
OS_y [m]	-	-	-	-
τ_s [s]	-	-	-	-
$C_{comf,rms}$ [-]	**0.54**	1.09	**0.22**	0.42

Table 7.3: Performance and comfort comparison between LAS and LAHA

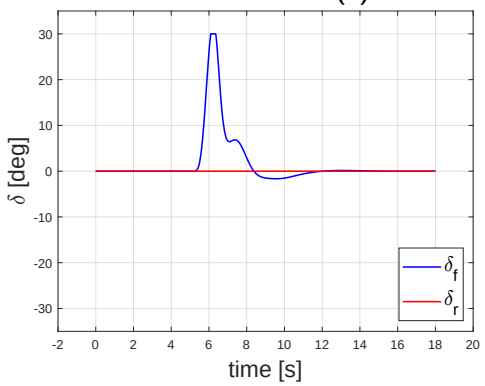
Simulated Scenarios



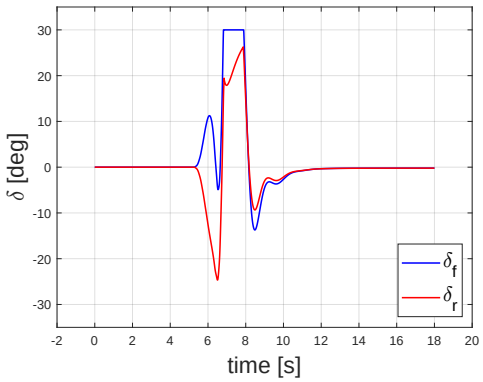
(a) : XY plot.



(b) : Plot of the yaw angles.

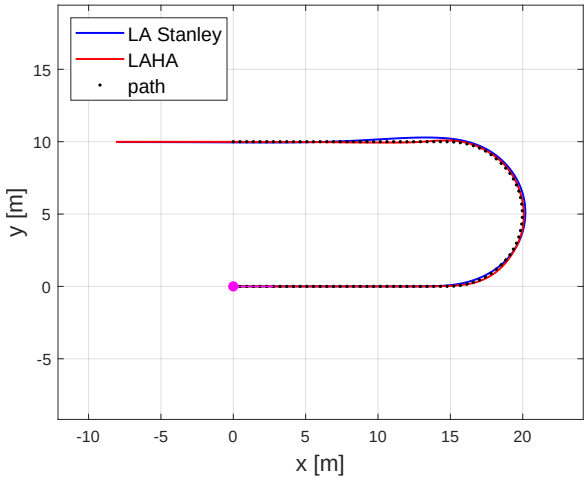


(c) : Plot of the steering angles - LA Stanley controller.

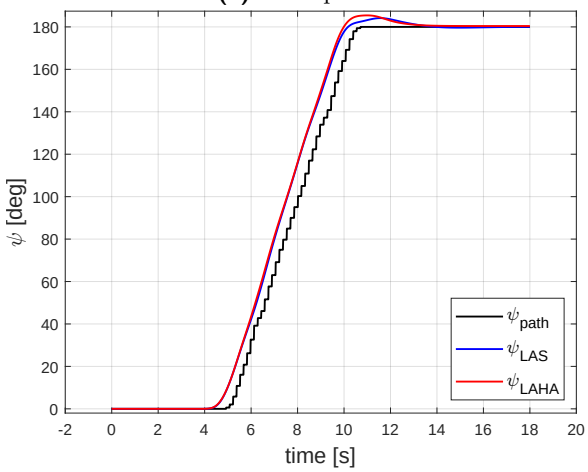


(d) : Plot of the steering angles - LAHA controller.

