

POLITECNICO DI TORINO

DIMEAS - DEPARTMENT OF MECHANICAL AND AEROSPACE  
ENGINEERING

MASTER OF SCIENCE DEGREE  
IN  
AUTOMOTIVE ENGINEERING

Master's degree thesis

**Combined lateral and longitudinal  
control for autonomous driving  
based on Model Predictive Control**



**Supervisors:**

Prof. Nicola Amati  
Prof. Andrea Tonoli  
Dr. Angelo Bonfitto

**Candidate:**

Irfan Khan  
237701

Academic Year 2018-2019



## Abstract

Autonomous ground vehicles, as an important part of intelligent transportation system, are attracting more attention than ever before. Their control system usually consists of three modules: environment perception, planning and decision-making, and vehicle control. Vehicle control is one of the most critical part of the whole architecture, as it is responsible for the vehicle guidance considering both safety and comfort. In general, control can be divided into lateral control and longitudinal velocity control. Their inter coordination leads to autonomous vehicle motion.

This thesis is focused on the development of a combined lateral and longitudinal controller for autonomous driving based on Model Predictive Control (MPC). The proposed strategy utilizes an adaptive MPC to perform lateral guidance and speed regulation by acting on the front wheel steering angle and acceleration/deceleration to minimize the vehicle's lateral deviation and relative yaw angle with respect to the reference trajectory, while driving the vehicle at the maximum acceptable longitudinal speed.

The technique exploits a stereo camera that utilizes the synthetic data coming from the simulated scenario for lane detection and reference trajectory generation i.e. center line of the lane, to perform the lateral guidance. Longitudinal control strategy is realized with a reference speed generator, which calculates the maximum speed by previewing the path ahead of the vehicle and stability of the vehicle at the same time. The proposed controller is tested with three different scenarios: highway, interurban and urban driving to check the performance at different speeds and varying environment. Dynamics of the vehicle is modeled using a 3 degree of freedom rigid vehicle model, while the internal plant model for MPC is modeled using a linear bicycle model.

The overall system has been developed using MATLAB®, Simulink®, Model Predictive Control Toolbox™ and Automated Driving System Toolbox™. In particular, scenarios were generated using the Scenario Designer application with the Automated Driving System Toolbox that allows to develop and test Advanced Driver Assistance Systems (ADAS) and autonomous driving systems providing computer vision algorithms and generating synthetic data.

## **Acknowledgements**

I would like to express my great gratitude to Prof. Nicola Amati, supervisor of this thesis, for allowing me to carry out this project and supporting me with his suggestions and availability. I would also like to thank Prof. Andrea Tonoli and Dr. Angelo Bonfitto for the help and valuable suggestions during the thesis work.

A big thank to Dr. Stefano Feraco for his friendship, professionalism, and enormous patience demonstrated daily during this "journey".

A special thank to my mother and father, who made so many sacrifices and allowed me to reach this goal. During the long course of the studies, they have always supported me and above all endured, making me grow and become the person I am today.

Thanks to my best friend Fanny who has always been by my side. With her constant support, but most of all with her gentleness and confidence in me, she continually urged me to always give my best.

Thanks to all the colleagues of the Politecnico, who accompanied me during these years, Johnny, Luis, Carlo, Andrea, and in particular Kiran with whom I lived the joys and sorrows of this university period.

I thank the friends who were always there for me, Usama, Chirag, Vishal, Abhishek and Zubair, making this journey easier.

Finally, but not least, I want to thank my "second family" Andrea, Alessia, Marta, Pietro and Elena who made me feel at home all the time.

# Contents

<b>List of Figures</b>	III
<b>1 Introduction</b>	1
1.1 Thesis motivation . . . . .	1
1.2 Classification of autonomous vehicles . . . . .	3
1.3 State of the art . . . . .	4
1.4 Thesis outline . . . . .	10
<b>2 Modelling</b>	11
2.1 Vehicle model for validation and simulation . . . . .	11
2.1.1 Kinematic model . . . . .	12
2.1.2 Dynamic model . . . . .	15
2.2 Tire models . . . . .	17
2.2.1 Pacejka tire model . . . . .	17
2.2.2 Linear tire model . . . . .	18
2.3 Vehicle model for MPC . . . . .	20
2.4 Driveline dynamics . . . . .	23
<b>3 Control design</b>	25
3.1 Perception . . . . .	27
3.1.1 Lane detection . . . . .	27
3.2 Reference trajectory generation . . . . .	31
3.2.1 Trajectory curvature computation . . . . .	32
3.2.2 Computation of vehicle model dynamic parameters . . . . .	35
3.3 Reference speed profile generation . . . . .	37
3.4 Model Predictive Control . . . . .	40

3.4.1	Overview of MPC . . . . .	40
3.4.2	MPC problem formulation . . . . .	44
<b>4</b>	<b>Results and discussions</b>	<b>50</b>
4.1	Driving scenarios . . . . .	50
4.2	Results and discussion . . . . .	54
4.2.1	Highway . . . . .	54
4.2.2	Inter-urban . . . . .	58
4.2.3	Urban . . . . .	62
4.3	Comparision with another controller based on MPC and PID . . . . .	66
<b>5</b>	<b>Conclusions and future works</b>	<b>69</b>
	<b>Bibliography</b>	<b>71</b>

# List of Figures

1.1	SAE's classification of autonomous vehicles . . . . .	3
1.2	Control strategy presented in [6] . . . . .	6
1.3	Control strategy presented in [12] . . . . .	8
1.4	Global architecture of control strategy for autonomous driving presented in this thesis . . . . .	9
2.1	Vehicle kinematic model . . . . .	12
2.2	3 DoF rigid vehicle model . . . . .	16
2.3	Linearized lateral tire forces in small slip angle region compared to Pacejka model . . . . .	19
2.4	Tire slip angle . . . . .	19
2.5	Bicycle model in terms of lateral deviation and relative yaw angle with respect to the center line of the lane . . . . .	21
2.6	Driveline dynamics architecture . . . . .	24
3.1	Detailed architecture of the control strategy . . . . .	26
3.2	Vehicle with camera location . . . . .	28
3.3	Focal length description . . . . .	28
3.4	Lane line feature extraction . . . . .	30
3.5	Lane line model . . . . .	31
3.6	Curve $\alpha$ and tangential angle $\phi$ . . . . .	33
3.7	Demonstration that the definition 3.1 can be derived from the definition 3.2 . . . . .	34
3.8	Osculating circle and radius of curvature . . . . .	35
3.9	Definition of lateral deviation and relative yaw angle with respect the center line of the lane . . . . .	36

3.10	Center line, curvature, lateral deviation and relative yaw angle computation . . . . .	36
3.11	Block diagram for Model Predictive Control . . . . .	41
3.12	Basic concept for Model Predictive Control . . . . .	43
4.1	(a)Highway driving scenario: S is the vehicle's starting point. F is the end of the road track. (b) Detected road curvature $k$ . . . . .	52
4.2	(a)Inter urban driving scenario: S is the vehicle's starting point. F is the end of the road. (b) Detected road curvature $k$ . . . . .	52
4.3	(a)Urban driving scenario: S is the vehicle's starting point. F is the end of the road. (b) Detected road curvature $k$ . . . . .	53
4.4	Measured vehicle's longitudinal speed $V_x$ (solid) vs. vehicle's longitudinal speed reference $V_{ref}$ (dashed) . . . . .	54
4.5	Lateral deviation $e_1$ . . . . .	55
4.6	Relative yaw angle $e_2$ . . . . .	55
4.7	Longitudinal acceleration command $a_x$ . . . . .	56
4.8	Front wheels steering angle command $\delta$ . . . . .	56
4.9	GG plot with the ellipse representing the adherence limits . . . . .	57
4.10	Measured vehicle's longitudinal speed $V_x$ (solid) vs. vehicle's longitudinal speed reference $V_{ref}$ (dashed) . . . . .	58
4.11	Lateral deviation $e_1$ . . . . .	59
4.12	Relative yaw angle $e_2$ . . . . .	59
4.13	Longitudinal acceleration command $a_x$ . . . . .	60
4.14	Front wheels steering angle command $\delta$ . . . . .	60
4.15	GG plot with the ellipse representing the adherence limits . . . . .	61
4.16	Measured vehicle's longitudinal speed $V_x$ (solid) vs. vehicle's longitudinal speed reference $V_{ref}$ (dashed) . . . . .	62
4.17	Lateral deviation $e_1$ . . . . .	63
4.18	Relative yaw angle $e_2$ . . . . .	63
4.19	Longitudinal acceleration command $a_x$ . . . . .	64
4.20	Front wheels steering angle command $\delta$ . . . . .	64
4.21	GG plot with the ellipse representing the adherence limits . . . . .	65
4.22	(a)Highway exit driving scenario: S is the vehicle's starting point. F is the end of the road track. (b) Detected road curvature $k$ . . . . .	66

4.23	Results comparison:(a), (c) and (e) depicts the result of the combined MPC controller, while (b), (d) and (f) depicts the results for decoupled controller based on MPC and PID in terms of $V_x$ (solid) vs. vehicle's longitudinal speed reference $V_{ref}$ (dashed), Lateral deviation $e_1$ and Relative yaw angle $e_2$ . . . . .	67
4.24	Results comparison:(a), (c) and (e) depicts the result of the combined MPC controller, while (b), (d) and (f) depicts the results for decoupled controller based on MPC and PID in terms of vehicle side slip angle $\beta$ with $\beta_{limit}$ , Front wheels steering angle command $\delta$ and GG plot with the ellipse representing the adherence limits . . . . .	68

# Chapter 1

## Introduction

### 1.1 Thesis motivation

Every year 1.25 million people die and as many as 50 million are injured in road traffic accidents worldwide, according to United Nations statistics. Human error is involved in about 95% of all road traffic accidents in the EU, and in 2017 alone, 25,300 people died on the Union's roads [1]. Every year children, the elderly or the physically challenged have little or no access to individual mobility. And every year half a million metric tons of CO<sub>2</sub> emissions in Germany alone could be saved by eliminating the endless search for a parking space, which studies show account for up to 30% of inner city traffic. Driverless cars can drastically reduce these figures and improve road safety, while new digital technologies can also reduce traffic congestion and emissions of greenhouse gases and air pollutants. Mobility can also be improved, for example by opening up road transport to the elderly and those with reduced mobility or disabilities. For this reason, in the last few years many university researches and car companies are focusing on the development of Advanced Driver Assistance Systems (ADAS) and self-driving vehicles.

According to European Road Safety Observatory (ERSO) [2], ADAS can be defined as: “*vehicle-based intelligent safety systems which could improve road safety in terms of crash avoidance, crash severity mitigation and protection and post-crash phases. ADAS can, indeed, be defined as integrated in-vehicle or infrastructure based systems which contribute to more than one of these crash-phases*”.

Nowadays, a number of ADAS are produced by carmakers and available in automobiles. More than driver assistance, ongoing research and development in the automotive field are oriented to driverless cars with the design of systems for partially/fully automated driving, with several commercial entities pushing the bounds alongside academia. Google has perhaps the most experience in the area, having tested its fleet of autonomous vehicles for more than 2 million miles, with expectation to soon launch a pilot MoD service project using 100 self-driving vehicles. Tesla is early to market their work, having already provided an autopilot feature in their 2016 Model S cars. Uber’s mobility service has grown to upset the taxi markets in numerous cities worldwide, and has furthermore recently indicated plans to eventually replace all their human driven fleet with self-driving cars, with their first self-driving vehicle pilot program already underway [1].

Autonomous driving cars can be considered vehicles that perform the transportation task without the human intervention, using algorithms executed by an on-board computer to simulate the behaviour of the driver and make decision. The core competencies of an autonomous vehicle software system can be broadly categorized into three categories, namely environment perception, planning and decision-making, and vehicle control. The environment perception module obtains information on surroundings by external sensors, such as lasers, cameras and radar, and then fuses the information by building environment maps to determine drivable surfaces. The planning and decision-making module gathers and handles task information and combines it with vehicle states and drivable surfaces information to determine the desired path and the speed profile. The vehicle control module coordinates the engine, brakes and steering to track the desired path and speed by providing the signal necessary for actuators to guide the autonomous vehicle.

Recently, autonomous vehicle technology attracts automotive industry due to its potential applications such as automated highways, urban transportation, etc. However, fully automated driving remains a complex task which involves challenging aspects and requires skills in domains such as vision and image processing, trajectory generation and path planning, modeling and automatic control. The latter problem is of a paramount importance for vehicle guidance, i.e. steering and velocity control. As will be shown state of the art, the steering and velocity tracking problems are considered separately or in a coupled way in the literature.

## 1.2 Classification of autonomous vehicles

The SAE International (Society of Automotive Engineers) established in 2014 a classification system to describe the progression of the automation of vehicles, as shown in Figure 1.1. The United Nations and the US Department of Transportation have adopted SAE J3016 guidelines [3], that is today considered the industry standard.

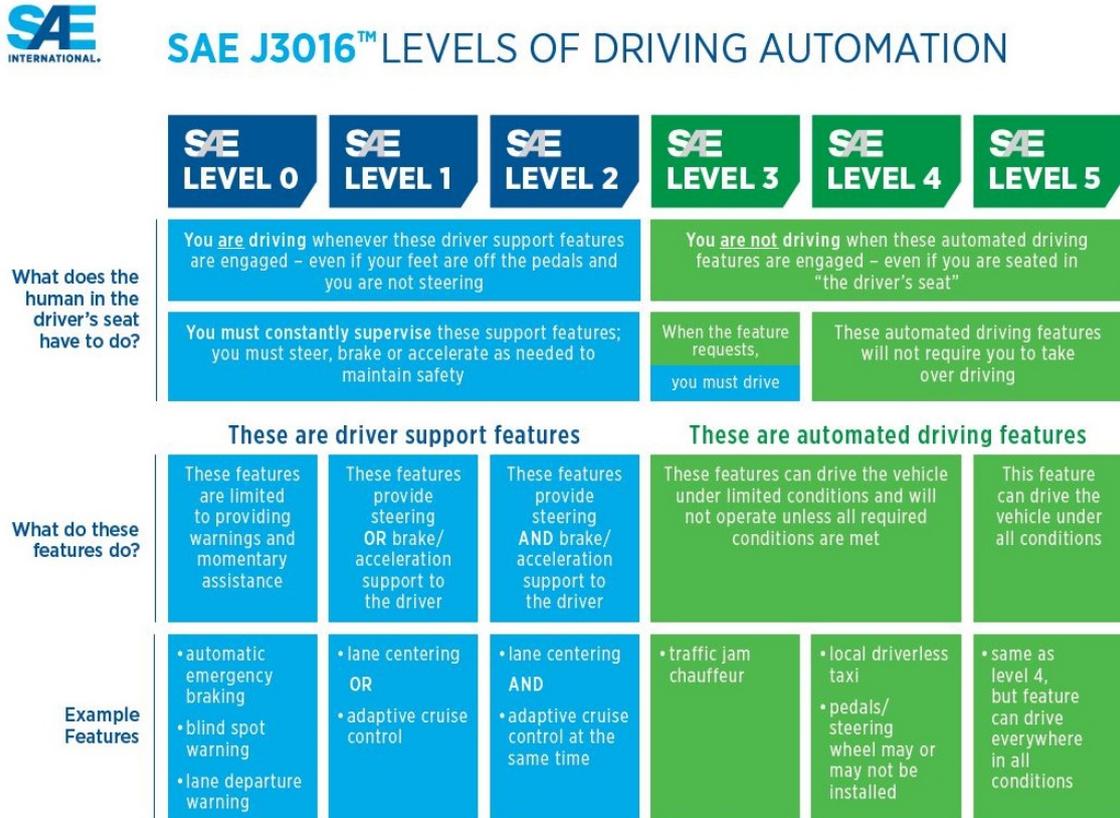


Figure 1.1: SAE’s classification of autonomous vehicles

The classification is based on the amount of responsibility and attentiveness required by the driver. Six levels are defined accordingly, from a situation in which everything is controlled by the human (Level 0) to the full automation of the vehicle under all driving conditions (Level 5). The definition of each level takes into account the specific role played by the driver, the driving automation system and by other vehicle systems and components that might be present.

SAE’s levels are descriptive and informative rather than normative, and technical rather than legal, they clarify the role of the ADS which are progressively included

in the vehicles. ADS is the acronym of Automated Driving Systems. It refers to both hardware and software tools collectively capable of performing dynamic driving tasks (e.g. driving environment monitoring, longitudinal and lateral motion control, maneuver planning).

The six levels can be defined as:

- *Level 0 - No automation*: steering or speed control may be momentarily assisted by the vehicle, but the human driver is in charge of all the aspects of driving;
- *Level 1 - Driver assistance*: longitudinal or lateral support under well-defined driving scenarios (e.g. highway) are guaranteed, because the vehicle takes over either the speed or the steering control on a sustained basis;
- *Level 2 - Partial automation*: both speed and steering control are taken over by the vehicle, therefore continuous longitudinal and lateral support under well-defined driving scenarios are guaranteed. A Level 2 vehicle is equipped with a wider set of ADAS;
- *Level 3 - Conditional automation*: the vehicle becomes capable of taking full control under well-defined driving scenarios, but the driver must be always in the condition of suddenly taking back control when required by the system;
- *Level 4 - High automation*: human interaction is not needed anymore, the vehicle takes full control and complete a journey in full autonomy under limited driving scenarios. Pedals and steering wheel are likely to be still present to guarantee the possibility to drive in scenarios that go beyond the defined uses cases (e.g off-road);
- *Level 5 - Full automation*: the vehicle takes full control under all driving scenarios, no more provisions for human control are present. The concept of journey will be disruptively innovated, the entire vehicle design revolutionized.

### 1.3 State of the art

In recent years a lot of researches have been focused on autonomous driving and, for this reason, an overview of the existing projects has been done at the beginning

of this work. In particular, the state of the art regarding the development of the vehicle control for autonomous driving has been analysed.

Vehicle control is one of the most critical part of the autonomous driving vehicle architecture, as it is responsible for the vehicle guidance considering both safety and comfort. To improve handling performance and safety of vehicle, a considerable number of advanced driver assistance systems (ADAS) for vehicle lateral dynamics and longitudinal collision have been developed and utilized commercially. Lane keep assist (LKA) is one of the lateral control system that is commonly implemented on-board in the autonomous vehicles, which automatically takes the lead on the car to ensure it stays in its lane. Therefore, the LKA can be defined as a path tracking problem.

Many researchers have reported the work on lateral control strategies. In [4], it is presented the state of the art and challenges for vehicle modeling and the review of the control strategies in path tracking control.

An adaptive Model Predictive Control is presented in [5]. In this work the longitudinal velocity is assumed constant. For this reason, the work is focused only on the development of the lateral control. The goal of MPC controller consists in the minimization of the lateral deviation from the center line and the steady state yaw angle error, while satisfying respective safety constraints. These constraints refer to the steering angle offset present in the steering system. In order to estimate and adapt in real-time the maximum possible bound of the steering angle offset from data, they use a robust Set Membership Method based approach. The results of this control show that is well-suited for scenarios with sharp curvatures on high speed. The work in [6] is focused on the realization of a controller to implement a lane keeping system using Model Predictive Control (MPC) theory. Figure 1.2 shows how the controller is developed in this research. The output is the optimal steering angle of the front wheel computed minimizing the cost function of the MPC controller. The cost function is composed of the steering angles and the error between the reference and the predictive trajectory. The generation of the reference trajectory is performed fitting five preview points coming from sensors. In order to demonstrate the effectiveness and robustness of the approach, a co-simulation of MATLAB/Simulink and CarSim has been executed.

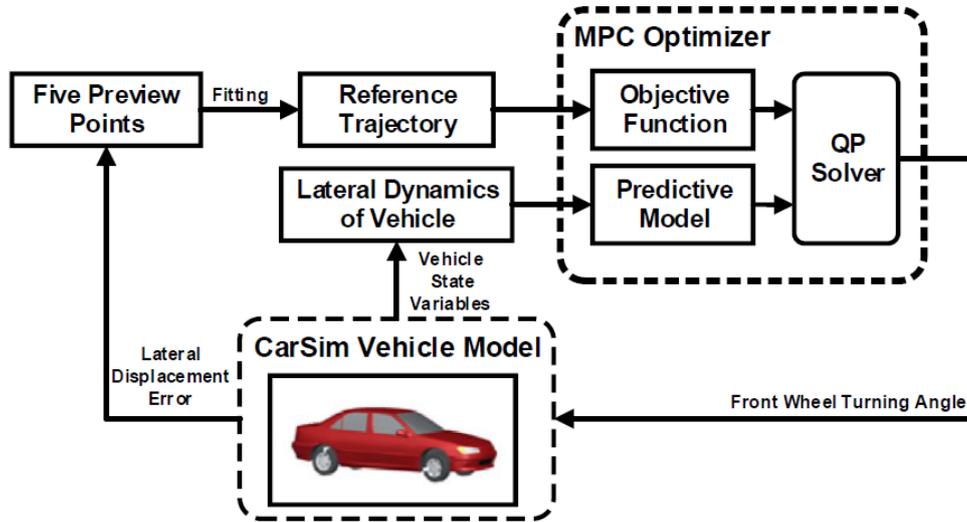


Figure 1.2: Control strategy presented in [6]

A linear MPC controller that realizes a lane keeping and an obstacle avoidance systems for low curvature roads has been presented in [7]. The control developed in this work has been divided in two successive stages: the first stage computes a braking or throttle profiles based on the prediction horizon; the second stage realizes the MPC using the linear time-varying models of the vehicle lateral dynamics derived by the profiles of the first stage. The MPC estimates the steering angle command based on the optimal breaking or throttle command.

The speed tracking task is also relevant in fully automated driving. Now a days, the cruise controller (CC) is widely used to ensure vehicle speed regulation. An extension of the CC is the Adaptive CC (ACC) which employs external information for regulation of both vehicle speed and inter vehicular distance. So, the longitudinal planning is mainly responsible for calculating the desired velocity or acceleration command of an autonomous vehicle according to its surrounding environment information.