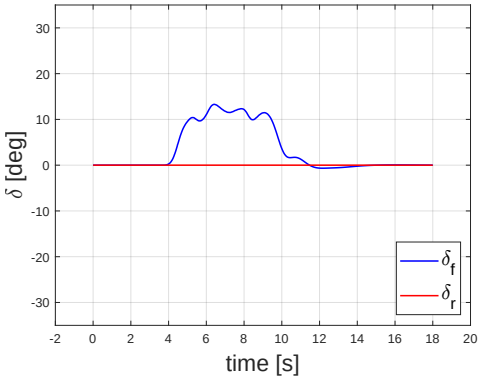
Figure 7.8: The L-turn scenario.



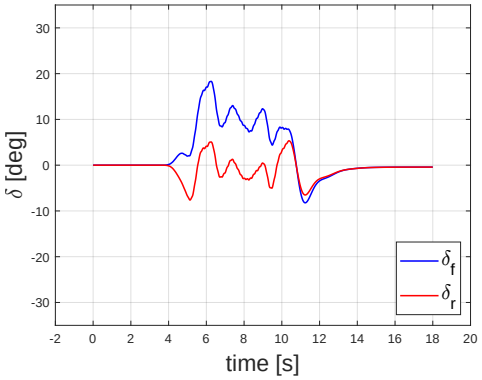
(a) : XY plot.



(b) : Plot of the yaw angles.



(c) : Plot of the steering angles - LA Stanley controller.



(d) : Plot of the steering angles - LAHA controller.

Figure 7.9: The U-turn scenario.

7.3.4 SRHA and HA comparison

Performance Evaluation

Table 7.4 compares the performance and driving comfort levels between the SRHA and HA algorithms.

The purpose of the HA algorithms with cross-track error saturation reduction method is to also use the virtual control action δ_ψ for faster cross-track error reduction.

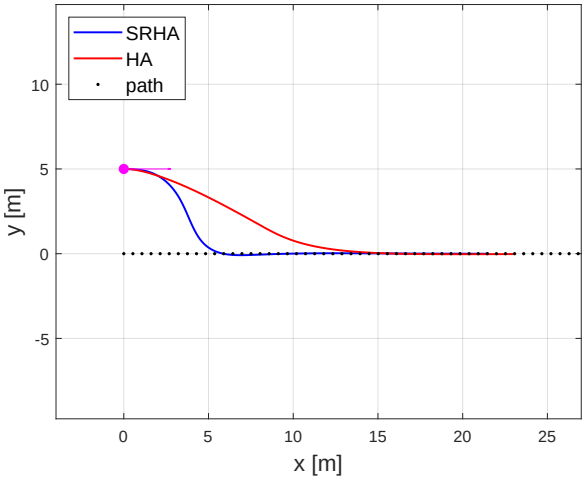
The comparison of the two algorithms was done using the straight-line scenario, shown in 7.10, so the values OS_y and τ_s were measured. For the SRHA, the settling time value is much lower, as expected.

Due to the more aggressive control action interventions, there is a higher values of the overshoot and comfort coefficient, but the value of the overshoot is still below 0.1 m, which can be considered excellent.

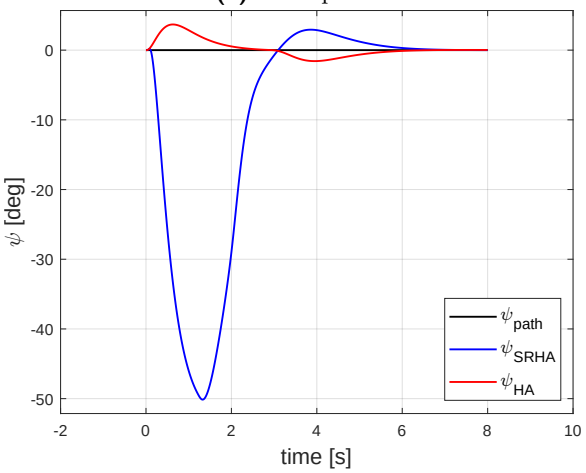
Performance and comfort - SRHA and HA		
Manuever:	Straight line	
Algorithm:	SRHA	HA
$e_{\perp,max}$ [m]	-	-
$e_{\perp,rms}$ [m]	-	-
$e_{\psi,max}$ [deg]	-	-
$e_{\psi,rms}$ [deg]	-	-
OS_y [m]	0.09	**0.03**
τ_s [s]	**2.65**	4.97
$C_{comf,rms}$ [-]	2.88	**1.72**

Table 7.4: Performance and comfort comparison between SRHA and HA

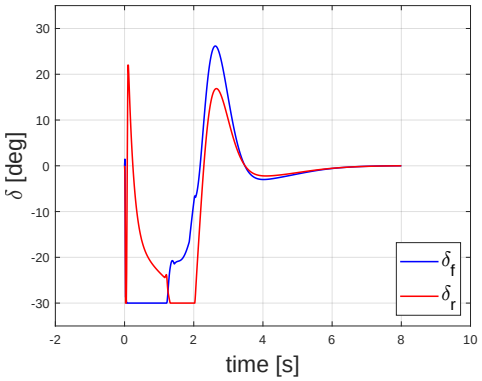
Simulated Scenarios



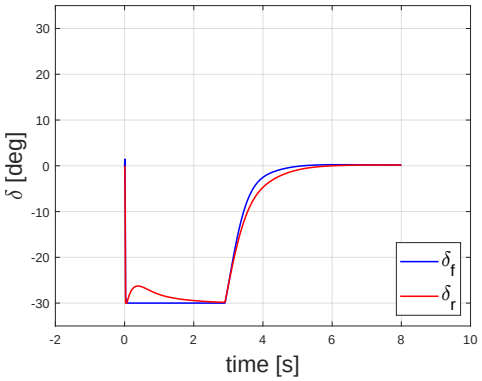
(a) : XY plot.



(b) : Plot of the yaw angles.



(c) : Plot of the steering angles - SRHA controller.



(d) : Plot of the steering angles - HA controller.

Figure 7.10: The straight-line scenario.

7.3.5 HAcc

Performance Evaluation

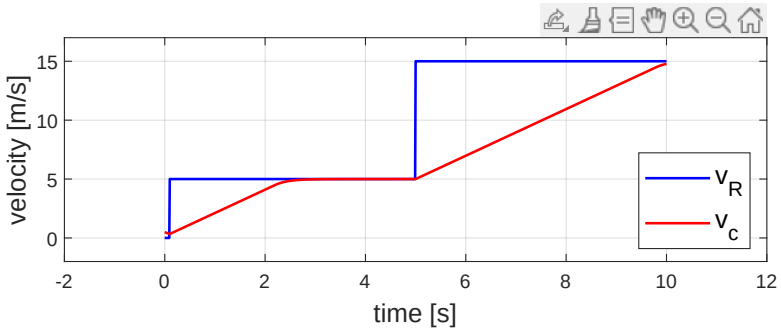
In figure 7.11, several testing scenarios were made to show the behavior of the longitudinal controller.

The main objective was to achieve an overshoot-free response. The acceleration controller (the architecture of the controller is shown in section 4) was thus made as a proportional controller, with the saturation levels of $a_{max} = 2$ $\mathrm{m} \cdot \mathrm{s}^{-2}$ for the acceleration and $a_{max} = 7$ $\mathrm{m} \cdot \mathrm{s}^{-2}$ for the deccalaration.

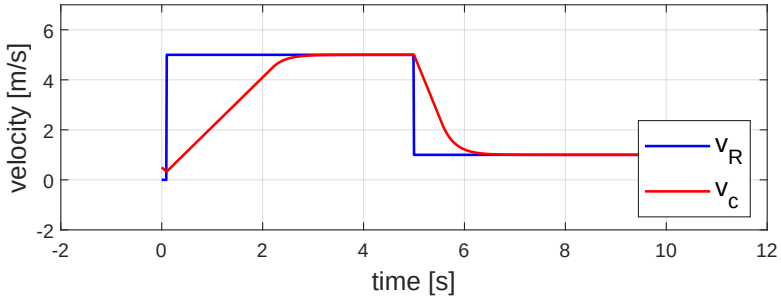
As shown in figures 7.11c, 7.11d, there is a non-zero steady-state error for the ramp reference signal. For zero steady-state ramp tracking, a higher level of system astatism is required. [22]

In conclusion, the longitudinal controller meets the basic parameters of an adaptive cruise-control system.