An interesting review of the development of adaptive cruise control systems is presented in [8]. An ACC design for traffic jam based on MPC technique is proposed and experimentally validated in [9]. It is mainly focused on decreasing the computational load for the practical use of MPC by using low-order prediction model. The work in [10] presents the design of a parameterized ACC based on explicit model predictive control. It uses only a few design parameters, i.e. tuning knobs, that are

directly related to the key characteristics of the behavior of the ACC to adapt to different desirable driving behavior. So, the main goal of this work is to make the ACC driver dependent.

In the above studies, lateral and longitudinal control problems have been investigated in a decoupled way. In fact, numerous studies dealing with the lateral guidance of automotive vehicles are based on the assumption of a constant speed. They focus only on how to eliminate the path error according to desired path and how to improve the robustness against the uncertainties but does not take into account the longitudinal dynamics. On the other hand, those dealing with longitudinal control do not take account of the coupling with the lateral motion. However, there are strong couplings between the two dynamics at several levels: dynamic, kinematic and tyre forces. Consequently, the simultaneous inclusion of longitudinal and lateral control becomes unavoidable in order to improve performance guidance in a large operating range. Nevertheless, the control design based on a complex mathematical model of the vehicle becomes a difficult task due to these couplings. Therefore, different control approaches have been proposed in the literature to cope with this interesting problem.

For example, in [11], a coupled lateral and longitudinal control strategy based on Linear time-varying Model predictive control (LTV-MPC) is presented. However, it is designed mainly for collision avoidance exploiting the handling limits but does not consider real world scenarios application. Also, only negative accelerations are considered because the control is designed for collision avoidance by following an evasion trajectory computed explicitly.

A further coupled method is presented in [12], where the lateral dynamics control is designed with a non-linear Model Predictive Control (NLMP) and the longitudinal control task is addressed with Lyapunov-based synthesis algorithm, as shown in Figure 1.3. This study focuses more on the fuel consumption reduction by a finely tuned longitudinal control. However, as stated by the authors, the application of this technique in real time is limited due to calculation time and may result into a computationally demanding method.

Similarly, another coupled method is presented in [13], where the simultaneous lateral and longitudinal control for a full drive by wire autonomous vehicle is designed with a non-linear Model Predictive Control (NLMP). Non linear prediction model

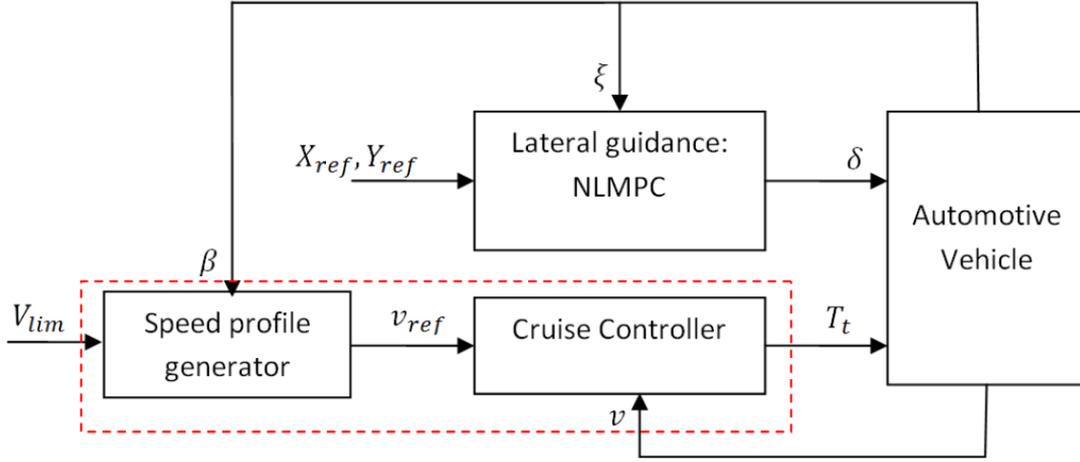


Figure 1.3: Control strategy presented in [12]

utilizes a spatial transformation to derive the dynamics of the vehicle about the reference trajectory. The motion of the vehicle is controlled using three control efforts target longitudinal and lateral forces and target yaw moment. The control is only tested for an emergency double lane change maneuver. So, its effectiveness in wide range of operating conditions is still a doubt. However, computational complexity of NLMPC remains a major problem in Automotive industry application. Parallel advances in theory and computing systems have enlarged the range of applications where real-time MPC can be applied. Yet, for a wide class of “fast” applications the computational burden of MPC is still a serious barrier for its implementation. Nevertheless, the capability of handling constraints in a systematic way makes MPC a very attractive control technique, especially for applications where the process is required to work in wide operating regions and close to the boundary of the set of admissible states and inputs. This has motivated the study of alternative MPC approaches, requiring the solution of simpler optimization problems in real-time. Most of these approaches are based on linear or piecewise-linear approximations of the nonlinear model of the plant.

This thesis presents a combined lateral and longitudinal controller for autonomous driving based on an adaptive MPC is proposed. The proposed control strategy maximizes the longitudinal speed while remaining in constrained speed range and

without exceeding the adherence condition. At the same time, it eliminates the path error between the actual location and the desired path in terms of lateral deviation and desired yaw angle, assuring the handling stability during the motion. The command signals generated by MPC and provided to the vehicle are the front wheel steering angle and actuation of throttle/brake pedals.

The overall autonomous driving system has been implemented with MATLAB and Simulink<sup>1</sup>. The technique exploits a simulated stereo camera that utilizes the synthetic data coming from the simulated driving scenario for lane detection, as shown in Figure 1.4. In the real implementation, this information is obtained from a lane detection algorithm based on the real-time streaming of a stereo-camera data. Since, in our case the simulations are conducted in MATLAB/Simulink, the lane boundaries information is extracted from three simulated driving scenarios, reproducing a highway, inter-urban road and urban road. Dynamics of the vehicle is modeled using a 3 degree of freedom rigid vehicle model.

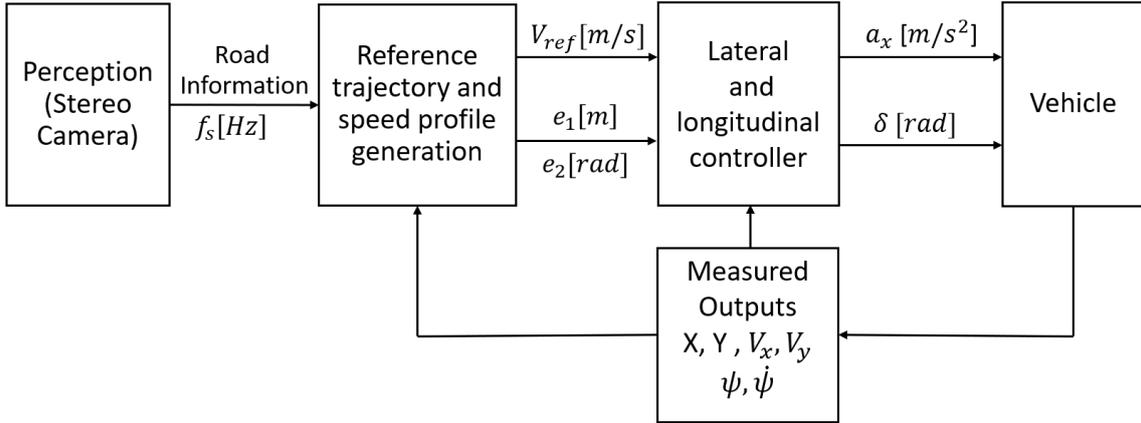


Figure 1.4: Global architecture of control strategy for autonomous driving presented in this thesis

<sup>1</sup><https://it.mathworks.com/>

## 1.4 Thesis outline

The thesis is organized as follows:

- *Chapter 2*: It presents the vehicle modeling used for the validation and control synthesis. In particular, a 3 degree of freedom rigid vehicle model, tire model and a linearized vehicle model for MPC control design are discussed in detail.
- *Chapter 3*: First an overview of the overall control strategy is presented. Later based on the framework of MPC, a predictive optimization problem is formulated and solved and the MPC implemented for this work is explained in detail.
- *Chapter 4*: Three different scenarios are presented to evaluate the performance of the controller by means of simulations. The results are presented and discussed.
- *Chapter 5*: In the final chapter conclusions and future works are reported.

# Chapter 2

## Modelling

This chapter presents the mathematical modelling of the the vehicle dynamics for the validation and the control synthesis, referring to the overall model (Figure 1.4). Model-based control is highly affected by the quality of the models provided. On the one hand, accurate models are typically computationally expensive and provide accurate predictions. On the other hand, simple models are less computationally demanding, but provide less accurate predictions. Since the MPC will be evaluated through simulations, a validation model is needed. The validation model needs to well describe the behaviour of a real vehicle. So, for this thesis, dynamics of the vehicle is modeled using the 3 degree of freedom rigid vehicle model (Single Track), which is imported from Vehicle dynamics Blockset in Simulink®. The derivation of the kinematic and the dynamic vehicle model is described in Section 2.1.1 and 2.1.2. For this thesis we used a linearized dynamic model of the vehicle, derived in Section 2.2, as the prediction model.

### 2.1 Vehicle model for validation and simulation

In this section, both kinematic and dynamic models of the vehicle are presented with their assumptions and constraints.

### 2.1.1 Kinematic model

Kinematics is a branch of classical mechanics that explains the motion of points, bodies and groups of objects without considering the forces that affect the motion. The equations of motion described by a kinematic model refer purely to geometric relationships that control the system, for this reason kinematics is often called the “geometry of motion” in field of study [14].

The beginning of a kinematics problem consists of the geometry description of the system and the declaration of the initial conditions of the values that refer to position, velocity and acceleration of system points.

As shown in Figure 2.1, the following kinematic model of the vehicle has been considered [15].

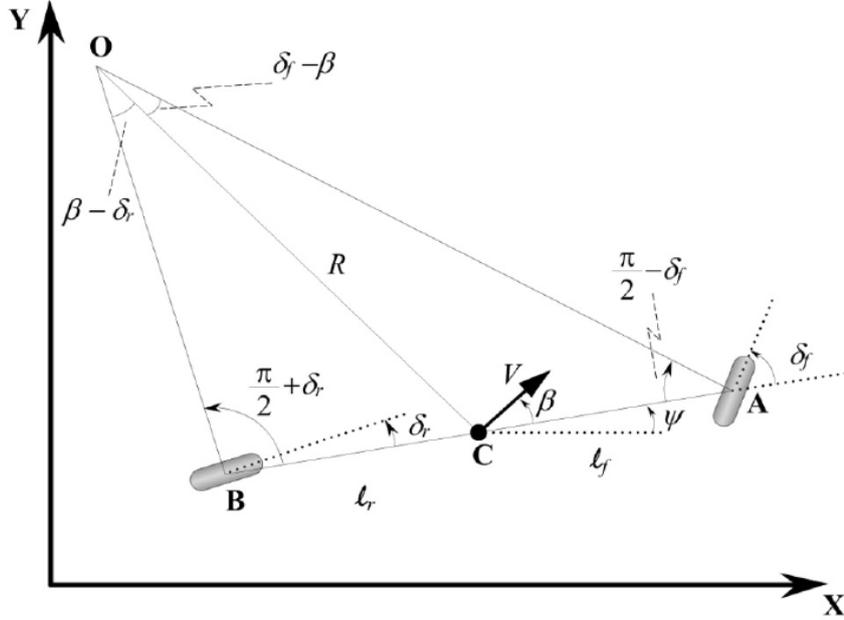


Figure 2.1: Vehicle kinematic model

The image presents a bicycle model in which the two front wheels and the two rear wheels are represented by one single central tires at points A and B, respectively. The steering angle for the front wheel is indicated with  $\delta_f$ , while  $\delta_r$  refers to the steering angles for the rear wheel. In this work, the vehicle model is assumed as a front-wheel-only steering, therefore the rear steering angle  $\delta_r$  is set to zero.

The point C in the figure represents the center of gravity (c.g.) of the vehicle.

The distances from this point to the points A and B are indicated with  $l_f$  and  $l_r$  respectively. The sum of these two terms corresponds to the wheelbase  $L$  of the vehicle:

$$L = l_f + l_r \quad (2.1)$$

Since the vehicle is assumed to have planar motion, three coordinates are necessary to describe the vehicle motion:  $X$ ,  $Y$  and  $\Psi$ . ( $X$ ,  $Y$ ) represent the inertial coordinates of the location of the center of gravity of the vehicle, while  $\Psi$  indicates the orientation of the vehicle and it is called yaw angle. The vector  $V$  in the model refers to the velocity at the c.g. of the vehicle. This vector makes an angle  $\beta$ , called slip angle, with the longitudinal axis of the vehicle.

The point O refers to the instantaneous center of rotation of the vehicle and it is defined by the intersection of lines AO and BO. These two lines are drawn perpendicular to the orientation of the two wheels. The length of the line OC corresponds to the radius of the vehicle trajectory  $R$ , and it is perpendicular to the velocity vector  $V$ .

Applying the sine rule to triangles OCA and OCB, remembering that  $\delta_r$  is equal to zero, it is possible to define the following equations:

$$\frac{\sin(\delta_f - \beta)}{l_f} = \frac{\sin(\frac{\pi}{2} - \delta_f)}{R} \quad (2.2)$$

$$\frac{\sin(\beta)}{l_r} = \frac{1}{R} \quad (2.3)$$

After some manipulation and multiplying by  $\frac{l_f}{\cos(\delta_f)}$ , equation 2.2 becomes:

$$\tan(\delta_f) \cos(\beta) - \sin(\beta) = \frac{l_f}{R} \quad (2.4)$$

Likewise, multiplying by  $l_r$ , equation 2.3 can be re-written as:

$$\sin(\beta) = \frac{l_r}{R} \quad (2.5)$$

Adding equations 2.4 and 2.5, the following relation has been obtained:

$$\tan(\delta_f) \cos(\beta) = \frac{l_f + l_r}{R} \quad (2.6)$$

This formula allows to write the radius  $R$  of the vehicle trajectory as a function of the front steering angle  $\delta_f$ , the slip angle  $\beta$ , and  $l_f$ .

If the value of radius  $R$  changes slowly due to low velocity, the yaw rate  $\dot{\Psi}$  of the vehicle can be assumed equal to the angular velocity  $\omega$  that is defined as:

$$\omega = \frac{V}{R} \quad (2.7)$$

Therefore, the yaw rate  $\dot{\Psi}$  can be described as follows:

$$\dot{\Psi} = \frac{V}{R} \quad (2.8)$$

Using formula 2.6, the equation 2.8 can be re-written as:

$$\dot{\Psi} = \frac{V \cos(\beta)}{l_f + l_r} \tan(\delta_f) \quad (2.9)$$

After all these assumptions, the overall equations of the kinematic model can be defined as:

$$\dot{X} = V \cos(\Psi + \beta) \quad (2.10)$$

$$\dot{Y} = V \sin(\Psi + \beta) \quad (2.11)$$

$$\dot{\Psi} = \frac{V \cos(\beta)}{l_f + l_r} \tan(\delta_f) \quad (2.12)$$

### 2.1.2 Dynamic model

A kinematic model offers satisfactory results when the vehicle speed and steering angle are low enough, but when the speeds increase and the curvatures of the trajectory change in time, it is not possible any more to assume that the velocity vector of each wheel is parallel to the wheel symmetry plane. For this reason, instead of adopting a kinematic model, a vehicle dynamic model is developed.

In this thesis, dynamics of the vehicle is modeled using the 3 degree of freedom rigid vehicle model (Single Track), which is imported from Vehicle dynamics Blockset in Simulink [16]. This model accounts for the two displacements on the plane (longitudinal, depicted by subscript x and lateral depicted by subscript y) and the rotation around an axis normal to the plane (yaw motion). It implements a rigid two axle vehicle body model. So, the two front wheels and the two rear wheels of the vehicle are represented as a single center wheel. As our test vehicle is only steerable from the front wheels, the test vehicle is modeled to be only steerable from the front wheel.

The nomenclature refers to the model depicted in Figure 2.2. We denote by  $F_l$ ,  $F_c$  the longitudinal (or “tractive”) and lateral (or “cornering”) tire forces, respectively,  $F_x$ ,  $F_y$  the longitudinal and lateral forces acting on the vehicle center of gravity,  $F_z$  the normal tire load, X, Y the absolute car position in inertial coordinates,  $l_f$ ,  $l_r$  (distance of front and rear wheels from center of gravity),  $g$  the gravitational constant,  $m$  the car mass,  $I_{zz}$  the car inertia,  $\alpha$  the slip angle,  $\delta$  the wheel steering angle and  $\Psi$  the heading angle. The lower scripts  $f$  and  $r$  particularize a variable at the front wheels and the rear wheels, respectively, e.g.  $F_{lf}$  is the front wheel longitudinal force. Newton Euler equations (2.13), (2.14) denote the longitudinal and lateral momentum with respect to CG in the vehicle reference frame while yaw dynamics are considered by (2.15).

$$m\dot{V}_x = mV_y\dot{\Psi} + F_{xf} + F_{xr} - F_{aero} \quad (2.13)$$

$$m\dot{V}_y = -mV_x\dot{\Psi} + F_{yf} + F_{yr} \quad (2.14)$$

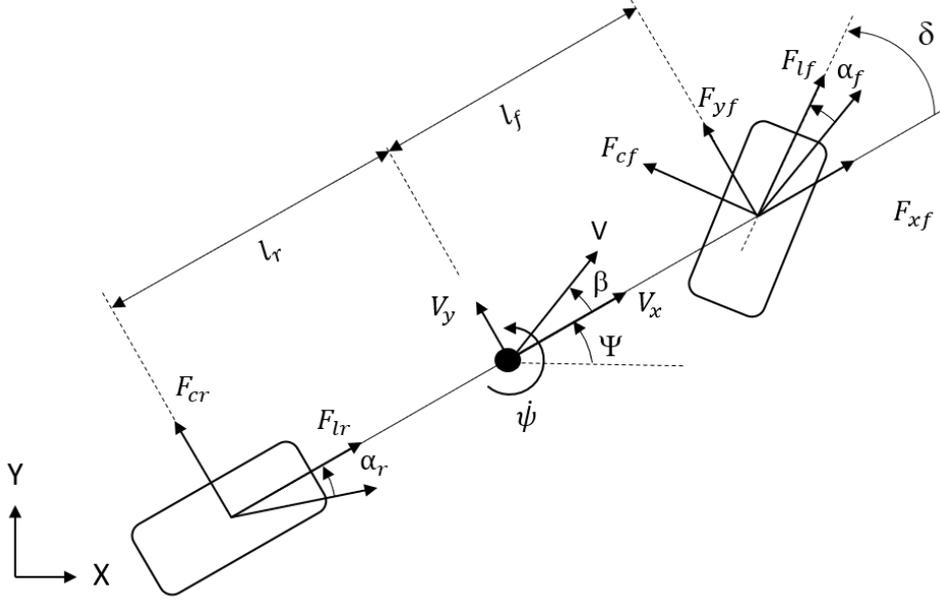


Figure 2.2: 3 DoF rigid vehicle model

$$I_{zz}\ddot{\Psi} = l_f F_{yf} - l_r F_{yr} \quad (2.15)$$

The forces acting on the vehicle center of gravity are related to tires forces and front steering angle  $\delta$  by the given equations.

$$F_{xf} = F_{lf} \cos \delta - F_{cf} \sin \delta \quad (2.16)$$

$$F_{yf} = F_{lf} \sin \delta + F_{cf} \cos \delta \quad (2.17)$$

$$F_{xr} = F_{lr} \quad (2.18)$$

$$F_{yr} = F_{cr} \quad (2.19)$$

## 2.2 Tire models

The tire forces have highly nonlinear behavior when slip ratio or slip angle is large. Thus it is of extreme importance to have a realistic nonlinear tire force model for the vehicle dynamics when operating the vehicle in the tire nonlinear region e.g. during racing. In such situations, large slip ratio and slip angle can happen simultaneously and the longitudinal and lateral dynamics of the vehicle is highly coupled and nonlinear due to the nature of the tire forces. Similar situation can occur even with small inputs when the surface friction coefficient  $\mu$  is small.

When the slip ratio and slip angle are both small, both longitudinal and lateral tire forces show linear behavior and are less coupled e.g. during normal driving. This situation holds true when the vehicle operates with moderate inputs on high  $\mu$  surfaces. In such situations, linearized tire models might serve well in control design, with proper constraints on the slip ratio and slip angle.

In this section, we will present two tire models. First one is a complex nonlinear semi-empirical model capturing the nonlinear and coupling behavior of the tire forces, while second one is a simple linear model. For this thesis we have used linear tire models since, the angles  $\alpha$ ,  $\beta$  and  $\delta$  are limited to small values.

### 2.2.1 Pacejka tire model

This subsection gives more details on the Pacejka tire model [17]. It is a complex nonlinear semi-empirical model being able to describe the nonlinear and coupled behavior of tire forces under wide operation range. Pacejka model describes the tire forces as functions of the tire normal force, slip ratio, slip angle and surface friction coefficient.

It uses a function of following form to fit the experiment data:

$$Y(X) = D \sin(C \arctan(B(1 - E)(X + S_h)) + E \arctan(B(X + S_h))) + S_v \quad (2.20)$$

where  $Y$  is either the longitudinal or lateral tire force.  $X$  is the slip ratio when  $Y$  is the longitudinal force and the slip angle when  $Y$  is the lateral force.  $B$ ,  $C$ ,  $D$ ,  $E$ ,  $S_h$  and  $S_v$  are the parameters fit from the experimental data.

It is critical to have a realistic tire model to take the slip phenomenon into account, specially if the car has to be driven at its limits of handling and used in racing

context. However, fitting a physical tire behaviour to its corresponding parameters value in a Pacejka model can be hard because of the model complexity. But for this thesis, the car is used for normal driving so, a linearized tire model is used, which is explained in the next section.