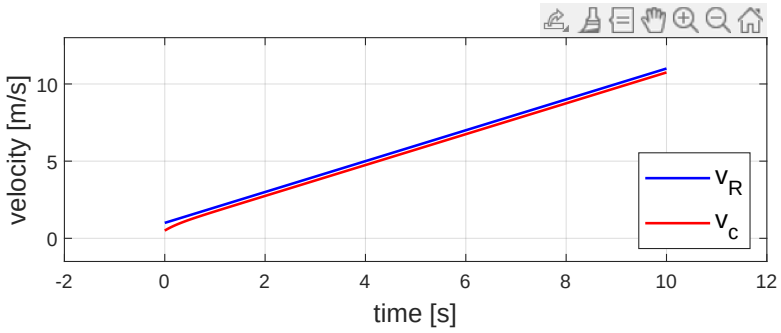
Simulated Scenarios



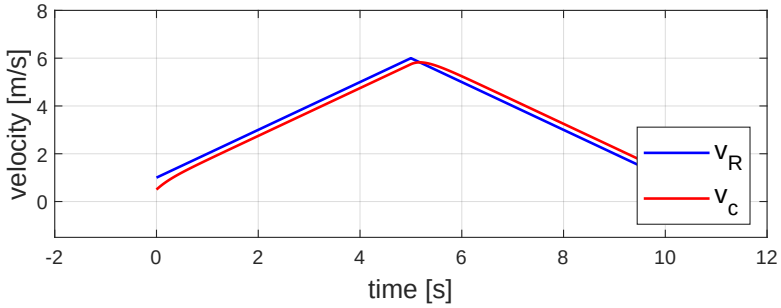
(a) : Plot of the reference and actual velocity.



(b) : Plot of the reference and actual velocity.



(c) : Plot of the reference and actual velocity.



(d) : Plot of the reference and actual velocity.

Figure 7.11: The testing of the longitudinal controller.

7.3.6 HAcc and LAHA Combination

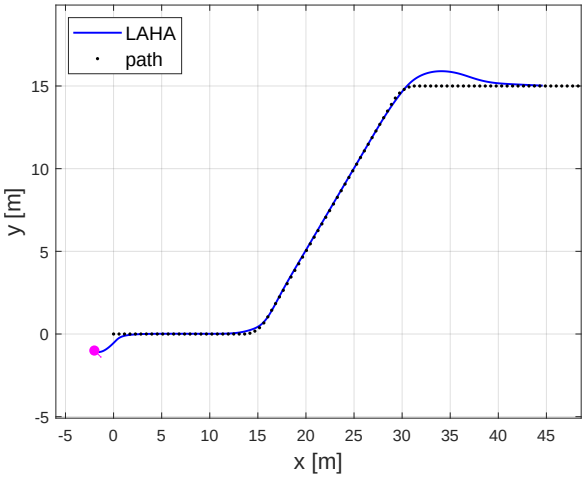
Performance Evaluation

For the trajectory test, the longitudinal controller HAcc and the better-performing LAHA lateral controller were chosen. The reference trajectory tracking test consists of adding reference velocity information to each segment of the reference path. Thus, the vehicle's velocity no longer needs to be constant throughout the movement of the vehicle. In section 7.1.3, the trajectory profile is shown.