### 2.2.2 Linear tire model

This subsection presents a linear tire model, which can be used for modeling lateral tire force inside the linear region [18]. This model is only valid when both slip angle and slip ratio are restricted to have small values. With small  $\alpha$  and  $s$  assumption, the lateral tire force is modeled:

$$F_c = C_\alpha(\mu, F_z)\alpha \quad (2.21)$$

where  $C_\alpha$  is called the tire's cornering stiffness coefficient and is a function of the friction coefficient  $\mu$  and normal force  $F_z$ .

Figure 2.3 compares the lateral tire forces computed from the linear tire model and the Pacejka model. The linear tire model is the simplest tire model we can possibly get and should be implemented only with small slip angles, i.e. inside the tire linear region. With these assumptions, the lateral tire forces  $F_{yf}$  and  $F_{yr}$  that act on the vehicle are modelled with the value of the wheel slip angle when it is small. As shown in Figure 2.4, the front wheel slip angle  $\alpha_f$  can be defined as the difference between the steering angle  $\delta$  of the front wheel and the orientation angle of the tire velocity vector  $\theta_{Vf}$  with respect to the longitudinal axis of the vehicle.

$$\alpha_f = \delta - \theta_{Vf} \quad (2.22)$$

In a similar way, the rear wheel slip angle is defined as:

$$\alpha_r = -\theta_{Vr} \quad (2.23)$$

Therefore, the lateral tire forces for the front and rear wheels of the vehicle is obtained as:

$$F_{yf} = 2C_{\alpha f}(\delta - \theta_{Vf}) \quad (2.24)$$

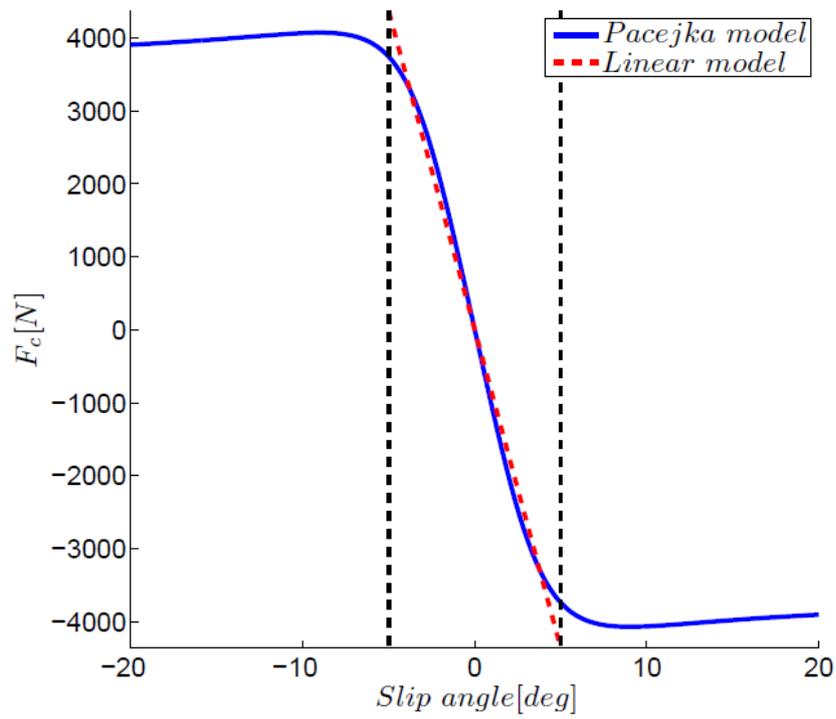


Figure 2.3: Linearized lateral tire forces in small slip angle region compared to Pacejka model

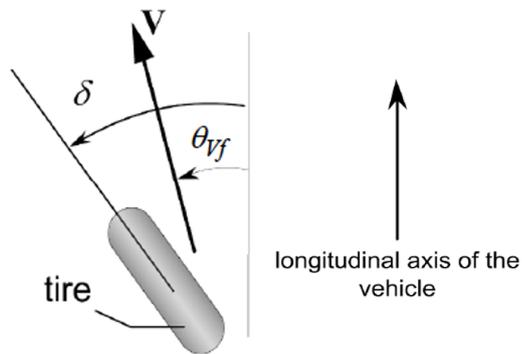


Figure 2.4: Tire slip angle

$$F_{yr} = 2C_{\alpha r}(-\theta_{Vr}) \quad (2.25)$$

where  $C_{\alpha f}$  and  $C_{\alpha r}$  are proportional constants. These constants are called cornering stiffness of front and rear wheel respectively. The factor 2 in the equations refers to the fact that there are two wheels for each axle.

In order to calculate the velocity angle of the front wheel  $\theta_{Vf}$  and the rear wheel  $\theta_{Vr}$ , the following formulas have been used:

$$\tan(\theta_{Vf}) = \frac{V_y + l_f \dot{\Psi}}{V_x} \quad (2.26)$$

$$\tan(\theta_{Vr}) = \frac{V_y - l_r \dot{\Psi}}{V_x} \quad (2.27)$$

Assuming small angle approximations, the equations 2.26 and 2.27 can be re-written as:

$$\theta_{Vf} = \frac{V_y + l_f \dot{\Psi}}{V_x} \quad (2.28)$$

$$\theta_{Vr} = \frac{V_y - l_r \dot{\Psi}}{V_x} \quad (2.29)$$

## 2.3 Vehicle model for MPC

In this thesis work, the goal is to implement a combined lateral and longitudinal control system based on MPC for autonomous driving. For this purpose, a 2 degree of freedom vehicle model is used to define the lateral dynamics of the vehicle for controller internal plant model in terms of error with respect to the reference trajectory. The two errors are lateral displacement error  $e_1$ , which is defined as the lateral distance between center of gravity of vehicle and the center line of the reference trajectory. Yaw angle error  $e_2$  is defined as the difference between the yaw angle of the vehicle and desired yaw angle as dictated by the reference trajectory, as represented in Figure 2.5. The rate of change of lateral displacement error and yaw angle error are given by the equations.

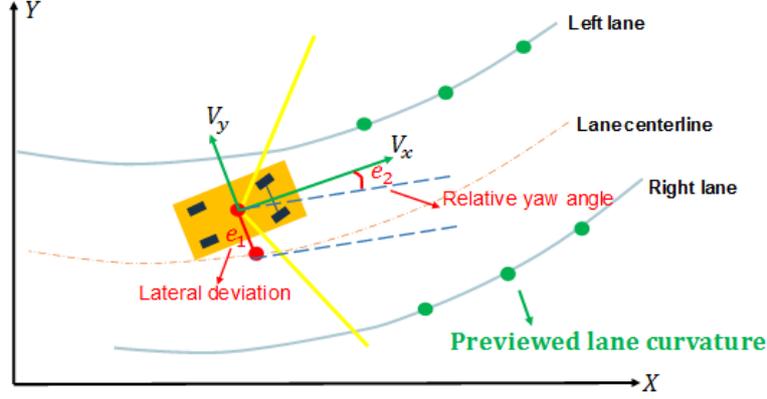


Figure 2.5: Bicycle model in terms of lateral deviation and relative yaw angle with respect to the center line of the lane

$$\dot{e}_1 = V_x e_2 + V_y \quad (2.30)$$

$$e_2 = \Psi - \Psi_{des} \quad (2.31)$$

The desired yaw angle rate is given by:

$$\dot{\Psi}_{des} = V_x \kappa \quad (2.32)$$

Where,  $\kappa$  denotes the the road curvature.

The state-space model for lateral dynamics can be obtained by linearizing the bicycle model described in section 2.1.2.  $\dot{x} = Ax + Bu$  is represented as:

$$\begin{bmatrix} \dot{y} \\ \dot{y} \\ \dot{\Psi} \\ \dot{\Psi} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & -\frac{2C_{\alpha f} + 2C_{\alpha r}}{mV_x} & 0 & -V_x - \frac{2C_{\alpha f}L_f - 2C_{\alpha r}L_r}{mV_x} \\ 0 & 0 & 0 & 1 \\ 0 & -\frac{2L_f C_{\alpha f} - 2L_r C_{\alpha r}}{I_z V_x} & 0 & -\frac{2L_f^2 C_{\alpha f} + 2L_r^2 C_{\alpha r}}{I_z V_x} \end{bmatrix} \begin{bmatrix} y \\ \dot{y} \\ \Psi \\ \dot{\Psi} \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{2C_{\alpha f}}{m} \\ 0 \\ \frac{2L_f C_{\alpha f}}{I_z} \end{bmatrix} \delta \quad (2.33)$$

For the longitudinal dynamics, the plant model used for control design is the transfer function between desired acceleration and actual vehicle speed and is given by:

$$P(s) = \frac{1}{s(\tau s + 1)} \quad (2.34)$$

Where,  $\tau$  is the time constant.

A traditional MPC controller includes a nominal operating point at which the plant model applies, such as the condition at which you linearize a nonlinear model to obtain the LTI approximation. If the plant is strongly nonlinear or its characteristics vary dramatically with time, LTI prediction accuracy might degrade so much that MPC performance becomes unacceptable. Adaptive MPC can address this degradation by adapting the prediction model for changing operating conditions. As described in the Model Predictive Control Toolbox™, adaptive MPC uses a fixed model structure, but allows the models parameters to evolve with time. Ideally, whenever the controller requires a prediction (at the beginning of each control interval) it uses a model appropriate for the current conditions. So, in an adaptive MPC, the plant model is updated at each time step as the operating point keeps changing. i.e. Vehicle longitudinal speed. The plant model used as the basis for adaptive MPC is an LTI discrete-time, state-space model with a sampling time  $T_s = 100$  ms. The combined state space model for lateral and longitudinal dynamics which is used as the internal plant model for MPC is represented below:

$$\begin{aligned}x(k+1) &= Ax(k) + B_u u(k) + B_d v(k) \\z(k) &= Cx(k)\end{aligned}\tag{2.35}$$

Where:

- $k$  is time index (current control interval).
- $x$  are plant model states.
- $u$  are manipulated inputs. These are the one or more inputs that are adjusted by the MPC controller.
- $v$  are measured disturbance inputs.
- $A$  is the state matrix.
- $B_u$  and  $B_d$  are the input matrices corresponding to inputs  $u$  and  $v$  respectively
- $C$  is the output matrix.

$$\begin{bmatrix} \dot{V}_x \\ \dot{V}_y \\ \dot{\psi} \\ \dot{e}_1 \\ \dot{e}_2 \end{bmatrix} = \begin{bmatrix} \frac{-1}{\tau} & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -\frac{2C_{\alpha f} + 2C_{\alpha r}}{mV_x} & -V_x - \frac{2C_{\alpha f}L_f - 2C_{\alpha r}L_r}{mV_x} & 0 & 0 \\ 0 & 0 & -\frac{2L_f C_{\alpha f} - 2L_r C_{\alpha r}}{I_z V_x} & -\frac{2L_f^2 C_{\alpha f} + 2L_r^2 C_{\alpha r}}{I_z V_x} & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & V_x \\ 0 & 0 & 0 & 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} V_x \\ V_y \\ \psi \\ e_1 \\ e_2 \end{bmatrix} + \begin{bmatrix} \frac{1}{\tau} & 0 \\ 0 & 0 \\ 0 & \frac{2C_{\alpha f}}{m} \\ 0 & \frac{2L_f C_{\alpha f}}{I_z} \\ 0 & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} a_x \\ \delta \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ -V_x \end{bmatrix} [\rho] \quad (2.36)$$

The inputs for the plant are separated to indicate that  $u$  correspond the front wheel steering angle and acceleration/deceleration command of the vehicle (controlled output of MPC), while  $v$  indicates the longitudinal velocity multiplied by the curvature  $\kappa$  (it is the disturbance). The inputs to the MPC  $y$  corresponds to the lateral deviation  $e_1$ , relative yaw angle  $e_2$  and velocity of the vehicle  $V_x$ . In the state vector,  $V_y$  denotes the lateral velocity,  $V_x$  denotes the longitudinal velocity and  $\phi$  denotes the yaw angle. The vehicle model refers to a high-performance autonomous car characterized by the parameters listed below.

- $m = 1575$  kg, the total vehicle mass;
- $I_z = 2875$  Nms<sup>2</sup>, the yaw moment of inertia of the vehicle;
- $l_f = 1.2$  m, the longitudinal distance from the center of gravity to the front wheels;
- $l_r = 1.6$  m, the longitudinal distance from the center of gravity to the rear wheels;
- $C_{\alpha f} = 19000$  N/rad, the cornering stiffness of the front tires;
- $C_{\alpha r} = 33000$  N/rad, the cornering stiffness of the rear tires.

## 2.4 Driveline dynamics

Generally, a lower level controller is implemented to calculate the throttle input, to track the desired acceleration determined by the MPC, which uses a simplified model of longitudinal vehicle dynamics. This simplified model is typically based on the assumptions that the torque converter in the vehicle is locked and that there is

zero-slip between the tires and the road.

For this thesis, based on the same assumptions of a simplified longitudinal dynamics model, a first order dynamics with a time constant of  $\tau = 0.5s$ , for the driveline is used. Which provides the required engine torque to track the desired acceleration. So, the engine torque required to track the desired acceleration is first calculated. This calculation is described in this section. Once the required engine torque has been obtained, engine maps and nonlinear control techniques are used to calculate to the throttle input command that will provide the required torque. The part for throttle input calculation has not been discussed in this thesis, and the reader can refer to [18] for more information on this topic.

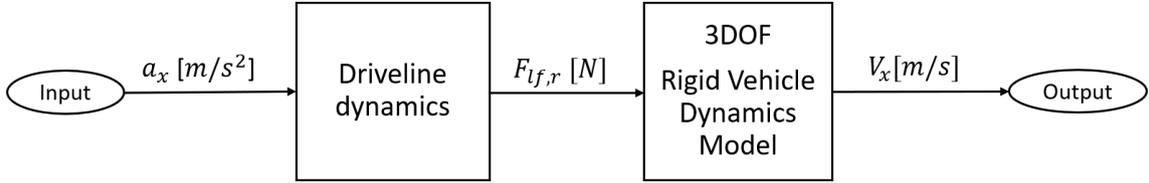


Figure 2.6: Driveline dynamics architecture

The required engine torque is given by:

$$T_{engine} = r_w(ma_x + 0.5\rho AC_x V_x^2) \quad (2.37)$$

So, the wheel torque for a given gear ratio  $i$  is given by:

$$T_{wheel} = T_{engine} \cdot i \quad (2.38)$$

Finally the force developed in the tires due to the torque is given by:

$$F_{wheel} = T_{wheel}/r_w \quad (2.39)$$

This force is given as an input to the vehicle dynamics model to accelerate and reach the reference longitudinal speed  $V_{ref}$ , as depicted in Figure 2.6.

# Chapter 3

## Control design

The aim of the controller is to safely achieve autonomous driving. The control strategy proposed here is considered in the global guidance architecture depicted in Figure 1.4. The architecture can be decomposed into three levels called Perception, Reference generation and Control:

- The Perception of the vehicle environment is of the utmost importance in the guidance architecture as it defines the environment in which the vehicle evolves. Its role is to provide the Reference generation with the necessary information.
- The Reference Generation provides reference signals. It allows the calculation of the geometric trajectory which defines the path to be followed as well as the reference speed profile. These two different reference signals calculated at this level are used by Control.
- The Control ensures the automated vehicle guidance along the generated trajectories providing the appropriate control signals, here the acceleration, the deceleration and the steering angle of the front wheel. Simultaneous longitudinal and lateral control is necessary to guarantee efficient vehicle guidance.

The architecture shown in Figure 3.1 highlights the interaction between the different blocks and present a combined controller for autonomous driving. Indeed, the lateral control is designed following a path tracking approach which helps to decouple the speed tracking and the vehicle positioning problems. However, the coupling of

the longitudinal and lateral dynamics is handled by the MPC using the constraints defined later in this section. The prediction model used here has two control inputs, i.e. the steering angle of the front wheel and the acceleration/deceleration. The steering angle is the variable of interest for lateral control and constitutes the optimization vector in the MPC problem. While, the applied acceleration/deceleration is used to track the reference velocity provide by the speed profile generator. Which is then used to calculate the required torque for the desired acceleration. In this way, MPC based lateral and longitudinal controller ensure the coupled path and speed tracking. Note that no active lateral stabilisation aspect is considered in the control design. In extreme lateral manoeuvres, vehicle stability may then be lost, e.g. when large steering manoeuvres are performed at high speed. In order to preserve vehicle lateral stability during guidance, the longitudinal reference speed should be adapted. To do so, a reference speed profile generator has been adopted, described in section 3.3.

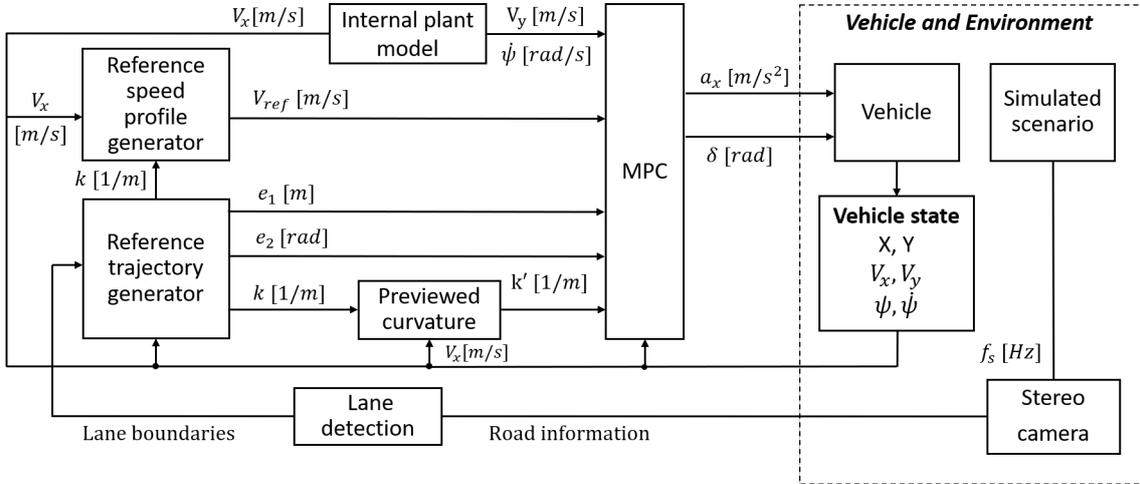


Figure 3.1: Detailed architecture of the control strategy

As mention in section 1.1, the overall system has been implemented in MATLAB and Simulink.

## 3.1 Perception

According to the the block scheme in Figure 3.1, in this section the stereo camera and lane detection block has been presented. In this work, the stereo camera is implemented in Simulink using the *Vision Detection Generator* block from *Automated Driving Toolbox*. Which generate vision detections from simulated scenarios at the intervals of 100ms. Therefore, simulated driving scenarios are used to simulate the environment and generate the synthetic data required for the control algorithm of the vision detection. In particular the detection of road lanes has been performed following a visual perception example included in the MATLAB documentation [19], explained in the next section, that uses *Automated Driving System*, *Computer Vision System* and *Image Processing* toolbox.

The first step is the definition of the configuration of a monocular camera sensor. Configuration information includes the intrinsic (Focal length and optical center of the camera) depicted in Figure 3.3 and extrinsic parameters (Orientation (pitch, yaw, and roll) and the camera location within the vehicle to define the camera orientation with respect to the vehicle’s chassis) in the *Vision Detection Generator* block. The camera is mounted on top of the vehicle at a height of 1.5 meters above the ground and a pitch of 1 degree toward the ground in this thesis work as shown in Figure 3.2. This information is later used to establish camera extrinsics that define the position of the camera coordinate system with respect to the vehicle coordinate system.

Focal length = [800, 800];

Optical center of the camera = [320, 240];

### 3.1.1 Lane detection

Lane detection is a well-research area of computer vision that allows to realize functions for ADAS and autonomous vehicles. One of these functions is the lane keeping, and the lane detection presented in this thesis has been developed to give reliable information to implement it. This block receives, as an input, images acquired by the simulated stereo camera, with a frame rate equal to 10 Hz and provides the equation

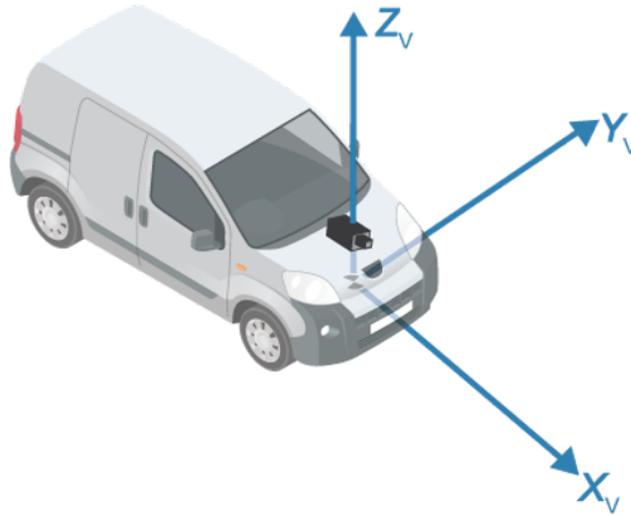


Figure 3.2: Vehicle with camera location

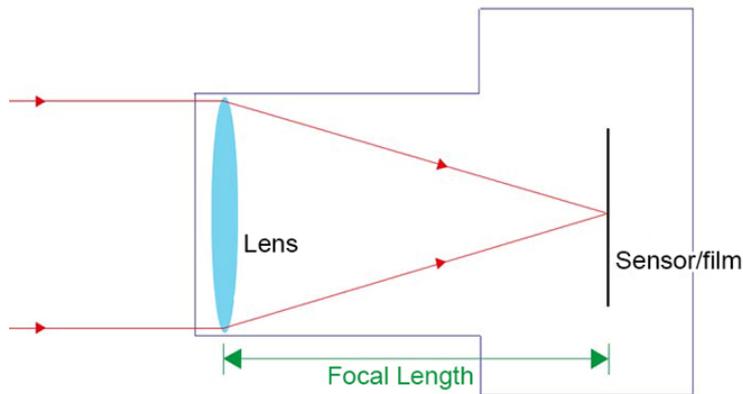


Figure 3.3: Focal length description

of the left and right lane boundaries of the current lane in the current field-of-view of the stereo camera to the reference trajectory generator block. Which provides the current curvature, lateral deviation and relative yaw angle of the vehicle with respect to the center line using the method explained in the next sections.

Lane detection has been divided in two parts:

- Lane line feature extraction;
- Lane line model.

Firstly, the Region of Interest (ROI) has to be defined. It defines the area to

transform in bird’s-eye-view images so that it is possible to have a sufficient prediction of the road in front of the vehicle and a suitable side view in order to see a lane. After the extraction of the ROI, the *birdsEyeView* object has been developed to perform the transformation of the original image into the bird’s-eye-view image using Inverse Perspective Mapping. A result of this transformation can be seen in Figure 3.4.

The bird’s-eye-view image allows the function to perform the feature extraction and the lane line model, as explained in hereafter. 3.1.1. Lane line feature extraction [20] consists in identifying pixels that belong to the white line of the road and eliminating the marking pixels of non-lane line, in bird’s-eye-view images that coming from the previous phase.

The extraction is developed using an approach that is based on the observations of pixels contrast compared between the lane markings and the road pavement. The recognition of lines is implemented by searching for pixels that are “lane-like”. This type of pixels are groups of points with a very different colour contrast with respect to the adjacent points on both sides.

The approach developed to this purpose is called *ridge detection* that tries to identify ridges, or edges, in an image. Ridge detection technique has been chosen for its simplicity and relative effectiveness. It is based on tensor field construction of first order derivatives and it is able to get the response of gradient directions that makes it easier to remove anomalous values if their directions deviate too much from the expected lane line direction [21].

In order to improve the lane line feature segmentation, the method requires to transform the bird’s-eye-view images from RGB to grey-scale, as shown in Figure 3.4. Automated Driving System toolbox provides a function that uses a ridge detector to extract the lane line feature, *segmentLaneMarkerRidge*.

This function receives in input the bird’s-eye-view image in grey-scale intensity, the *birdsEyeView* object created in the Inverse Perspective Mapping phase and a scalar value that indicates the approximate width of the features of the lane line to detect. The last value allows the function to determine the filter used to threshold the intensity contrast. *segmentLaneMarkerRidge* can receive an additional input arguments, the lane sensitivity, a non-negative scalar factor that allows to define if a value needs

to be retained or not. This value improves the detection and extraction of features [22] [23].

As output, the function returns a binary image with true pixels representing the information about lane features, as shown in Figure 3.4. After the feature extrac-



Figure 3.4: Lane line feature extraction

tion, the lane line model fitting has been developed. This step allows to create a parametric model of the lane detected to the visualization of the features extracted in the image. The main purpose of this phase is to get a compact high level representation of the path, which can be used for decision making [24].

In this thesis work the built-in *findParabolicLaneBoundaries* function has been used to fit the lane line model. This function uses RANSAC algorithm [26] to find the lane line boundaries. As the function name suggests, the model created is a parabolic model that fits a set of boundary points and an approximate width. The selected boundary points correspond to inliers only if they fall into the boundary width. The final parabolic model has been obtained using a least-squares fit on the inlier points.

The function receives in input the candidate points in vehicle coordinate from the features extraction phase and it provides array of *parabolicLaneBoundary* objects for each model. The returned array includes the three coefficients [a b c] of the parabola, like a second-degree polynomial equation  $ax^2+bx+c$ , and in addition the strength, the type, and the minimum and maximum  $x$  positions of the computed boundary. The last three parameters are used to reject some curves that could be



Figure 3.5: Lane line model

invalid using heuristics [27]. For example, in order to reject short boundaries, the difference between the minimum and maximum  $x$  positions has been compared with a specific threshold, if the minimum threshold is not reached, the found boundaries are rejected; or, to reject weak lines, the value of the strength has to be higher than another threshold set ad hoc.

The founded lane line models in vehicle coordinate have been inserted to the bird's-eye-view image and to the original image taking from the camera, as shown in Figure 3.5.

The better results of the lane detection have been found in roads with straight line and light curves, while some limitations have been found when there are cross-roads, roundabouts and very high curvature roads.

The reader can refer to the thesis [28] for more information related to lane detection.

## 3.2 Reference trajectory generation

The trajectory generation phase consists to find the trajectory and compute its curvature based on the information of the lane line model coming from the previous

step. This phase refers to the problem of trajectory planning, also called motion planning, in automotive context, that has the purpose to find a trajectory feasible for the vehicle, and safe and comfortable for the passenger.

The motion planning for an autonomous vehicle is based on the same theory handled in robotics area. In fact, as in the field of robotics, it is necessary to provide and distinguish some definitions such as path and trajectory, and global and local planning.

Firstly, it is significant to give the definitions of path and trajectory and underline that they have two different meanings:

- *Path* is the pure geometric description of motion;
- *Trajectory* is the merge of the path and the time laws (velocities and accelerations) required to follow the path.

The other significant definitions are global and local planning:

- *Global planning* means the generation of the path or trajectory knowing the entire environment and its information such as the position of the obstacle and the lane boundaries;
- *Local planning* means, instead, the computation of the path according to sensor data that represent local environment information.

In this thesis, for the sake of simplicity, no strict distinction has been adopted to distinguish path and trajectory when needed.

Moreover, the indication of the trajectory (or similarly path) is defined as a local path as mention in the previous definitions.

The trajectory computed for this work consists of the center line of the lane. It is computed like the average between the left line of the lane and the right ones.

### 3.2.1 Trajectory curvature computation

The controller of the lane keeping needs to receive the curvature of the trajectory like input to perform the control action on the steering angle.

“*The curvature of a curve parametrized by its arc length is the rate of change of direction of the tangent vector [29]*”.

Considering a curve  $\alpha(s)$ , where  $s$  is the arc length and the tangential angle  $\phi$ , computed counterclockwise from the  $x$ -axis to the tangent  $T = \alpha'(s)$ , as shown in Figure 3.6, the curvature  $\kappa$  of  $\alpha$  is defined, following the definition, as:

$$\kappa = \frac{d\phi}{ds} \quad (3.1)$$

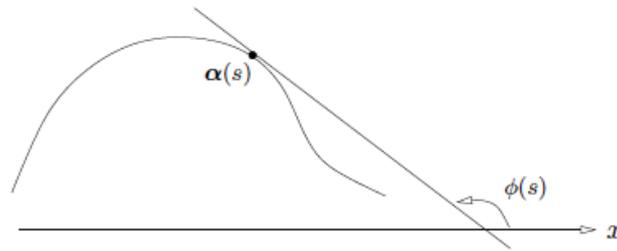


Figure 3.6: Curve  $\alpha$  and tangential angle  $\phi$

The curvature can be also defined as the value of the turning of the tangent  $T(s)$  along the direction of the normal  $N(s)$ , that is:

$$\kappa = T' \cdot N \quad (3.2)$$

It is easily to derive the first definition 3.1 from the second 3.2 (Figure 3.7), as follows:

$$\kappa = T' \cdot N = \frac{dT}{ds} \cdot N = \lim_{\Delta s \rightarrow 0} \frac{T(s + \Delta s) - T(s)}{\Delta s} \cdot N = \lim_{\Delta s \rightarrow 0} \frac{\Delta\phi \cdot \|T\|}{\Delta s} = \frac{d\phi}{ds} \quad (3.3)$$

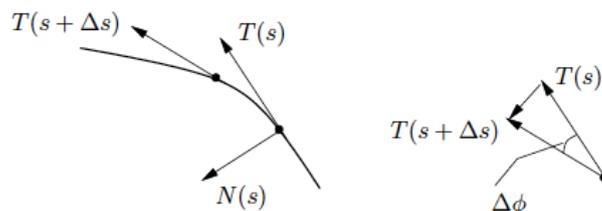


Figure 3.7: Demonstration that the definition 3.1 can be derived from the definition 3.2

To perform the measure of how sharply the curve bends, the absolute curvature of the curve at a point has been computed and it consists of the absolute value of the curvature  $|\kappa|$ .

A small absolute curvature corresponds to curves with a slight bend or almost straight lines. Curves with left bend have positive curvature, while a negative curvature refers to curves with right bend.

With the second definition 3.2 it is possible defined that the curvature of a circle is the inverse of its radius everywhere. For this reason, the radius of curvature  $R$  has been identified as the inverse of the absolute value of the curvature  $\kappa$  of the curve at a point.

$$R = \frac{1}{|\kappa|} \quad (3.4)$$

The circle with radius equal to the curvature radius  $R$ , when  $\kappa \neq 0$ , and positioning at the center of curvature is called *osculating circle*, as shown in Figure 3.8. It allows to approximate the curve locally up to the second order.

The curvature can be expressed in terms of the first and second derivatives of the curve  $\alpha$  for simplicity in the computation, by the following formula:

$$\kappa = \frac{|\alpha''|}{[1 + (\alpha')^2]^{\frac{3}{2}}} \quad (3.5)$$

In order to compute the curvature in this thesis work, the *Geom2d* toolbox in MATLAB has been used. This toolbox provides the *polynomialCurveCurvature* function that allows to compute the local curvature at specific point of a polynomial curve.

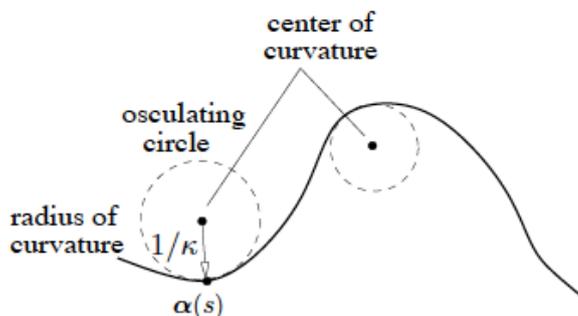


Figure 3.8: Osculating circle and radius of curvature

It receives in input the curve in parametric form  $x = x(t)$  and  $y = y(t)$  and the point in which the curvature has to be evaluate.

The function *polynomialCurveCurvature* computes the curvature following the formula 3.5 that becomes:

$$\kappa = \frac{|x'y'' - x''y'|}{[(x')^2 + (y')^2]^{\frac{3}{2}}} \quad (3.6)$$

### 3.2.2 Computation of vehicle model dynamic parameters

The last phase of the lane detection algorithm refers to the computation of vehicle model dynamic parameters. These values are necessary in order to achieve the goal of the control stage for the lane keeping. The controller has to minimize the values of lateral deviation and relative yaw angle in order to compute the optimal steering angle.

Lateral deviation and relative yaw angle are defined as follow:

- *Lateral deviation* is the distance of the center of mass of the vehicle from the center line of the lane;
- *Relative yaw angle* is the orientation error of the vehicle with respect to the road.

These parameters are computed geometrically after a 2D reconstruction of the road (Figure 3.9): the lateral deviation is considered the distance between the camera

mounted at the center of the vehicle that is become the origin of the new reference frame created by *monoCamera* object, and the center line computed in the previous phase; while, the relative yaw angle is identified as the angle between the vector of the longitudinal velocity and the tangent to the center line. With the information

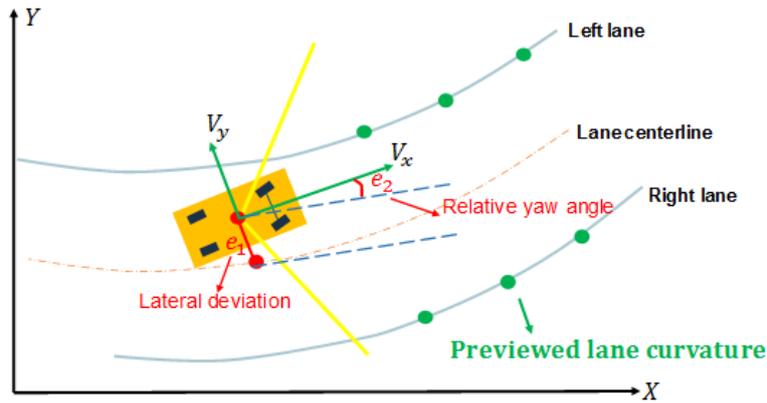


Figure 3.9: Definition of lateral deviation and relative yaw angle with respect the center line of the lane

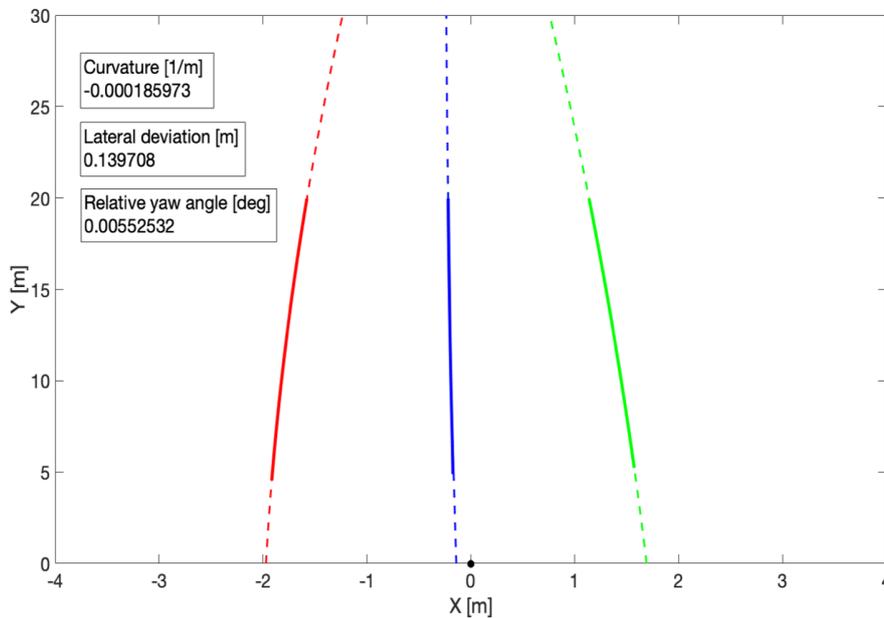


Figure 3.10: Center line, curvature, lateral deviation and relative yaw angle computation

about the lane line model, the function performs a reconstruction of the road in order to compute the center line of the lane and the relative curvature, as specified in the previous section. Based on the computed trajectory, the lateral deviation and the relative yaw angle of the vehicle has been calculated as described in this section. Figure 3.10 shows an example of the plot in MATLAB about these computations.

### 3.3 Reference speed profile generation

The following subsections are devoted to determine the reference speed profile, two different criteria are considered here, which are available in literature. First one is based on the geometry of the road and the second one is based on the lateral comfort of the vehicle. So, maximum admissible longitudinal speed is estimated based on the road information and the speed for lateral comfort is calculated based on the information about the desired lateral acceleration. Both of them are exploited to calculate the reference speed profile by the speed profile generator.

*Road information criteria:* the performance of the path-following depends on the speed with which this following is done. The cruise speed is also important for the stability of the vehicle on the road. In fact, no controller can ensure the path-following if the cruise speed is excessive. Thus, the speed of the vehicle should be reduced when approaching a bend. This adaptation of the cruise speed depends on the difficulty to cross the bend. There are several systems designed by automakers for assisting driver when approaching a bend, like those developed by Daimler-Chrysler defining the maximum admissible speed based on the curvature of the road:

$$V_{max} = \sqrt{\frac{g\mu}{\kappa}} \quad (3.7)$$

where  $g$ ,  $\mu$  and  $\kappa$  are respectively the gravity, the friction coefficient and the road curvature. The description given by the model (3.7) is incomplete and may be inappropriate to determine the maximum admissible speed in some situations. Indeed, the only parameter considered in this model is the road curvature. However, other characteristics of the road can be considered. For this reason, more sophisticated models are proposed. The National Highway Traffic Safety Administration (NHTSA) recommends for the calculation of the maximum entry speed in bends the

following model:

$$V_{max} = \sqrt{\frac{g}{\kappa} \left( \frac{\phi_r + \mu}{1 - \phi_r \mu} \right)} \quad (3.8)$$

where  $\phi_r$  is the road camber angle.

Then, the acceleration  $a$  that should be applied to bring the speed of the vehicle to the maximum admissible speed given by (3.8) should be less than:

$$a_{max} = \sqrt{\frac{V^2 - V_{max}^2}{2(d - t_r V)}} \quad (3.9)$$

where  $V$  is the current vehicle speed,  $d$  distance to the summit of the bend and  $t_r$  the time-delay due to driver reaction. The purely geometric models (3.7) and (3.8) can be evaluated in real-time and can be used in a predictive way as the road data are already employed in the MPC strategy. Notice that these criteria do not handle the vehicle lateral dynamics. Thus, in our work these criteria are combined with other indicators on the lateral stability presented in the following section.

The determination of comfort speed is essential so that the lateral acceleration of the host vehicle does not exceed a critical value: in fact, for high lateral accelerations, vehicle model goes non-linear and controlling the vehicle becomes more difficult. Therefore, based on human comfort experiments published in [30], the absolute value of lateral acceleration is limited via velocity-dependent constraints as

$$a_{y,comfort} = a_{y,o} \left( 1 - \frac{V_x}{V_{max}} \right) \quad (3.10)$$

$$V_{comfort} = \sqrt{\frac{a_{y,comfort}}{\kappa}} \quad (3.11)$$

where,  $a_{y,o} = 4m/s^2$  is the acceptable lateral acceleration,  $a_{y,comfort}$  is the desired lateral acceleration of the host vehicle,  $V_{comfort}$  is the desired velocity of the host vehicle in the terms of comfort in curve,  $g$  is the gravitational acceleration,  $V_{max}$  is the maximum speed of the vehicle. Note that (3.10) decreases linearly and monotonically for higher velocities. The rationale behind (3.10) is the following: experimental studies on human driving show that drivers tend to have lower lateral acceleration

values at higher speeds: so, a velocity-dependent lateral comfort constraint is designed in view of human comfort. Therefore, the speed profile is set as

$$V_{ref} = \min(V_{max}, V_{comfort}) \quad (3.12)$$

For lateral stability of the vehicle an additional condition is applied to improve the lateral motion. So, a desired longitudinal acceleration is calculated from physical limitation in braking with cornering.

$$a_x = \frac{\sqrt{(\mu gm)^2 - \sum(F_y)^2}}{m} \quad (3.13)$$

$$\sqrt{F_x^2 + F_y^2} < \mu F_z \quad (3.14)$$

In this way, a constrain on the longitudinal acceleration is imposed using the Kamm inequality, which keeps the forces developed in the tires within the physical limitations of the tire-road friction. Where,  $F_y$  can be either estimated or it can be measured using recently developed technology like smart tires or load sensing bearings to compute the  $a_x$  in real time.

The information on lateral dynamics is of capital importance as it helps to determine loss of control and help to preserve the lateral stability. In this work, following criteria is used, which gives the  $\beta_{limit}$ :

$$\beta_{limit} < 10^\circ - 7^\circ \frac{(V_x)^2}{(40m/s)^2} \quad (3.15)$$

where,  $\beta$  is the sideslip angle of the vehicle and  $V_x$  is the vehicle speed.

The Reference Generation provides the lateral deviation and relative yaw angle to be minimized by the vehicle and a speed profile taking legal speed limits and vehicle comfort into account.

## 3.4 Model Predictive Control

In this section, theory behind Model Predictive Control will be explored together with the derivation of the Adaptive MPC, used to control the vehicle.

The aim of this thesis is to design a controller that allows autonomous driving. We decided to use Model Predictive Control, due to its abilities to work with constraints both on the states and the control signals. This is crucial for the control of a vehicle since it is constrained not only by mechanics of the vehicle but also by the environment. For example, a vehicle should not exceed the speed limits or drive too close to other vehicles.

### 3.4.1 Overview of MPC

Model Predictive Control (MPC) is an advanced control method that works in discrete time. From a set of state values, and with respect to a model, it optimizes a problem around an objective and gives a sequence of control signals as outputs. The first set of control values are then used as inputs to the system plant, and after a short period, set as the system time step, the new state values are measured and the process is repeated. In this section we will shortly describe the history of MPC and give some basic examples of its structure and the theory behind it.

The beginning of MPC was at Shell Oil Company in 1979 where an idea named as "Dynamic Matrix Control" was presented by Cutler and Ramaker [31]. DMC was the first type of predictive control that could be applied in industry. The idea was to handle multi variable control systems without any constraints and predict future values for linear systems. The idea that the algorithm would predict future plant behavior was discovered to lead to a less aggressive output and a smoother convergence to the target set point. Throughout the 80s MPC was popular mainly in industries such as chemical plants and oil refineries [32], i.e. in slow processes where the computational time of the solvers would not be a problem. In the 90s the theory of MPC matured and with faster solvers and computers the algorithm was now feasible for faster, more demanding systems. Today MPC has many applications, and as we will demonstrate in this thesis, one of them is in autonomous driving.

According to Qin and Badgwell [33], the overall objectives of a MPC controller are:

1. Prevent that input and output constraints are violated;
2. Optimize some output variables, while others outputs are kept in a specified ranges;
3. Prevent that the input variables have excessive movement;
4. Control the major number of process variables when a sensor or actuator is down or is not available.

Three critical steps affect the process of a MPC controller: prediction model, optimization solution and feedback correction.

A general architecture of a Model Predictive Control used for autonomous driving vehicle is given by Figure 3.11.