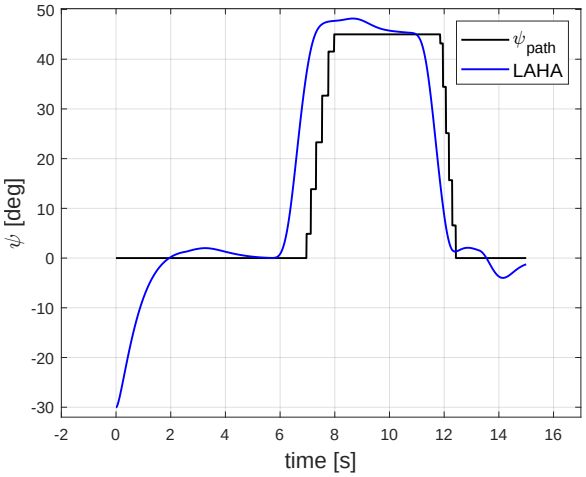
In figure 7.12, the experiment is shown. The lateral controller performance can be mainly judged by figures 7.12a and 7.12b, showing the vehicle's motion in the XY plane (O coordinate system) and the vehicle's yaw angle. As the velocity in the first turn was decreased, and with the use of LAHA algorithm, there is no overshoot. The reference velocity in the second turn was higher, so an overshoot was expected.

The velocity profile is shown in figure 7.12d. The longitudinal controller responds to changes in the reference signal and replicates the reference signal reasonably closely.

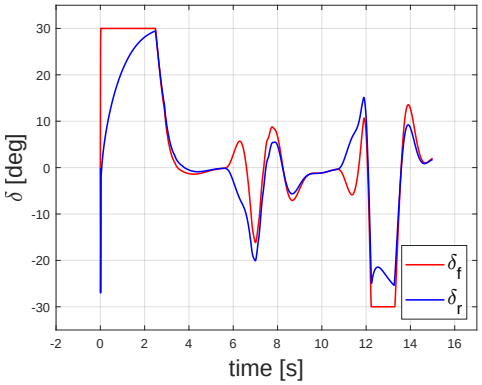
Simulated Scenarios



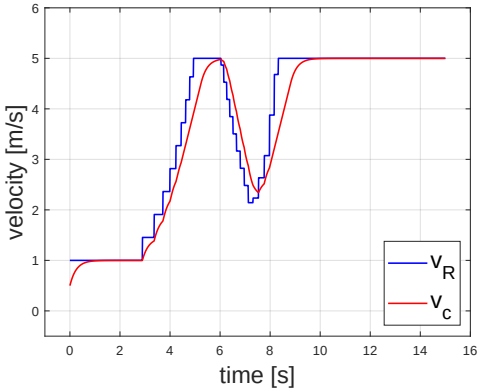
(a) : XY plot.



(b) : Plot the yaw angles.



(c) : Plot of the steering angles.



(d) : Plot of the reference and actual velocity.

Figure 7.12: The testing of the combined loading.