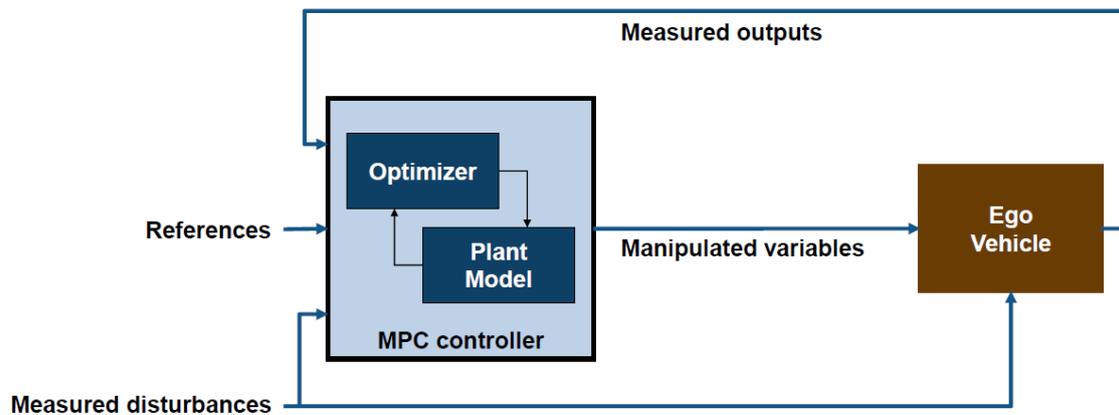


Figure 3.11: Block diagram for Model Predictive Control

MPC controller has two main functional blocks: the optimizer and the vehicle model. The dynamic optimizer allows to find the optimal input that gives the minimum value of the cost function taking into account all the constraints. The vehicle and the plant model refers to the 3DoF rigid vehicle model and a linearized state space model, described in the section 2.3. Generally, a non linear model is used for the validation of the controller, while the plant model used for the MPC is a linearized version of the actual plant.

The MPC controller provides the optimal output to send to a plant based on a finite horizon using an iterative approach. Its main goal is to calculate a sequence of *control moves*, that consist of manipulated input changes, so that the predicted output moves to the set point in an optimal manner.

Referring to Figure 3.12,  $y$  is the actual output,  $\hat{y}$  is the predicted output and  $u$  consists of the manipulated input. At the current sampling time  $k$ , the initial value of the plant state is known and the MPC computes a set of  $M$  values of the input  $u(k+i-1)$ ,  $i = 1, 2, \dots, M$ , where  $M$  is called *control horizon*. This set refers to the current input  $u(k)$  and to  $(M - 1)$  prediction inputs, and it is held constant after the  $M$  control moves. The inputs are computed so that a set of  $N$  predicted outputs  $\hat{y}(k + i)$ ,  $i = 1, 2, \dots, N$  reaches the set point in optimal manner.  $N$  is called *prediction horizon* and consists of the number of future steps to look ahead [35].

When we are driving we never look straight down at the road, but farther ahead. The reason is of course so that we can plan our driving. When a sharp turn approaches we need to brake ahead of time. A driver always looks far enough to ensure safe driving, so called minimum braking distance, in case of an unexpected obstacle on the road. This should also apply in control. In Model predictive control there is a finite prediction horizon set for each optimization, i.e., how far the controller looks into the future. To decide the length of the horizon we can again draw an analogy to human driving. While driving at high speeds you need a longer prediction horizon since the minimum braking distance is also longer. The prediction horizon must be long enough such that distance between the two cars are larger than the minimum braking distance. A longer horizon is usually ideal but is often limited by sensor limitations. The computational complexity also increases for longer horizons, mainly for complex non-linear systems. So, the values of control horizon  $M$  is usually kept

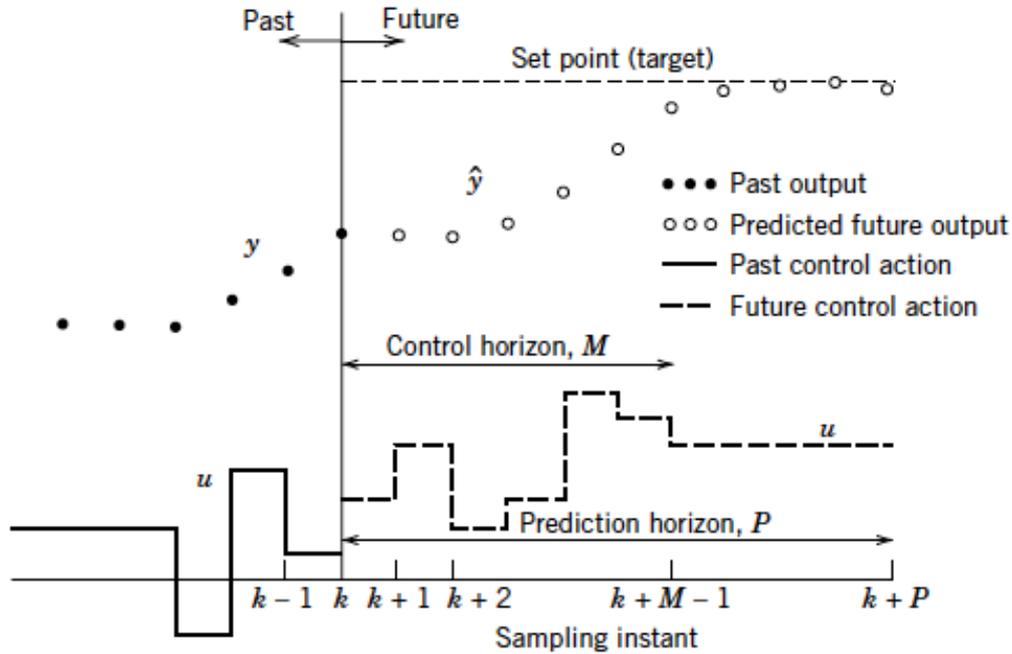


Figure 3.12: Basic concept for Model Predictive Control

lower than the prediction horizon  $N$  as the controller apply only first control step and solves the optimization problem again. In practical situations, only the first value of the whole set of  $M$  values is implemented as the input of the system because the model of the process is simplified and inaccurate. Moreover, this set can add disturbances or noises in the process that could produce an error between the actual output and the predicted one.

For this reason, the plant state has to be measured again to be adopted as the initial state for the next step. The re-measurement of the information state is reported with a feedback to the dynamic optimizer of the MPC controller and adds robustness to the control [34]. When the plant state is re-sampled, the whole process computes again the calculations starting from the new current state. The window of the prediction horizon shifts forward at every time step. This is the reason why the Model Predictive Control is also called *Receding Horizon Control*.

### 3.4.2 MPC problem formulation

The MPC controller implemented in this thesis is based on the method of multiple-step optimization and feedback correction. Thanks to this method, the controller has good performances of control.

Lateral control deals with the actuation of the steering of the vehicle to keep it in the center of the lane and follow the curved road. It is modeled as a reference path tracking problem for the MPC with the objective of minimizing the lateral deviation  $e_1$  and relative yaw angle  $e_2$ . While, the longitudinal control deals with the actuation of the throttle/brake to control the longitudinal speed of the vehicle. It is modelled as a reference speed tracking problem, which is generated using the reference speed profile calculated using (3.12). Based on the reference velocity MPC computes the desired acceleration command to attain it. In other words, the objective of the MPC is to converge the speed of the vehicle to the desired reference speed. The inputs for the MPC are actual longitudinal velocity  $V_x$ , lateral deviation  $e_1$  and relative yaw angle  $e_2$ , which are the outputs of the actual plant model. i.e. 3DoF rigid vehicle model. Based on these three inputs the MPC solves the optimization problem as reference tracking. The reference variables are given by reference velocity  $V_{ref}$ , while  $e_1$  and  $e_2$  are set equal to zero. So, The goal of the MPC controller is to compute the optimal steering angle and throttle/brake command to perform the autonomous driving. In order to achieve this goal, the controller calculates the steering angle and throttle/brake by minimizing its cost function.

The description of the Adaptive MPC has been divided into two parts:

- *Problem formulation* in which is explained how the MPC problem has been formulated;
- *Output prediction* in which is defined how the predicted output has been computed.

### Problem formulation

The formulation of the MPC problem developed in this thesis starts defining a linear state-space model derived in section 2.3, which is represented as:

$$\begin{aligned} x(k+1) &= Ax(k) + B_u u(k) + B_d v(k) \\ y(k) &= Cx(k) \end{aligned} \quad (3.16)$$

Where:

- $A$  is the state matrix;
- $B_u$  and  $B_d$  are the input matrices corresponding to inputs  $u$  and  $v$  respectively;
- $C$  is the output matrix.

Given the linear model defined in equation 3.16, the Model Predictive Control algorithm is implemented as solving the following optimization problem at each time step:

$$\begin{aligned} \min_u J &= \sum_{j=1}^N \|y_p(k+j|k) - y_{ref}(k+j|k)\|_{Q_y} + \sum_{j=0}^{M-1} \|u(k+j|k)\|_{R_u} \\ \text{s.t. } x(k+j+1|k) &= Ax(k+j|k) + B_u u(k+j|k) + B_d v(k+j|k) \\ x(k|k) &= x(k) \\ y(k+j|k) &= Cx(k+j|k) \\ |u(k+j|k)| &\leq u_{limit} \end{aligned} \quad (3.17)$$

Where  $u$  is the manipulated variable.  $Q_y$  and  $R_u$  are weights for outputs and manipulated variables respectively. This optimization problem refers to find the value of input  $u$  that minimizes the sum of the weighted norms of the error between the predicted output vector  $y_p$  and the reference vector for those states  $y_{ref}$  and the input vector  $u$  for a defined prediction horizon  $N$  and control horizon  $M$ . The predicted output  $y$  has to satisfy the linear model, while the value of  $u$  should not exceed a specified limit  $u_{limit}$ .

The state vector  $y$  is given by:

$$\begin{bmatrix} V_x & e_1 & e_2 \end{bmatrix}^T$$

While, the state vector  $y_{ref}$  is given by:

$$\begin{bmatrix} V_{ref} & 0 & 0 \end{bmatrix}^T$$

$V_x$  is directly taken from the vehicle dynamics block as an output while  $e_1$  and  $e_2$  are taken from the reference trajectory block. These three states are sent as feedback to the MPC controller in order to correct the control variables in the future step time with respect to the reference states.

The weighted norm of the vector  $y = \begin{bmatrix} y_1 & y_2 & y_3 \end{bmatrix}^T$  corresponds to:

$$\|y(k+j|k)\|_{Q_y} = \begin{bmatrix} y_1 & y_2 & y_3 \end{bmatrix} \begin{bmatrix} q_{11} & 0 & 0 \\ 0 & q_{22} & 0 \\ 0 & 0 & q_{33} \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \\ y_3 \end{bmatrix} \quad (3.18)$$

where the weights  $q_{11}$ ,  $q_{22}$  and  $q_{33}$  are tuned to provide the needed damping on the corresponding output. The same definition is applied to the weighted norm of  $u$  given by:

$$\|u(k+j|k)\|_{R_u} = \begin{bmatrix} u_1 & u_2 \end{bmatrix} \begin{bmatrix} r_{11} & 0 \\ 0 & r_{22} \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} \quad (3.19)$$

**Output prediction**

The values of the predicted output  $y(k + j|k)$ ,  $j = 1, 2, \dots, N$ , where  $N$  is the prediction horizon, have been computed using the linear state-space model described by the formula 3.16.

In particular, in order to make the computation, the following values have to be known:

- Present output measurement  $y(k|k) = y(k)$ ;
- Applied input  $u(k|k) = u(k)$ ;
- Entire set of predicted input values  $v(k + j|k)$ ,  $j = 0, 1, 2, \dots, N$ .

If the prediction state is defined as follows:

$$\begin{aligned}
 x(k + 1|k) &= Ax(k) + B_u u(k|k) + B_d v(k|k) \\
 x(k + 2|k) &= Ax(k + 1|k) + B_u u(k + 1|k) + B_d v(k + 1|k) = \\
 &= A^2 x(k) + AB_u u(k|k) + AB_d v(k|k) + B_u u(k + 1|k) + B_d v(k + 1|k) \\
 &\quad \vdots \\
 x(k + N|k) &= Ax(k + N - 1|k) + B_u u(k + N - 1|k) + B_d v(k + N - 1|k) = \\
 &= A^N x(k) + A^{N-1} B_u u(k|k) + A^{N-1} B_d v(k|k) + A^{N-2} B_u u(k + 1|k) + \\
 &A^{N-2} B_d v(k + 1|k) + \dots + B_u u(k + N - 1|k) + B_d v(k + N - 1|k)
 \end{aligned} \tag{3.20}$$

The prediction output can be identified by the following equations:

$$\begin{aligned}
 y(k|k) &= Cx(k) \\
 y(k+1|k) &= Cx(k+1|k) \\
 y(k+2|k) &= Cx(k+2|k) \\
 &\vdots \\
 y(k+N|k) &= Cx(k+N|k)
 \end{aligned} \tag{3.21}$$

Using the equations 3.20 and 3.21, it is possible to express the predicted outputs  $y(k+1|k), \dots, y(k+N|k)$  as a function of the predicted inputs  $u(k|k), \dots, u(k+N-1|k)$ , noted that the other signals are assumed to be known as stated above.

In order to make the relation between the equations 3.20 and 3.21 clearer, the prediction output of the future can be defined as follows:

$$Z(k) = Gx(k) + HU(k) + EV(k) \tag{3.22}$$

Where:

- $Z(k)$  is the augmented vector of the predicted outputs;
- $U(k)$  is the augmented vector of the computed future inputs;
- $V(k)$  is the augmented vector of the predicted disturbances.

These vectors are obtained by the chaining of the input and the output vectors in the present time until the future  $N$  vectors ( $N - 1$  vectors for the input  $u$  and  $v$ ), and they are defined as follows:

$$Z(k) \equiv \begin{bmatrix} z(k|k) \\ z(k+1|k) \\ \vdots \\ z(k+N|k) \end{bmatrix}; U(k) \equiv \begin{bmatrix} u(k|k) \\ u(k+1|k) \\ \vdots \\ u(k+N-1|k) \end{bmatrix} \text{ and } V(k) \equiv \begin{bmatrix} v(k|k) \\ v(k+1|k) \\ \vdots \\ v(k+N|k) \end{bmatrix}$$

The matrices G, H and E are determined in the following way:

$$G = \begin{bmatrix} C \\ CA \\ CA^2 \\ \vdots \\ CA^N \end{bmatrix}; H = \begin{bmatrix} 0 & 0 & 0 & \dots & 0 \\ CB_1 & 0 & 0 & \dots & 0 \\ CAB_1 & CB_1 & 0 & \dots & 0 \\ \vdots & \vdots & \vdots & & \vdots \\ CA^{N-1}B_1 & CA^{N-2}B_1 & CA^{N-3}B_1 & \dots & CB_1 \end{bmatrix} \text{ and}$$

$$E = \begin{bmatrix} 0 & 0 & 0 & \dots & 0 \\ CB_2 & 0 & 0 & \dots & 0 \\ CAB_2 & CB_2 & 0 & \dots & 0 \\ \vdots & \vdots & \vdots & & \vdots \\ CA^{N-1}B_2 & CA^{N-2}B_2 & CA^{N-3}B_2 & \dots & CB_2 \end{bmatrix}$$

As mentioned before, the proposed control strategy maximizes the longitudinal speed while remaining in constrained speed range and without exceeding the adherence condition. At the same time, it eliminates the path error between the actual location and the desired path in terms of lateral deviation and desired yaw angle, assuring the handling stability during the motion.

# Chapter 4

## Results and discussions

In this chapter three different driving scenarios for simulation are presented and latter the simulation results are presented and discussed.

### 4.1 Driving scenarios

We have considered three different simulation scenarios represented in Figure 4.1, Figure 4.2, Figure 4.3 to validate the proposed control strategy:

- *Scenario 1. Highway driving:* this part has mainly straights and high curvature turns. Also, a highway exit is added at the end to depict the transition from highway to inter urban. The vehicle can drive at the maximum vehicle speed considering the speed limits of the highway (130km/h for Italian Highways). The road is modeled with three lanes. The lines between the lanes are dashed and each lane has a width of 3.2m. For the simulation, the car is placed in the middle lane with an initial longitudinal speed equal to 0km/h.
- *Scenario 2. Inter Urban driving:* this part considers rural and suburban roads, which lies between city and highway driving. The shape of the road includes both straights with high curvature and some sharp turns. It connects city with rural areas or city with highways. The speed limit in suburban area is considered as 90km/h. The road is modeled with three lanes. The lines between the lanes are dashed and each lane has a width of 3.2m. For the

simulation, the car is placed in the middle lane with an initial longitudinal speed equal to 0km/h.

- *Scenario 3. Urban driving:* this part has complex city road elements, like signalized intersections, streets, parking places and buildings, and moving obstacles. For simplicity, we have realized a roundabout and some sharp turns in the scenario to check the performance of the controller in challenging condition. The speed limit in urban area is considered as 47 km/h. The road is modeled with two lanes. The lines between the lanes are dashed and each lane has a width of 3m. For the simulation, the car is placed in the right lane with an initial longitudinal speed equal to 0km/h.

Together, they form a comprehensive set of key environments for research and development of automated driving. As mentioned in section 1.1, the overall system has been implemented in MATLAB and Simulink and the driving scenarios are created using the *Driving scenario designer application* in the Automated driving toolbox.

In the figures below, all the three driving scenarios with their road curvature are represented. The urban driving scenario Figure 4.3 is characterized by higher values of curvature relative to sharp turns, while the highway driving scenario Figure 4.1 has lower curvature values. Inter-urban Figure 4.2 lies between these two with intermediate curvature.

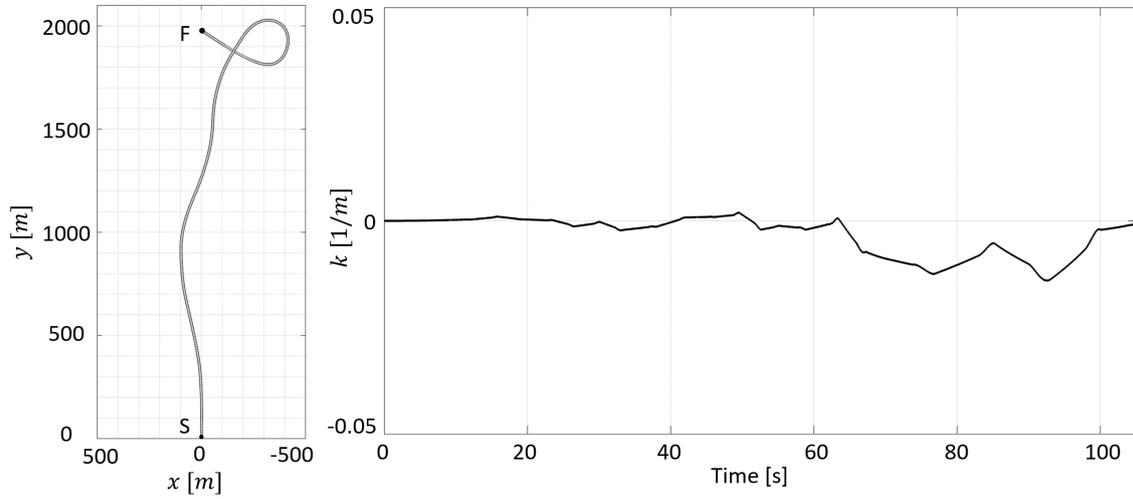


Figure 4.1: (a) Highway driving scenario: S is the vehicle's starting point. F is the end of the road track. (b) Detected road curvature  $k$

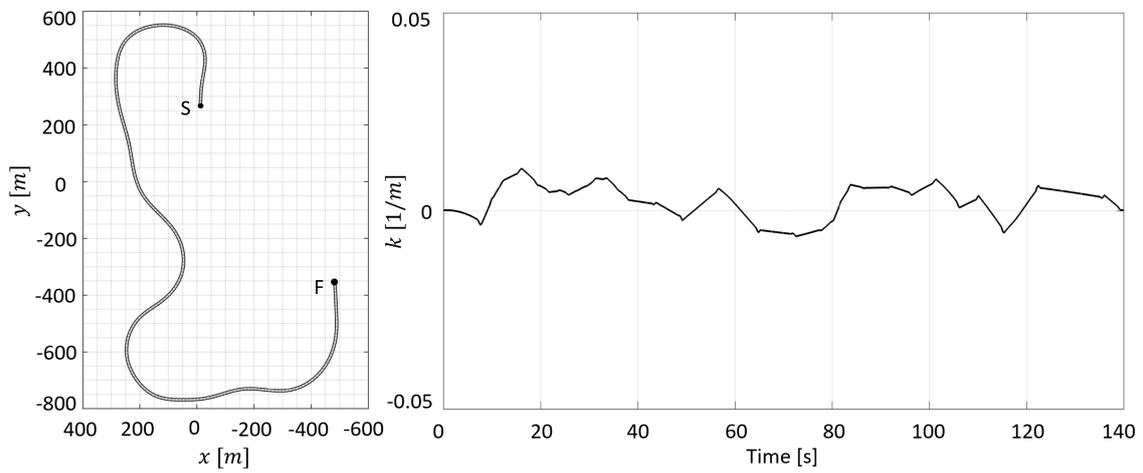


Figure 4.2: (a) Inter urban driving scenario: S is the vehicle's starting point. F is the end of the road. (b) Detected road curvature  $k$

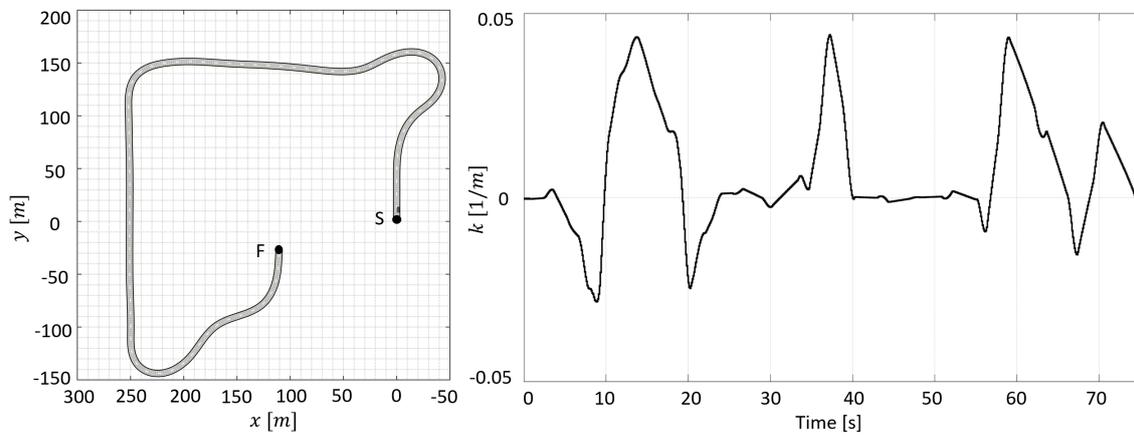


Figure 4.3: (a) Urban driving scenario: S is the vehicle's starting point. F is the end of the road. (b) Detected road curvature  $k$

## 4.2 Results and discussion

In this section the results from the simulations are presented and discussed. For validation purposes two parameters  $e_1$  and  $e_2$  are used. The former gives information about how much the vehicle deviates in the lateral direction from the center line of the lane and the latter how much the vehicle's yaw angle deviates from the desired yaw angle. i.e relative yaw angle. A lateral deviation limit value equal to  $\pm 0.1$  m is considered as acceptable. Similarly, the relative yaw angle should be limited to  $\pm 0.15$  rad. Lateral acceleration is limited using the formula in equation(3.11), which keeps it in the comfort range of ( $\pm 4$   $m/s^2$ ) as seen in GG plot.

### 4.2.1 Highway

The results related to the highway scenario are reported in the Figures depicted below. The longitudinal speed reference  $V_{ref}$  is accurately tracked by the car during the simulation time Figure 4.4. Around 60s the speed decreases as it approaches the highway exit to preserve the lateral stability and keep the lateral acceleration within the comfort range. Lateral deviation  $e_1$  in Figure 4.5 is quite small and stays within  $\pm 0.05$ m, while the error in terms of  $e_2$  is also kept small as seen in Figure 4.6. The GG diagram in Figure 4.9 confirms that the car is driving within the limits of adherence conditions and also the acceleration is within the comfort level.

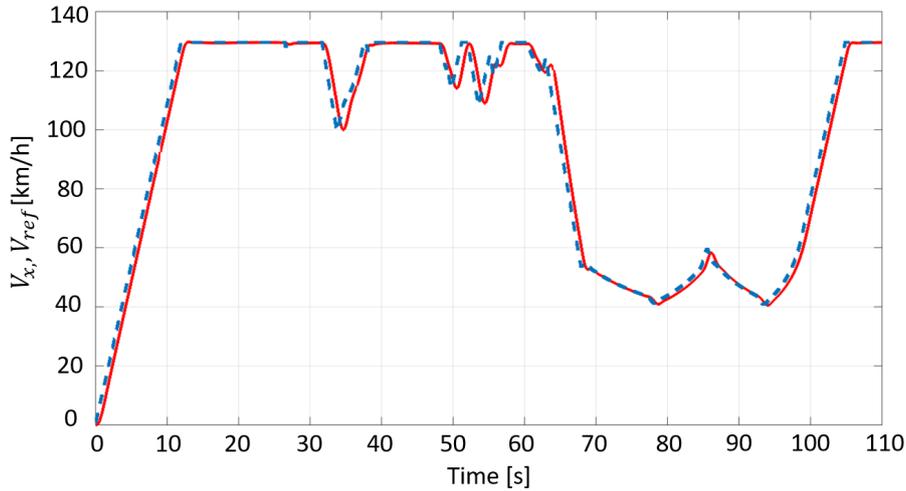


Figure 4.4: Measured vehicle's longitudinal speed  $V_x$  (solid) vs. vehicle's longitudinal speed reference  $V_{ref}$  (dashed)

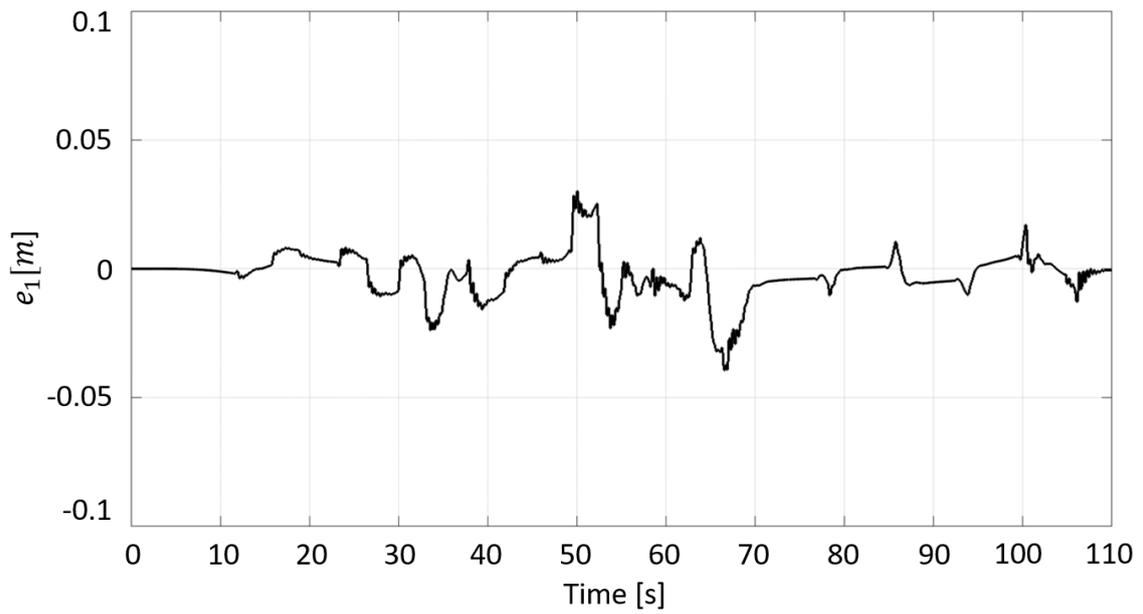


Figure 4.5: Lateral deviation  $e_1$

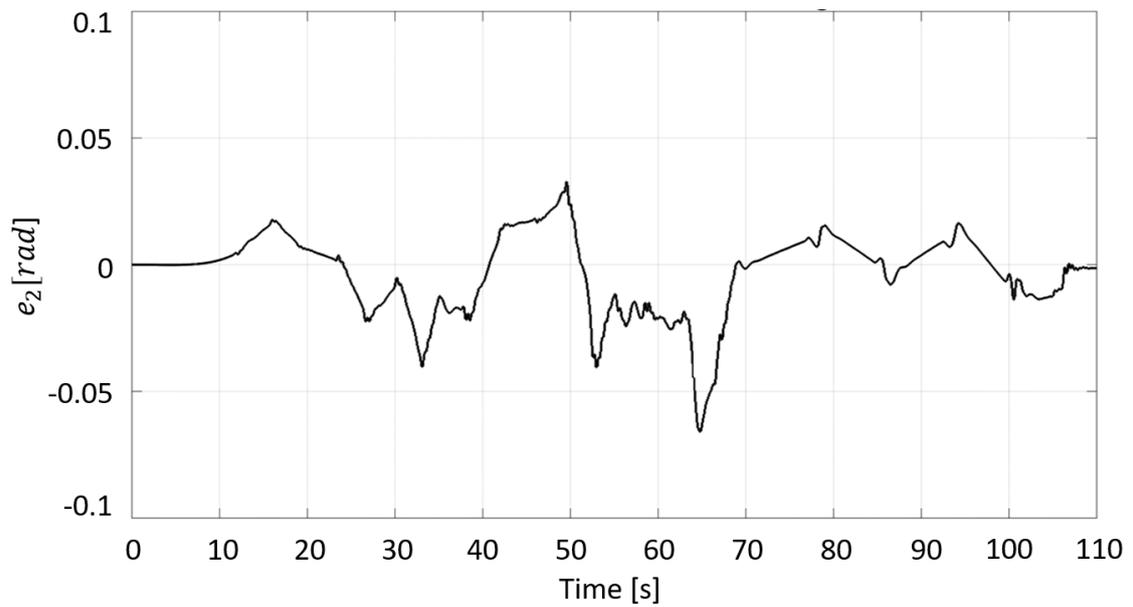


Figure 4.6: Relative yaw angle  $e_2$

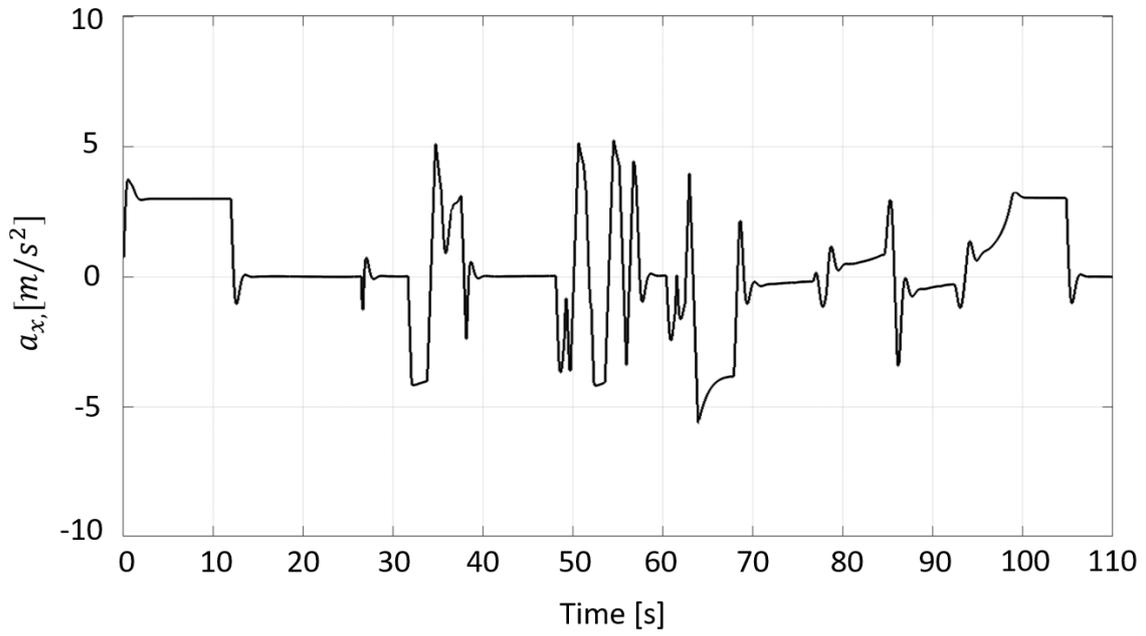


Figure 4.7: Longitudinal acceleration command  $a_x$

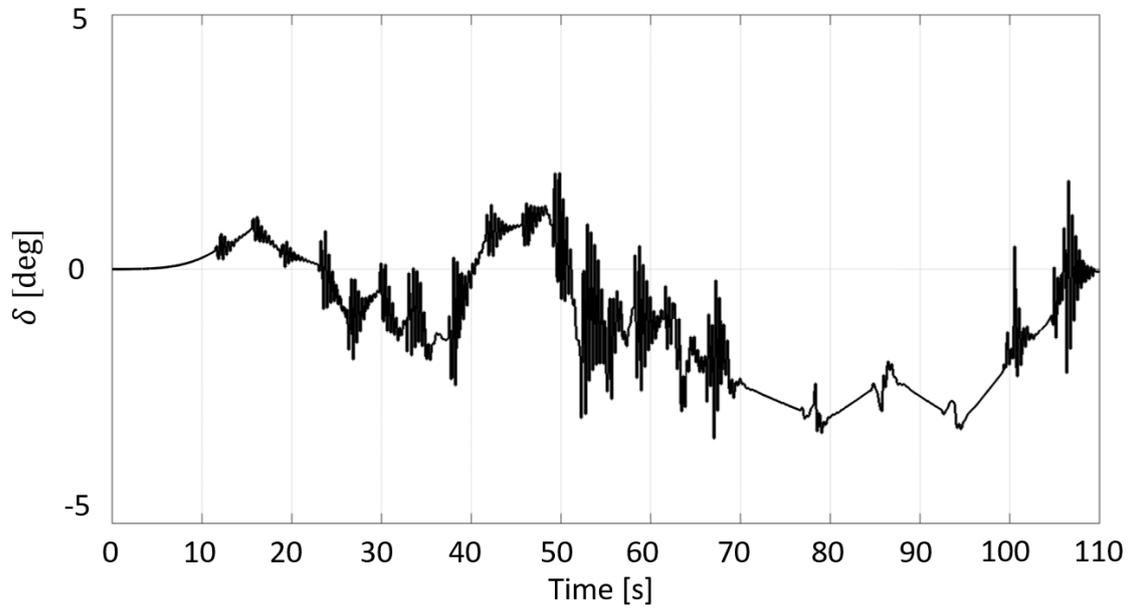


Figure 4.8: Front wheels steering angle command  $\delta$

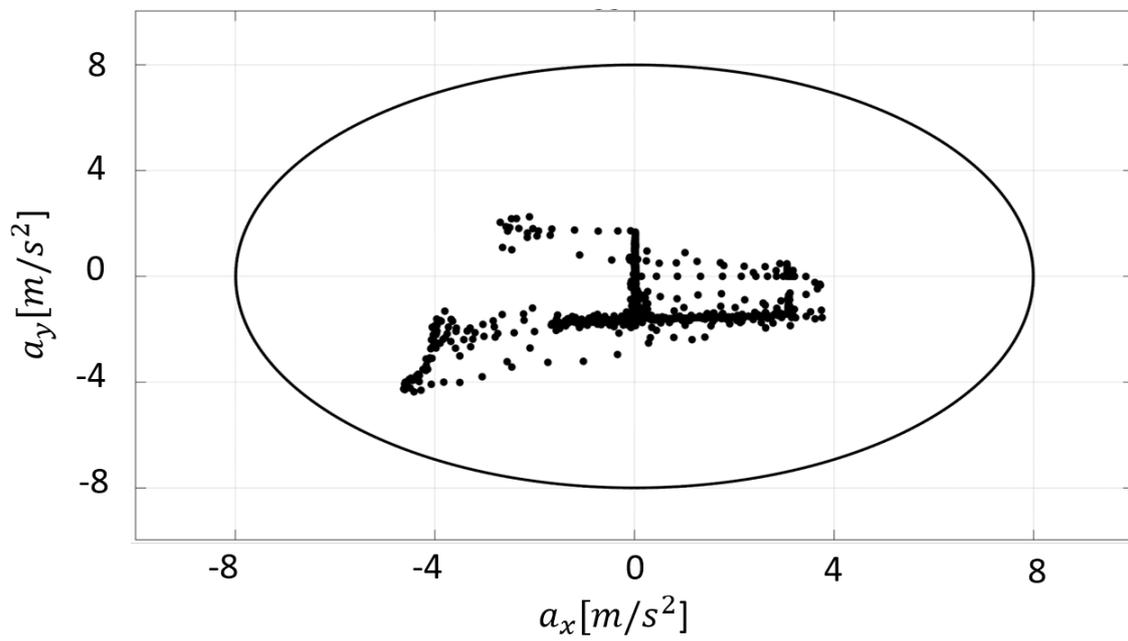


Figure 4.9: GG plot with the ellipse representing the adherence limits

## 4.2.2 Inter-urban

The results related to the inter-urban scenario are reported in the Figures depicted below. Also, in this case, the longitudinal speed reference  $V_{ref}$  is accurately tracked by the car Figure 4.10. A maximum lateral deviation  $e_1$  in Figure 4.11 equal to 0.04m is detected in regions with high curvature turns, while it assumes small values in the remaining part of the simulation ranging from  $\pm 0.03$  m. The error of the relative yaw angle  $e_2$  in Figure 4.12 remains in the admissible range ranging from  $\pm 0.05$  rad. The value of the  $\delta$  in Figure 4.14 varies between  $\pm 3$  degree to follow the center line of the lane. The GG diagram in Figure 4.15 confirms that the car is driving within the limits of adherence conditions and also the acceleration is within the comfort level.

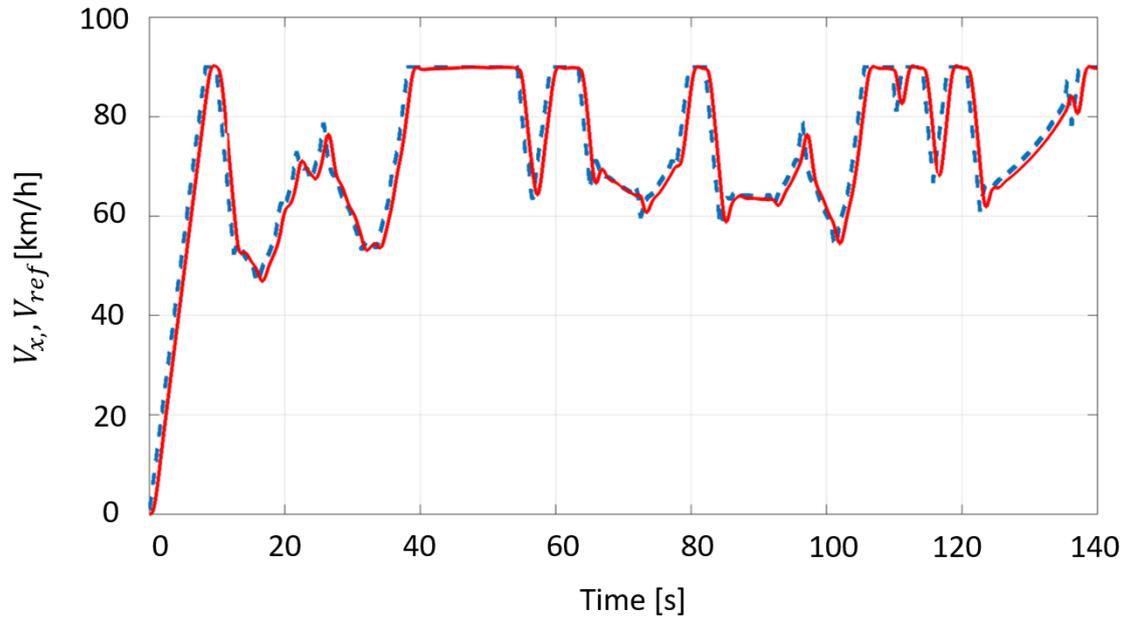


Figure 4.10: Measured vehicle's longitudinal speed  $V_x$  (solid) vs. vehicle's longitudinal speed reference  $V_{ref}$  (dashed)