7.4 Validation Experiments

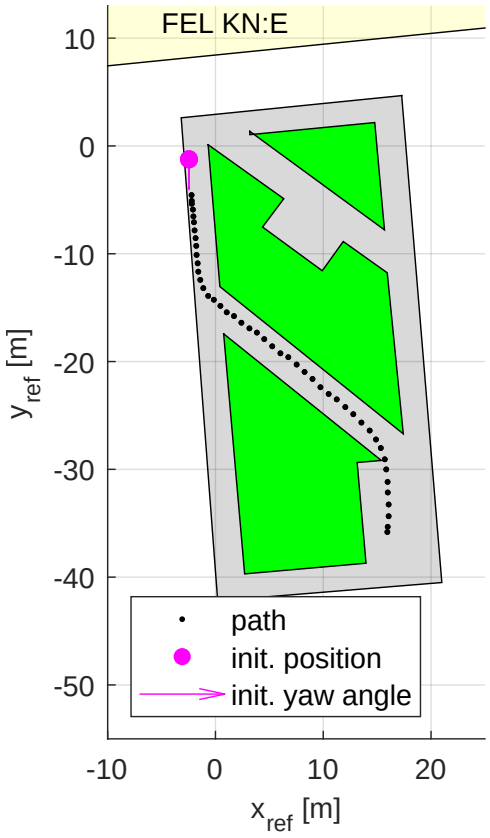


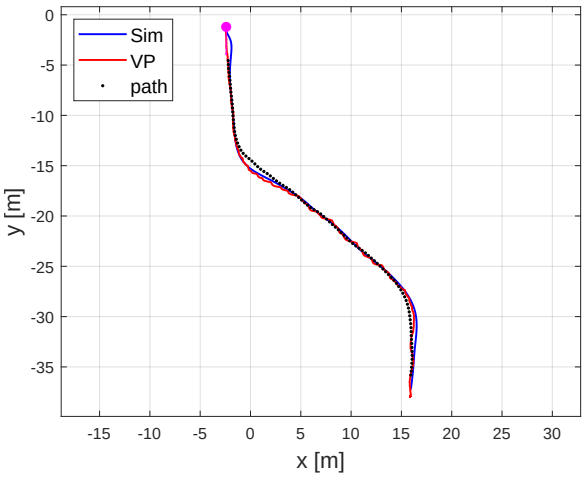
Figure 7.13: Validation scenario for the HA controller. [18]

Validation Scenario

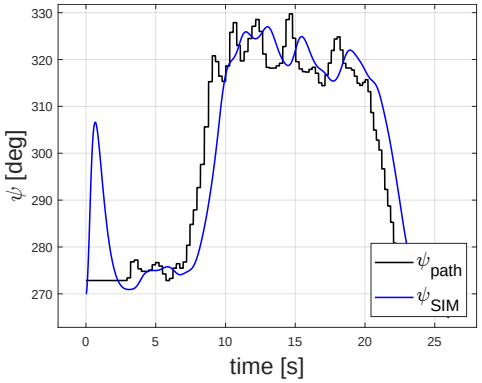
The scenario for the validation experiment is made of two approximately 45 degrees turns. The reference path was first completed by a vehicle without an active control system, controlled by an operator with remote control. Several samples were then selected from the GPS data to form the reference path. This reference path is located in front of the FEL KN:E building, as shown in figure 7.13.

Figure 7.14 shows the comparison between the simulated data and data from the validation platform (VP). The reference velocity was controlled externally by the remote control. The vehicle's velocity data was then used as a parameter v_c of the lateral vehicle's model in the simulation framework, so the simulated and VP data are compared at the same velocity of the vehicle.

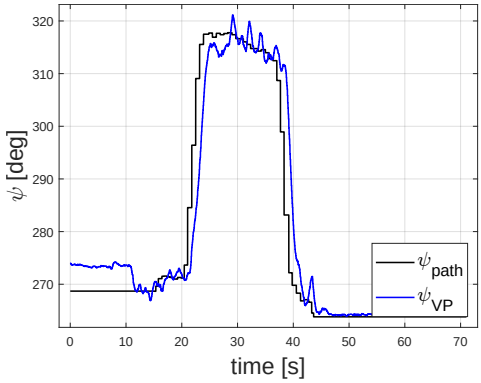
In figure 7.14e, the reference steering angles of the validation platform are shown. These are very abrupt changes that oscillate between mechanical saturation values. The simulation data do not have this oscillatory character. Unfortunately, data with actual wheel steering angles are not available.



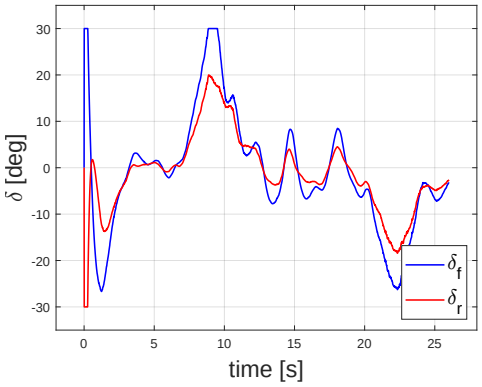
(a) : XY plot.



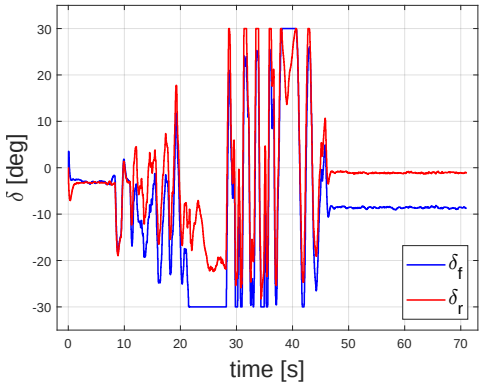
(b) : Plot of the yaw angles.



(c) : Plot of the yaw angles.



(d) : Plot of the steering angles - HA controller in simulation.



(e) : Plot of the steering angles - HA controller on validation platform.

Figure 7.14: The validation simulation.

Chapter 8
Conclusion

The main goal of this thesis was to design and evaluate a path-tracking algorithm for over-actuated vehicles. The linear single-track model was used for the model-based design and the simulation of several driving scenarios. A simulation framework was developed to test the designed algorithms and compare the performance and driving comfort levels. The designed algorithm was then validated on a validation platform.

8.1 Results

In the first part of this thesis, various mathematical models of the vehicle and their derivation were discussed. A linear single-track model was chosen for the design of the control system due to the possibility of constructing a linear state-space model and a relatively accurate description of the fundamental lateral dynamics of the vehicle.

In the second part of this thesis, longitudinal and lateral dynamics controllers were designed. A lateral dynamics controller with a hierarchical architecture was designed using the linear state-space model. First, two virtual control actions presented a solution to the over-actuation problem. Then, several control algorithms were presented for different driving scenarios. The primary design was extended with a look-ahead component of the reference path and a cross-track error saturation reduction method.

In the third part, the designed algorithms were compared with each other and with the lateral Stanley-inspired controller, which was taken as a proven reference controller with excellent results. The results showed a significant improvement in controller performance when using the look-ahead component of the reference path with little change in driving comfort. The simulation data of the Stanley-inspired controller confirmed these results. Overall, the results of the Stanley controller and the HA controller were very similar in terms of performance; the HA with the added look-ahead component had even smaller maximal and RMS values of cross-track error in the given scenarios. However, the HA paid the price with its greater aggressiveness of the control action interventions with worse driving comfort levels.

Finally, the proposed algorithm was integrated into the validation platform. The validation experiment served as a validation of the algorithm's func-

tionality on real hardware. The results were comparable to the simulation data.

In some driving scenarios, oscillations in the steering angles were evident. This may reduce the driving comfort for passengers and mechanically wear out the actuators of the wheels. The simulation data suggested that these oscillations were generated by the cross-track controller. Thus, this problem could be solved by harder damping of this controller or slowing its response to produce smaller control action values. The wheel oscillation problem could also be solved by adding another control loop to control the control action intervention. However, this solution could not be implemented in a validation platform because of the inability to measure the actual steering angle of the wheels, only the reference signal. Thus, the control action interventions could only be modeled with the reference shaping methods.

8.2 Future Work

8.2.1 Nonlinear Single-Track Model

The model could serve as an additional intermediate step in algorithm testing. Algorithms developed using the linear model would be further validated on the non-linear model before testing on the validation platform.

8.2.2 Difference in Road Surface

Road surface information could be added to the reference signal. By a neural network-based algorithm, the image from the camera of the validation platform is processed and differentiated between several types of surfaces. The information about the slipperiness of the surface could influence the control action interventions of the longitudinal and lateral controller.

Furthermore, the vehicle model could be extended to a twin-track model on which a torque vectoring method could be designed.

8.2.3 External Forces

The external forces in the x, y, and z axis in the vehicle coordinate frame could be added to the reference signal as disturbances. This would further expand the simulation framework and provide testing for greater system robustness.