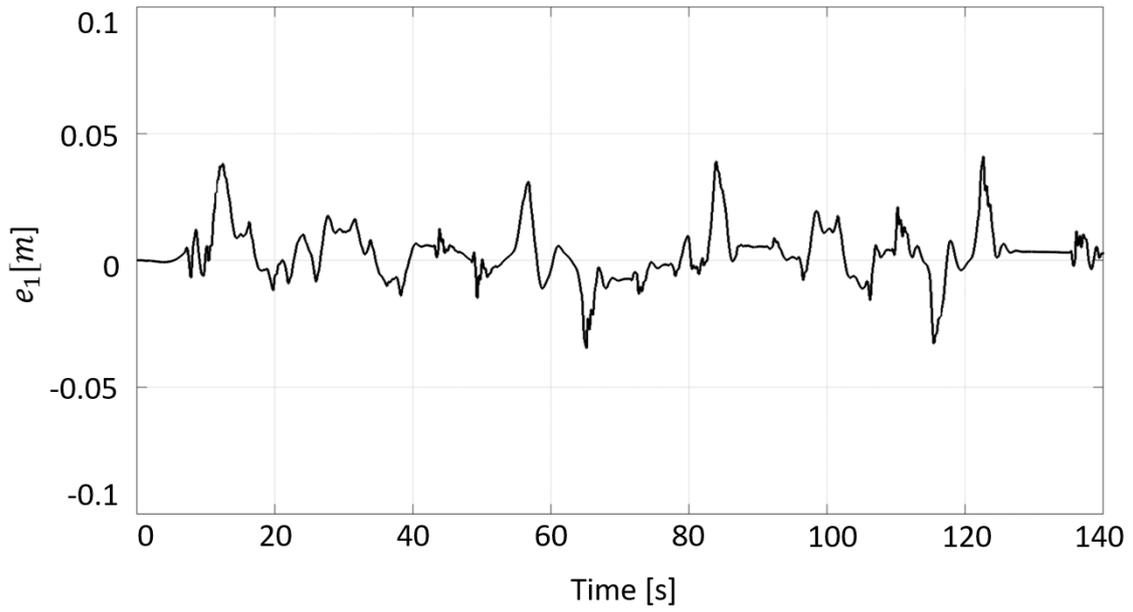


Figure 4.11: Lateral deviation  $e_1$

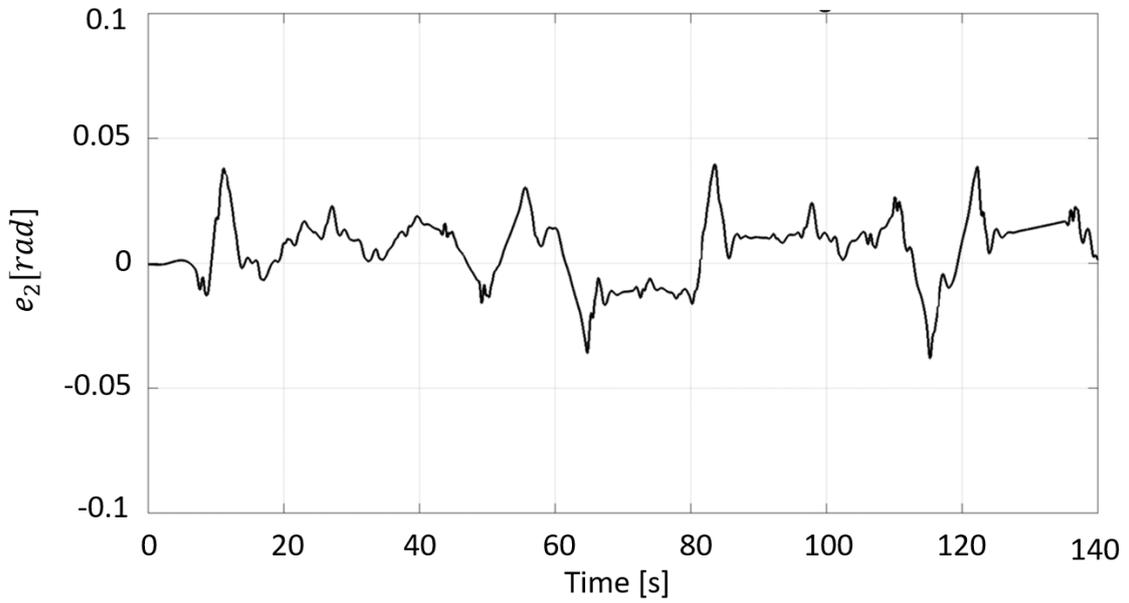


Figure 4.12: Relative yaw angle  $e_2$

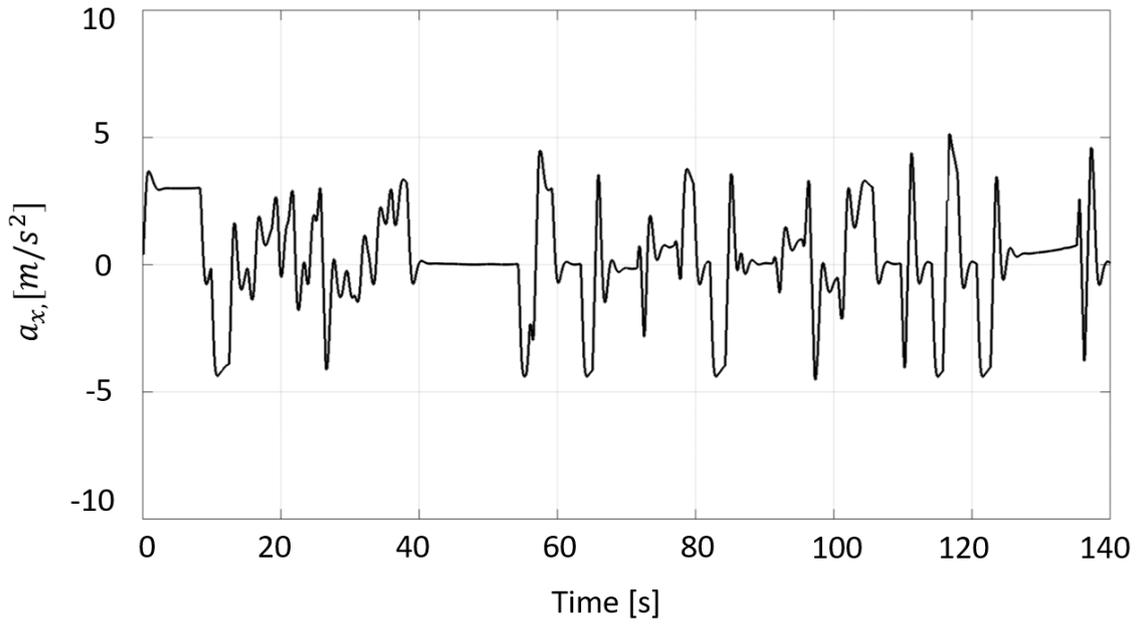


Figure 4.13: Longitudinal acceleration command  $a_x$

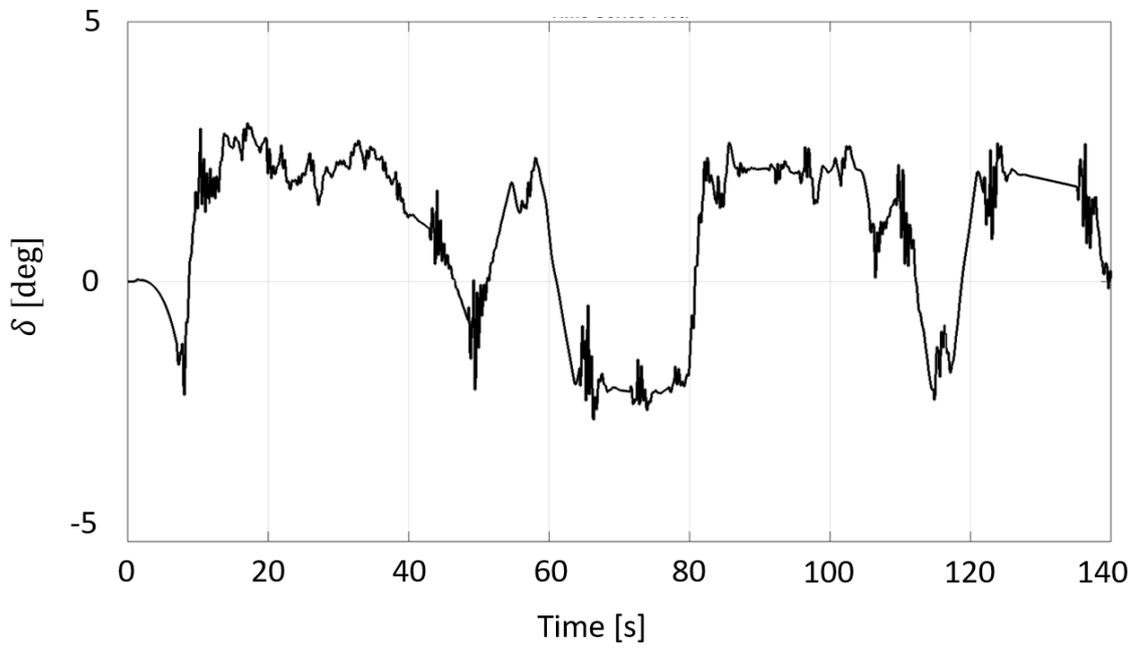


Figure 4.14: Front wheels steering angle command  $\delta$

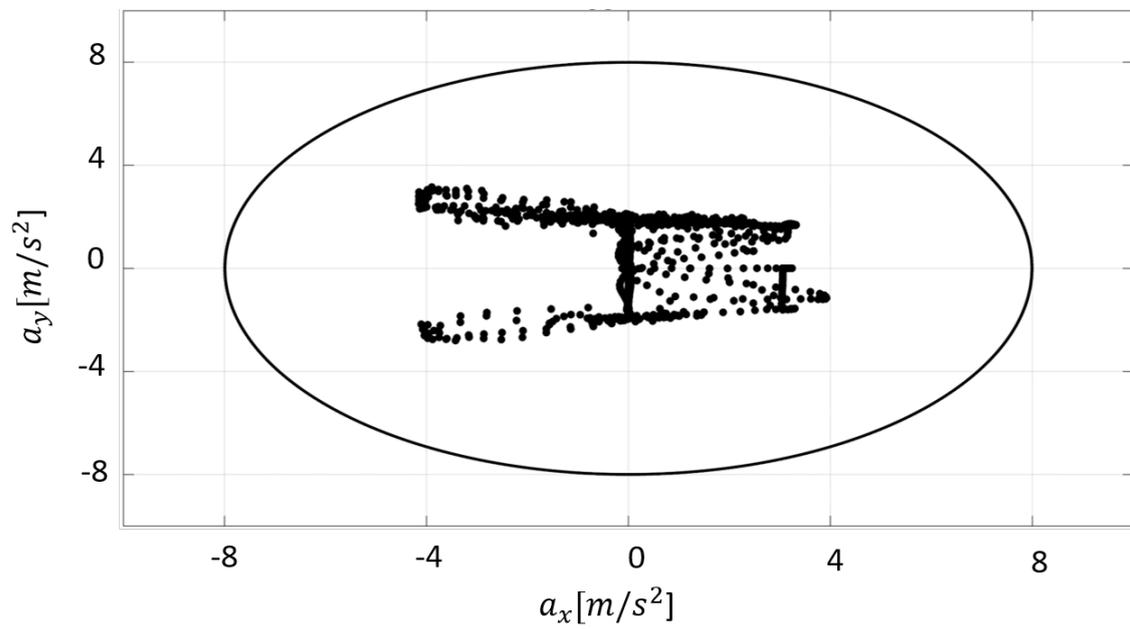


Figure 4.15: GG plot with the ellipse representing the adherence limits

### 4.2.3 Urban

The results relative to the urban scenario are reported in the Figures depicted below. The longitudinal speed reference  $V_{ref}$  is accurately followed by the vehicle, while remaining within the speed limits of 47km/h. A maximum lateral deviation  $e_1$  equal to 0.1 m in Figure 4.17 occurs for a very short period during a sharp turn, where the detected road curvature reaches 0.04 1/m. It corresponds to a road turn with a curvature radius equal to 25 m. The error in terms of relative yaw angle  $e_2$  is high reaching -0.11 rad at the roundabout and sharp turn of 90 degree. While in the rest of the region it varies between +0.05 rad. The value of the  $\delta$  in Figure 4.20 reaches high values of 8 degree to perform the maneuvers of roundabout and 90 degree turns. The GG diagram in Figure 4.21 confirms that the car is driving within the limits of adherence conditions and also the acceleration is within the comfort level.

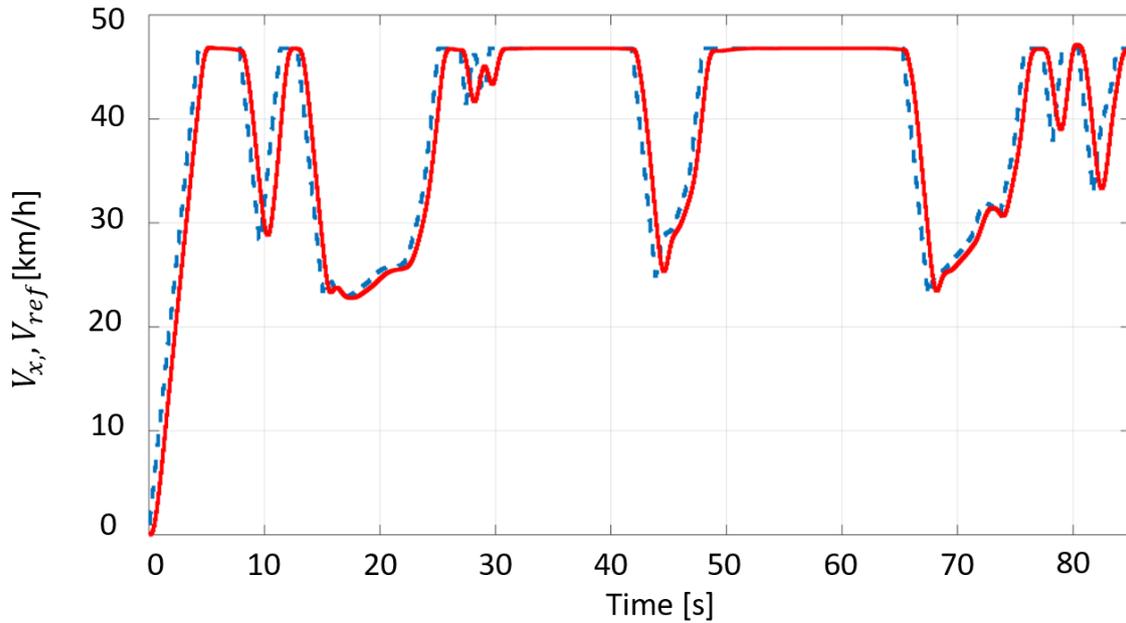


Figure 4.16: Measured vehicle's longitudinal speed  $V_x$  (solid) vs. vehicle's longitudinal speed reference  $V_{ref}$  (dashed)

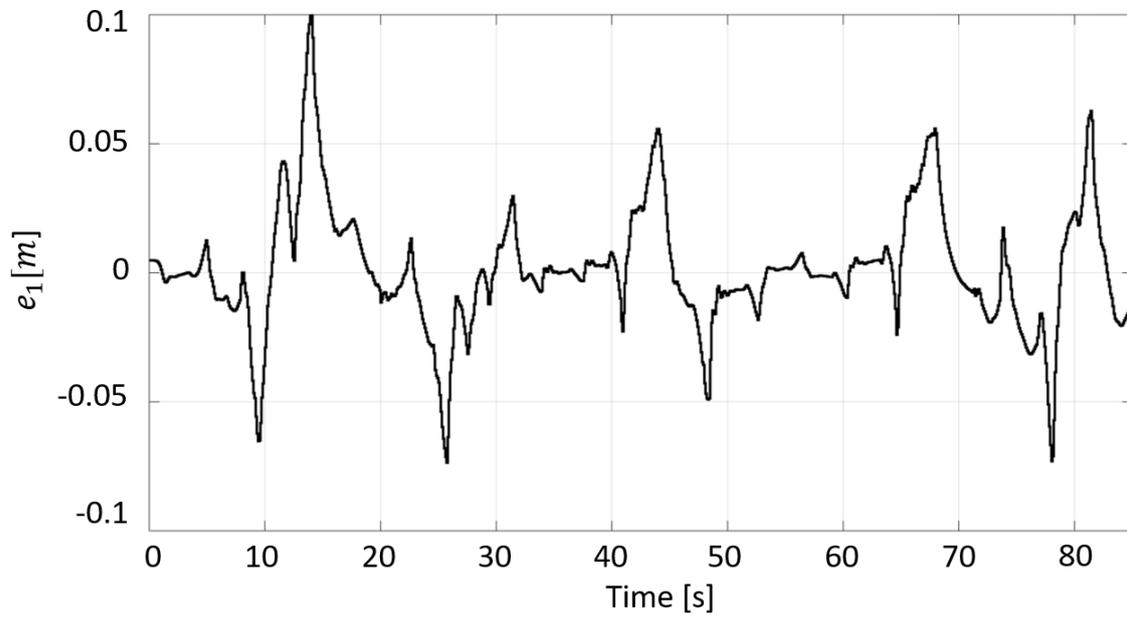


Figure 4.17: Lateral deviation  $e_1$

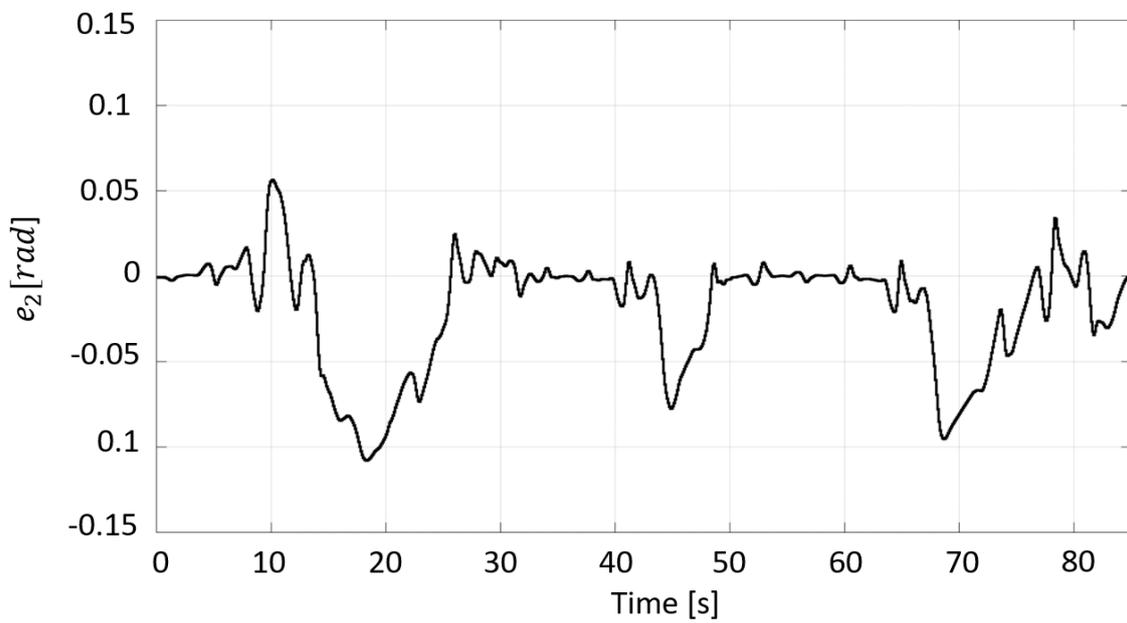


Figure 4.18: Relative yaw angle  $e_2$

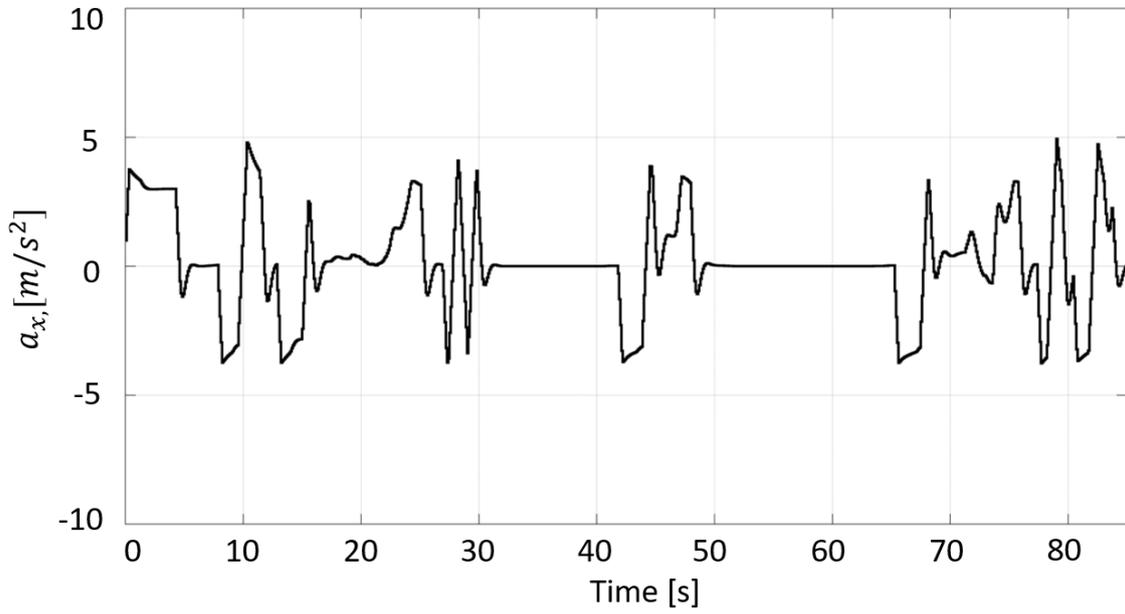


Figure 4.19: Longitudinal acceleration command  $a_x$

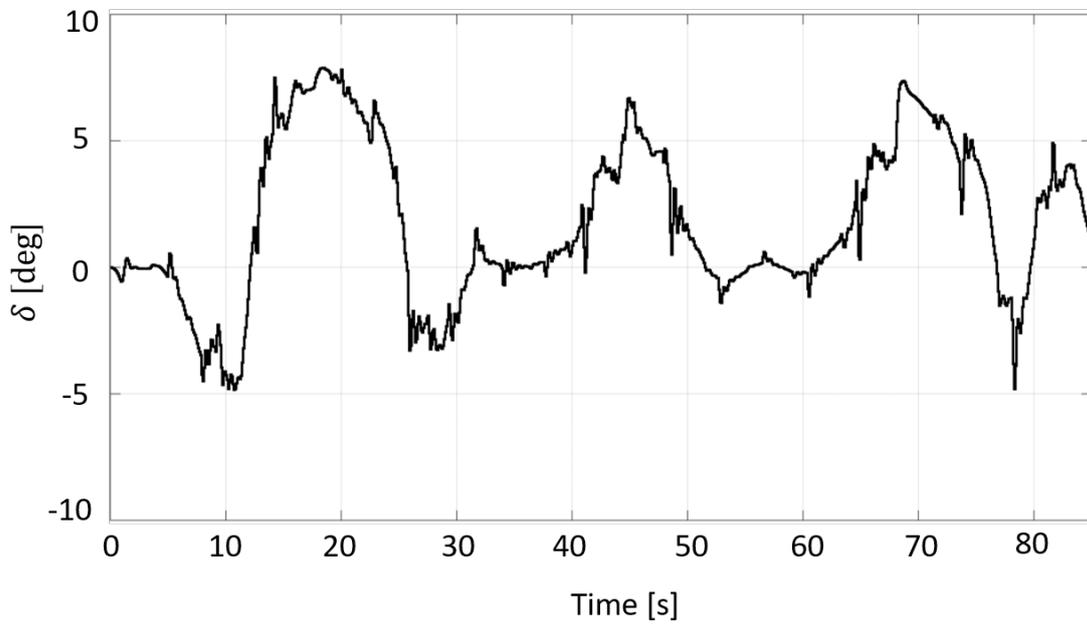


Figure 4.20: Front wheels steering angle command  $\delta$

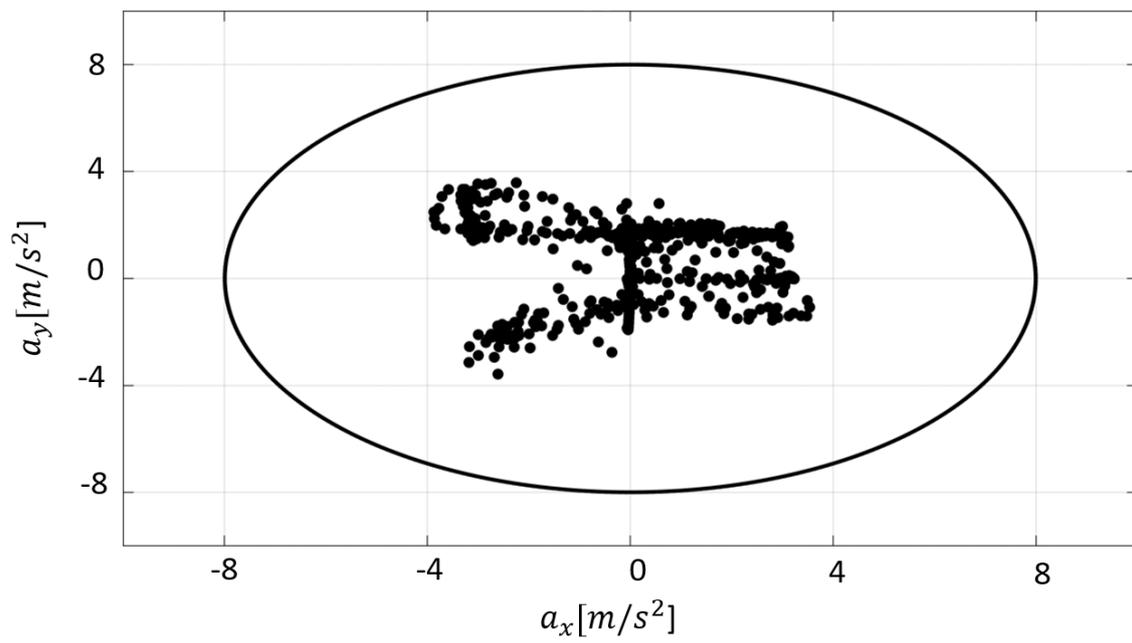


Figure 4.21: GG plot with the ellipse representing the adherence limits

### 4.3 Comparison with another controller based on MPC and PID

In this section, the adaptive MPC developed in this thesis is compared to a controller based on MPC and PID, developed by a student in the LIM for a same scenario representing highway exit Figure 4.22. Later, the results are compared and the advantages of the combined lateral and longitudinal controller respect to the decoupled one are discussed.