8.2.4 State Feedback

Using the single-track error model, state feedback could be proposed as another path-tracking algorithm.

Appendix A

Bibliography

[1] Gabriel M. Hoffmann, Claire J. Tomlin, Michael Montemerlo, and Sebastian Thrun. "Autonomous Automobile Trajectory Tracking for Off-Road Driving: Controller Design, Experimental Validation and Racing". In: *Stanford University* (2007).

[2] Rahul Kala and Kevin Warwick. "Multi-Level Planning for Semi-autonomous Vehicles in Traffic Scenarios Based on Separation Maximization". In: *Journal of Intelligent and Robotic Systems* (2013).

[3] Dave Ferguson, Thomas M. Howard, and Maxim Likhachev. "Motion Planning in Urban Environments". In: *Journal of Field Robotics* (2008).

[4] Steven M. LaValle. *Planning Algorithms*. Cambridge University Press, 2006.

[5] Sertac Karaman and Emilio Frazzoli. "Sampling-based Algorithms for Optimal Motion Planning". In: *University of California, Berkeley* (2019).

[6] David Stavens, Gabriel Hoffmann, and Sebastian Thrun. "Online Speed Adaptation using Supervied Learning for High-Speed, Off-Road Autonomous Driving". In: (2007).

[7] R. Craig Coulter. "Implementation of the Pure Pursuit Path Tracking Algorithm". In: (1992).

[8] Yiyawng Wu, Yhijiang Xie, and Ye Lu. "Steering Wheel AGV Path Tracking Control Based on Improved Pure Pursuit Model". In: *Journal of Phzsics: Conference Series* 012005.2093 (2021).

[9] Tomáš Werner. *Optimalizace*. Katedra kybernetika, Fakulta elektrotechnická, České vysoké učení technické, 2022.

[10] Reza N. Jazar. *Vehicle Dynamics: Theory and Application*. Springer, 2008.

[11] Rajesh Rajamani. *Vehicle Dynamics and Control*. Springer, 2012.

[12] tesla.com. *Tesla - Model S Owner's Manual*. URL: `https://www.tesla.com/ownersmanual/models/en_us/GUID-91E5877F-3CD2-4B3B-B2B8-B5DB4A6C0A05.html`.

[13] porsche.com. *Porsche - models - 718 Cayman*. URL: https://www.porsche.com/international/models/718/718-models/718-cayman/.

[14] bmw.co.uk. *BMW - THE BMW 1 SERIES*. URL: https://www.bmw.co.uk/en/all-models/1-series.html.

[15] M. Veneri and M. Massaro. "The effect of Ackermann steering on the performance of race cars". In: *Vehicle System Dynamics* (2020).

[16] Dieter Schramm, Manfred Hiller, and Roberto Bardini. *Vehicle Dynamics: Modeling and Simulation*. Springer, 2018.

[17] Hans B. Pacejka. *Tyre and Vehicle Dynamics*. Ed. by Butterworth-Heinemann. Elsevier, 2006.

[18] Jan Švancar. "Autonomous vehicle trajectory tracking algorithms". MA thesis. Czech technical university in Prague, Faculty of Electrical Engineering, Department of Control Engineering, 2023.

[19] Jason Kong, Mark Pfeiffer, Georg Schildbach, and Francesco Borrelli. "Kinematic and Dynamics Vehicle Models for Autonomous Driving Control Design". In: *IEEE Intelligent Vehicles Symposium* IV (2015).

[20] Petr Kulhánek. *Fyzika 1*. FEL ČVUT Praha, 2023.

[21] Rashid Mehmood. "Coarse Atitude Determination Algorithms Using Signal to Noise Ratio of GPS Satellites Signals". MA thesis. Institute of Space Technology (IST), 2015.

[22] Micheal Šebek and Martin Hromčík. "Steady state behaviour. Reference tracking and disturbance rejection." Automatic control, Czech Technical University in Prague, Faculty of Electrical Engineering, Department of of Control Engineering. 2012.