The other controller developed for autonomous driving uses an adaptive MPC for the lateral control, while for the longitudinal control a PID controller is used. Which tracks the reference velocity generated using only the road geometry and try to maximize it without any consideration on comfort and lateral stability. So, this control strategy does not take into account the coupling between the lateral and longitudinal dynamics and the problem is solved in a decoupled way.

The results are compared in the Figure 4.23 and Figure 4.24.

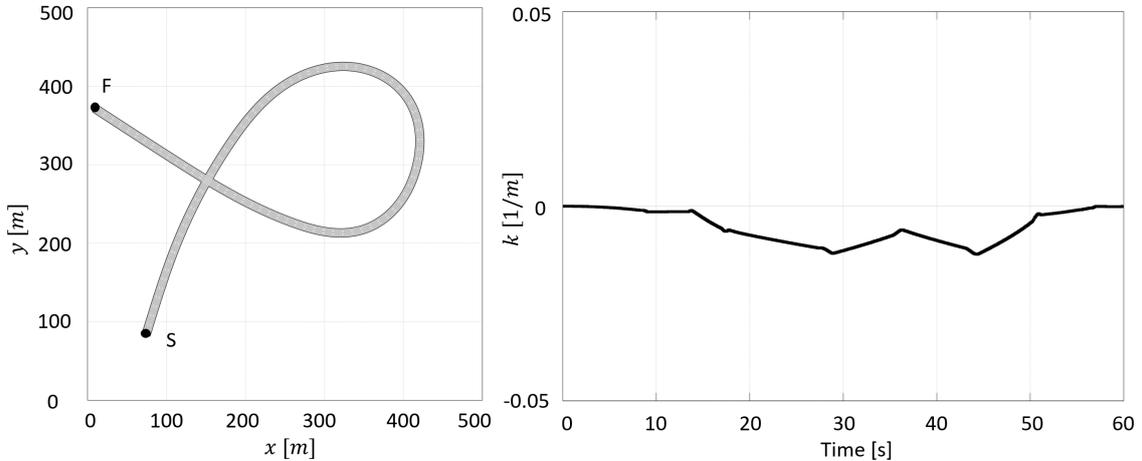


Figure 4.22: (a) Highway exit driving scenario: S is the vehicle’s starting point. F is the end of the road track. (b) Detected road curvature  $k$

Results on the left are from the combined MPC controller, while the ones on the right are from the decoupled MPC and PID controller.

It can be seen that the combined MPC controller can control the car much better and keep the lateral deviation and the relative yaw quite small compared to the

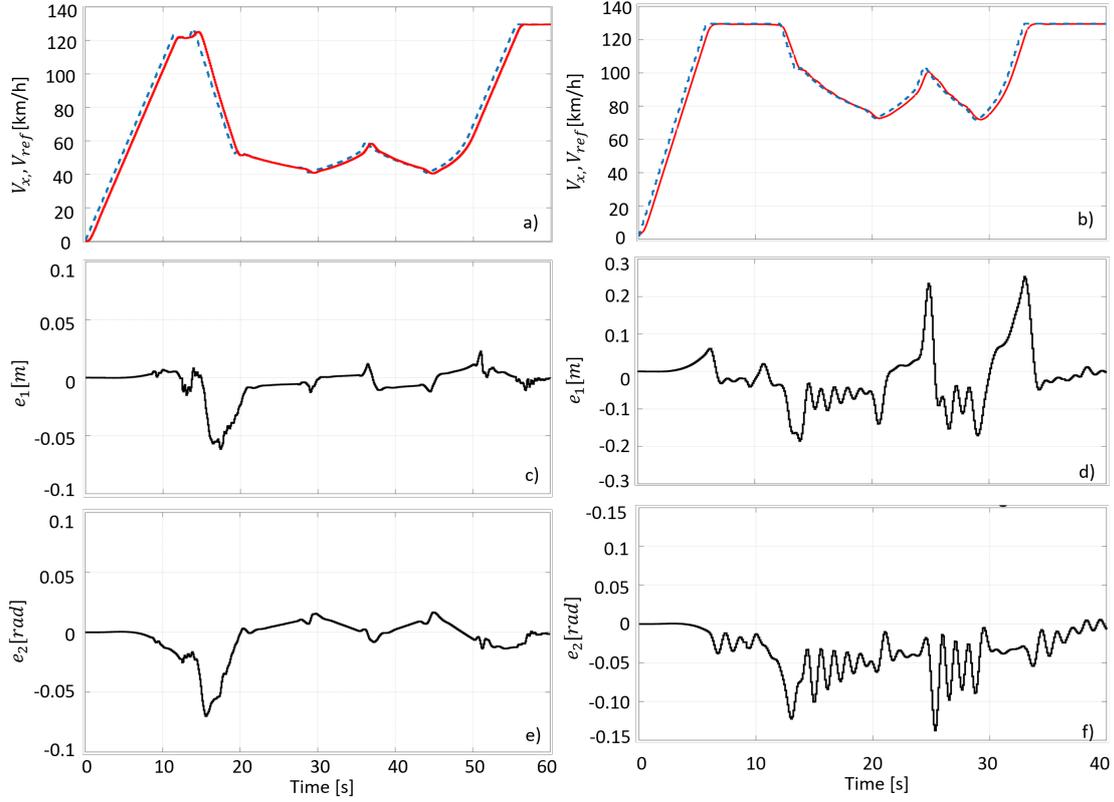


Figure 4.23: Results comparison:(a), (c) and (e) depicts the result of the combined MPC controller, while (b), (d) and (f) depicts the results for decoupled controller based on MPC and PID in terms of  $V_x$  (solid) vs. vehicle's longitudinal speed reference  $V_{ref}$  (dashed), Lateral deviation  $e_1$  and Relative yaw angle  $e_2$

other one based on MPC and PID while maintaining the lateral stability of the vehicle. Maximum lateral deviation Figure 4.23 (c) is 0.05m with the combined MPC but the value of lateral deviation Figure 4.23 (d) reaches 0.25m with the other one. The main difference can be seen in Figure 4.24 (a) and (b), where, the side slip angle  $\beta$  of the vehicle exceeds the limit value of  $\beta_{limit}$  given by equation(3.15) with the other controller. Since, the generation of the reference speed profile is generated using only the road geometry and the vehicle enters the exit of the highway with high velocity which can be seen in Figure 4.23 (b) around 12s. The loss of stability can also be seen in the front wheel steering angle Figure 4.24 (d) at the same moment. From Figure 4.24 (e), the accelerations developed are quite low and are within the comfort driving range for the combined MPC controller. While, in Figure 4.24 (f),

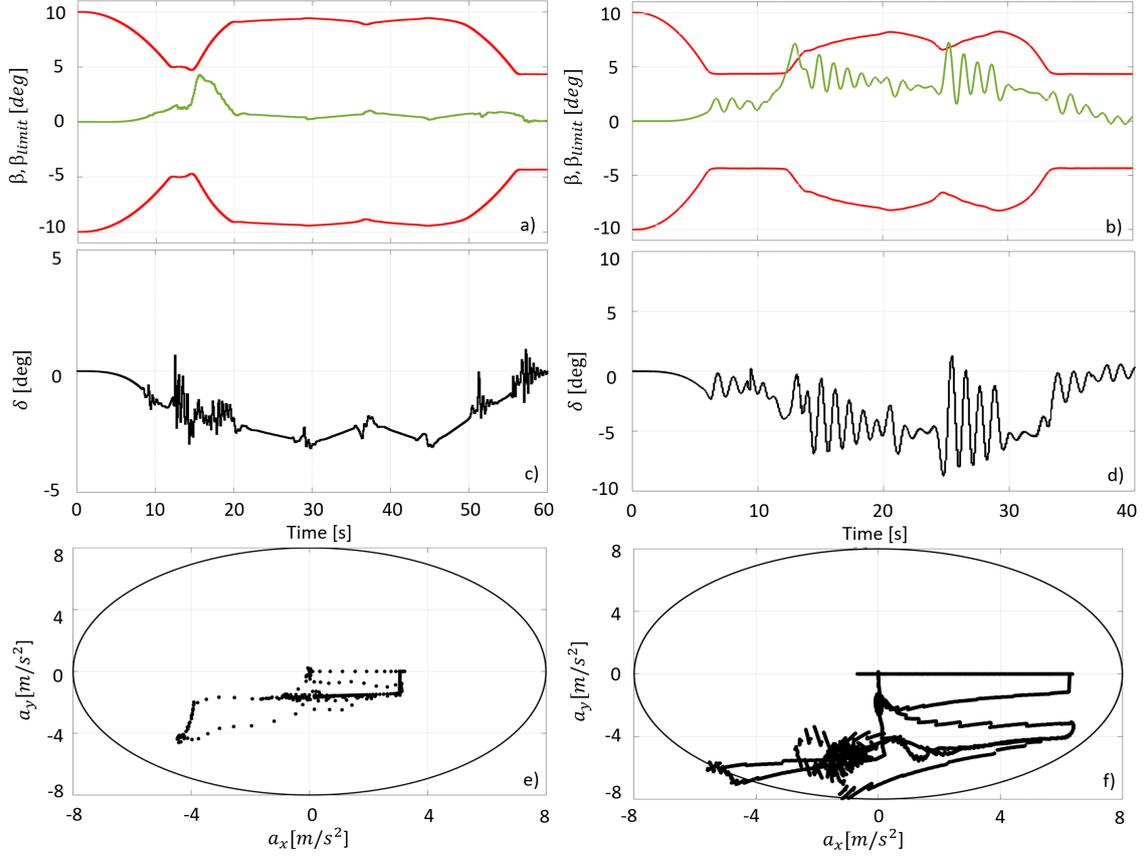


Figure 4.24: Results comparison:(a), (c) and (e) depicts the result of the combined MPC controller, while (b), (d) and (f) depicts the results for decoupled controller based on MPC and PID in terms of vehicle side slip angle  $\beta$  with  $\beta_{limit}$ , Front wheels steering angle command  $\delta$  and GG plot with the ellipse representing the adherence limits

the acceleration developed in the other case are quite higher and well beyond the comfort range. We can say this controller is designed more towards racing context rather than normal driving.

Overall, the controller designed in this thesis perform quite well by tracking the reference velocity accurately and keeping the errors, in terms of lateral deviation and relative yaw angle in the admissible limits. It also preserves the lateral stability of the vehicle and keeps the acceleration with in the comfort range.

# Chapter 5

## Conclusions and future works

In this thesis work, a combined lateral and longitudinal control strategy for autonomous driving was presented. To this end an adaptive MPC control was exploited, allowing to minimize the errors on the controlled variables, which are the lateral deviation, the relative yaw angle and the longitudinal speed of the vehicle w.r.t the reference longitudinal speed by acting on the steering angle of the front wheels and throttle/brake pedals. The strategy has been tested through simulation on MATLAB and Simulink with three driving scenarios namely: Highway, Inter-urban and Urban driving and provides good performance for lateral guidance and accurately follows the reference speed profile. The lateral deviation was kept within the acceptable range of  $\pm 0.1$  m for all the scenarios and also relative yaw angle was within the limit of  $\pm 0.15$  rad. Regarding comfort, lateral and longitudinal acceleration of the vehicle were well within the range. Finally, it was compared to another controller based on MPC and PID, which tackle the problem in decoupled way. From the results, it was seen that combined MPC controller was able to control the car in a much better way and kept the lateral deviation and the relative yaw quite small compared to the other one based on MPC and PID while maintaining the lateral stability of the vehicle.

Some future works can be done to improve and extend this thesis work. One extension of this thesis would be to implement the controller to a real vehicle. However, testing on real vehicles is expensive and often time consuming. So, it would be of interest to develop the Four Wheel Simulink model first and do further tests with it

before implementing to a real vehicle. One aspect to improve with the Four Wheel model is the driveline dynamics and tire modelling. For this thesis we have chosen to use a simpler model to predict the behaviour of a more advanced system. This was done to keep the complexity of the prediction model low and hence avoid too high computational expenses. However, the longitudinal dynamics can be modelled by adding driveline and tire dynamics to the prediction model, so that the behaviour of the cruise controller could be improved. It would be interesting to do so and compare the result regarding the computational burden and the accuracy with the prediction model used in this thesis.

In this thesis the trajectories were generated without considering other vehicles, by considering other vehicles when the trajectories are generated the cooperation between vehicles could be improved. One way of doing this would be to generate the references inside the control loop. However, this would increase the computational expenses for the control algorithm.

Some other future works can also be done to improve and extend this thesis work in the field of perception. First of all, in order to overcome the limitation of the lane detection function using the camera (crossroads or roads without lane marking), data coming from others sensors will be added, such as the data coming from a LiDAR and GPS. Making sensor fusion between camera and LiDAR, the detection will be improved in challenging scenarios. For the development of an autonomous driving vehicle, the lane detection will be combined with others detection systems such as vehicles, pedestrians, semaphores, traffic signs and road texts detection.

To conclude, this thesis has contributed for autonomous vehicle research at Mechatronics Laboratory LIM (Laboratorio Interdisciplinare di Meccatronica) and the developed project can be used by future students to improve and continue the work in this interesting field.

# Bibliography

- [1] Steer Davies Gleave: Roberta Frisoni, Andrea Dall'Oglio. *Research for TRAN Committee – Self-piloted cars: The future of road transport?*. European Parliament's Committee on Transport and Tourism. , 2016.
- [2] European Road Safety Observatory (ERSO). *Advanced driver assistance systems*. European Commission, 2018.  
Web link:<http://www.erso.com>.
- [3] SAE International On-road Automated Vehicle Standards Committee (2014) *SAE J3016: Taxonomy and definitions for terms related to on-road motor vehicle automated driving systems*.
- [4] Amer, Noor Hafizah, et al. *Modelling and control strategies in path tracking control for autonomous ground vehicles: a review of state of the art and challenges*. Journal of Intelligent and Robotic Systems 86.2 (2017): 225-254.. ICCCE 2008. International Conference on, 2008, pp. 82–88.
- [5] M. Bujarbaruah, X. Zhang, H. E. Tseng and F. Borrelli. *Adaptive MPC for Autonomous Lane Keeping*. 14th International Symposium on Advanced Vehicle Control (AVEC), Beijing, China, July 2018.
- [6] Y. Xu, B. Y. Chen, X. Shan, W. H. Jia, Z. F. Lu, G. Xu. *Model Predictive Control for Lane Keeping System in Autonomous Vehicle*. International Conference on Power Electronics Systems and Applications (PESA), IEEE, 2017, pp. 1-5.
- [7] V. Turri, A. Carvalho, H. E. Tseng, K. H. Johansson, F. Borrelli. *Linear Model Predictive Control for Lane Keeping and Obstacle Avoidance on Low Curvature Roads*. 16th International IEEE Conference on Intelligent Transportation System (ITSC), The Hague, The Netherlands, October 2013, pp. 378-383.
- [8] Xiao L, Gao F. *A comprehensive review of the development of adaptive cruise control systems*. Veh Syst Dyn.2010;48(10):1167–1192.

- [9] Takahama, Taku, and Daisuke Akasaka. *Model Predictive Control Approach to Design Practical Adaptive Cruise Control for Traffic Jam*. International Journal of Automotive Engineering 9.3 (2018): 99-104.
- [10] Naus, G. J. L., et al. *Design and implementation of parameterized adaptive cruise control: An explicit model predictive control approach*. Control Engineering Practice 18.8 (2010): 882-892.
- [11] Katriniok, A., Maschuw, J.P., Christen, F., Eckstein, L., Abel, D. (2013). *Optimal vehicle dynamics control for combined longitudinal and lateral autonomous vehicle guidance*. 2013 European Control Conference (ECC).
- [12] Attia, Rachid, Rodolfo Orjuela, and Michel Basset. *Coupled longitudinal and lateral control strategy improving lateral stability for autonomous vehicle*. 2012 American Control Conference (ACC). IEEE, 2012.
- [13] Song, P., Zong, C., and Tomizuka, M., *Combined Longitudinal and Lateral Control for Automated Lane Guidance of Full Drive-by-Wire Vehicles*. SAE Int. J. Passeng. Cars – Electron. Electr. Syst. 8(2):2015
- [14] Wikipedia [Online]. *Kinematics*. 2018.  
Web link: <https://en.wikipedia.org/wiki/Kinematics>.
- [15] D. Wang and F. Qi. *Trajectory planning for a four wheel steering vehicle*. IEEE International Conference on Robotics and Automation, Seoul, Korea, May 21-26, 2001.
- [16] MATLAB [Online]. *Vehicle Body 3DOF*. 2018.  
Web link: <https://it.mathworks.com/help/vdynblks/ref/vehiclebody3dof.html>.
- [17] Pacejka, Hans B. (2006). *Tyre and vehicle dynamics* (2nd ed.). SAE International. pp. back cover.
- [18] Rajamani R., *Vehicle Dynamics and Control*, Mechanical Engineering Series, 2012.
- [19] MATLAB [Online]. *Visual Perception Using Monocular Camera*. 2018.  
Web link: <https://it.mathworks.com/help/driving/examples/visual-perception-using-monocular-camera.html>.
- [20] Wikipedia [Online]. *Feature detection (computer vision)*. 2018.  
Web link: [https://en.wikipedia.org/wiki/Feature\\_detection\\_\(computer\\_vision\)](https://en.wikipedia.org/wiki/Feature_detection_(computer_vision)).

- [21] S. D. Pendleton, H. Andersen, X. Du, X. Shen, M. Meghjani, Y. H. Eng, D. Rus and M. H. Ang. *Perception, Planning, Control, and Coordination for Autonomous Vehicles*. Machines, 2017.
- [22] MATLAB [Online]. *segmentLaneMarkerRidge*. 2018.  
Web link: <https://it.mathworks.com/help//driving/ref/segmentlanemarkerridge.html>.
- [23] M. Nieto, J. A. Laborda and L. Salgado. *Road Environment Modeling Using Robust Perspective Analysis and Recursive Bayesian Segmentation*. Machine Vision and Applications, 2011.
- [24] A. B. Hillel, R. Lerner, D. Levi and G. Raz. *Recent Progress in Road and Lane Detection: A survey*. Machine Vision and Applications, Vol. 25, 2014, pp. 727–745.
- [25] Wikipedia [Online]. *Random sample consensus*. 2018.  
Web link: [https://en.wikipedia.org/wiki/Random\\_sample\\_consensus](https://en.wikipedia.org/wiki/Random_sample_consensus).
- [26] K. G. Derpanis. *Overview of the RANSAC Algorithm*. Image Rochester NY, Vol 4, 2010, pp. 2–3.
- [27] MATLAB [Online]. *findParabolicLaneBoundaries*. 2018.  
Web link: <https://it.mathworks.com/help/driving/ref/findparaboliclaneboundaries.html>
- [28] Mancuso. A. *Lane Detection and Lane Keeping for Autonomous Driving Vehicles*. Master’s degree thesis, Politecnico di Torino 2018.
- [29] J. W. Rutter. *Geometry of Curves*. Chapman & Hall/CRC, 2000.
- [30] Xu, Jin, et al. *An experimental study on lateral acceleration of cars in different environments in Sichuan, Southwest China*. Discrete Dynamics in nature and Society 2015.
- [31] J. Richalet, A. Rault, J. Testud and J. Papon. *Model predictive heuristic control: Applications to industrial processes*. Automatica, vol. 14, no. 5, pp. 413–428, 1978.
- [32] C. Cutler and B. Ramaker. *Dynamic Matrix Control - A Computer Control Algorithm*. Automatic Control Conference, San Francisco, CA, 1980.
- [33] S. J. Qin and T. A. Badgwell. *A Survey of Industrial Model Predictive Control Technology*. Control Eng. Practice, 11, 733, 2003.
- [34] F. Borrelli, A. Bemporad and M. Morari *Predictive Control for linear and hybrid*

- systems*. Cambridge University Press, Cambridge In preparation, 2015
- [35] D. E. Seborg, D. A. Mellichamp, T. F. Edgar, F. J. Doyle. *Process Dynamics and Control*. John Wiley and Sons, 2011.