POLITECNICO DI TORINO

Master's Degree in Automotive Engineering
A.a. 2021/2022
December 2022 Graduation Session

Autonomous Parking Maneuvers via Non-Linear Model Predictive Control

Supervisors

Prof. Carlo NOVARA

Eng. Mattia BOGGIO

Candidate

Michel JREIGE

Abstract

Autonomous driving is considered one of the most ground-breaking technologies of the near future that will completely reshape transportation systems. In this regard, more and more research efforts are being spent by automotive companies and academic institutions for developing vehicles with an ever-higher level of autonomy. Thanks to well-known environments and relatively low risks, particular attention has been devoted to Autonomous Parking. The aim is to simplify as much as possible the actions required by the driver to complete the parking, reducing time (to perform all the maneuver) and spaces. Indeed, the increase in the number of vehicles entails the need for even narrower spots, thus rendering manual parking operations more challenging. Furthermore, unskilled parking abilities may cause traffic jams. Modern control theory offers a multitude of approaches and design paradigms that can be exploited for this application. Among them, Nonlinear Model Predictive Control (NMPC) has the potential to become a key technology. Certainly, it: i) only requires a target point, ii) deals with linear and nonlinear constraints, iii) jointly performs trajectory planning and control. In that context, this thesis aims to develop a general NMPC framework capable of performing several parking maneuvers. More in detail, a suitable configuration of the parameters characterizing the NMPC has been found such that, providing different initial poses and targets, it is always able to generate the optimal trajectory and commands to guide the ego vehicle into the parking zone. Furthermore, to avoid collisions, Heaviside-based constraints have been used. Finally, to prove the effectiveness and robustness of the developed NMPC framework, a Monte Carlo campaign has been carried out considering different initial conditions. In all the performed tests, the NMPC succeeded in entering the parking zone without any collision.

Acknowledgements

During this thesis conducted on the premises of Politecnico di Torino, I took great pleasure in getting involved in the field of control and automotive engineering. I would like to express my gratitude to the people who made this course a reality.

With great gratitude, that I wish to thank Prof. Carlo Novara for his support and willingness throughout all the steps of this thesis. Besides, I am particularly grateful for the assistance and patience given by my advisor Eng. Mattia Boggio. His help was indispensable for the progress and feasibility of the thesis. Without their numerous advices and their constant supervision from the beginning to the end this thesis would not have been completed.

I am also thankful to the Lebanese University Faculty of Engineering Branch II and Politecnico di Torino, with all their faculty staff and professors, who gave me the tools and prepared me for the future ahead.

I would like also to extent my thanks to my friends and colleagues of PoliTO and Roumieh, with whom I shared this journey, and that made university feel like a second home.

A special thanks to my flatmates, for being good italian teachers, great listeners, and for always being their during the highs and the lows.

I would like to thank the closest people to my heart, for listening to my endless complaints, offering their sincerest advices, and on which I can always count.

Last but not least, I would like to express my deepest gratitude to my family for the unconditional love, support, and guidance they never failed to provide throughout the years. None of this would be possible without them.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 General

In the last few years, original equipment manufacturers (OEM) started rolling out commercial cars with the possibility of one or more Advanced Driver Assistance Systems (ADAS). This comes as the fruit of huge investments and research done on the subject. This was possible to do so because of the reliability and accuracy of these systems, or otherwise they would have not made it to the market. With an expanding application for ADAS and control systems for vehicles and its increasing reliability, it is only meant to increase in presence in everyday car, for example, in Europe, by 2025, it is expected that 100% of new vehicles will have at least one technology of this sort, despite the effect of the pandemic that slowed it down, [1].

Perhaps the most internationally recognized classification of automation levels is the Society of Automotive Engineers' (SAE) classification which is shown in Fig.1.1. This classification divides driving automation into 6 levels, starting from level 0, i.e., no automation whatsoever to level 5, meaning no need for a driver.

Level 0: The human driver does all the driving.

Level 1: The driver and the automated system share control of the vehicle. They provide control over steering or longitudinal dynamics. Examples are Adaptive Cruise Control, where the driver controls steering and the automated system controls speed; and Parking Assistance, where steering is automated while speed is under manual control. The driver must be ready to retake full control at any time.

level 2: An advanced driver assistance system (ADAS) on the vehicle can actually control both steering and braking/accelerating simultaneously under some circumstances. The human driver must continue to pay full attention, i.e., monitor the driving environment at all times and perform the rest of the driving task. In fact, contact between hand and wheel is often mandatory during SAE 2 driving, to confirm that the driver is ready to intervene.

Level 3: The driver can safely turn their attention away from the driving tasks, e.g. the driver can text or watch a movie. The vehicle will handle situations that call for an immediate response, like emergency braking. The driver must still be prepared to intervene within some limited time, specified by the manufacturer, when called upon by the vehicle to do so.

Level 4: An Automated Driving System (ADS) on the vehicle can perform all driving tasks and monitor the driving environment, in simpler terms do all the driving but in specified conditions. Self driving in fact is supported only in under these special circumstances. Outside of these circumstances, the vehicle must be able to safely abort the trip, i.e. park the car, if the driver does not retake control.

Level 5: No human intervention is required at all.

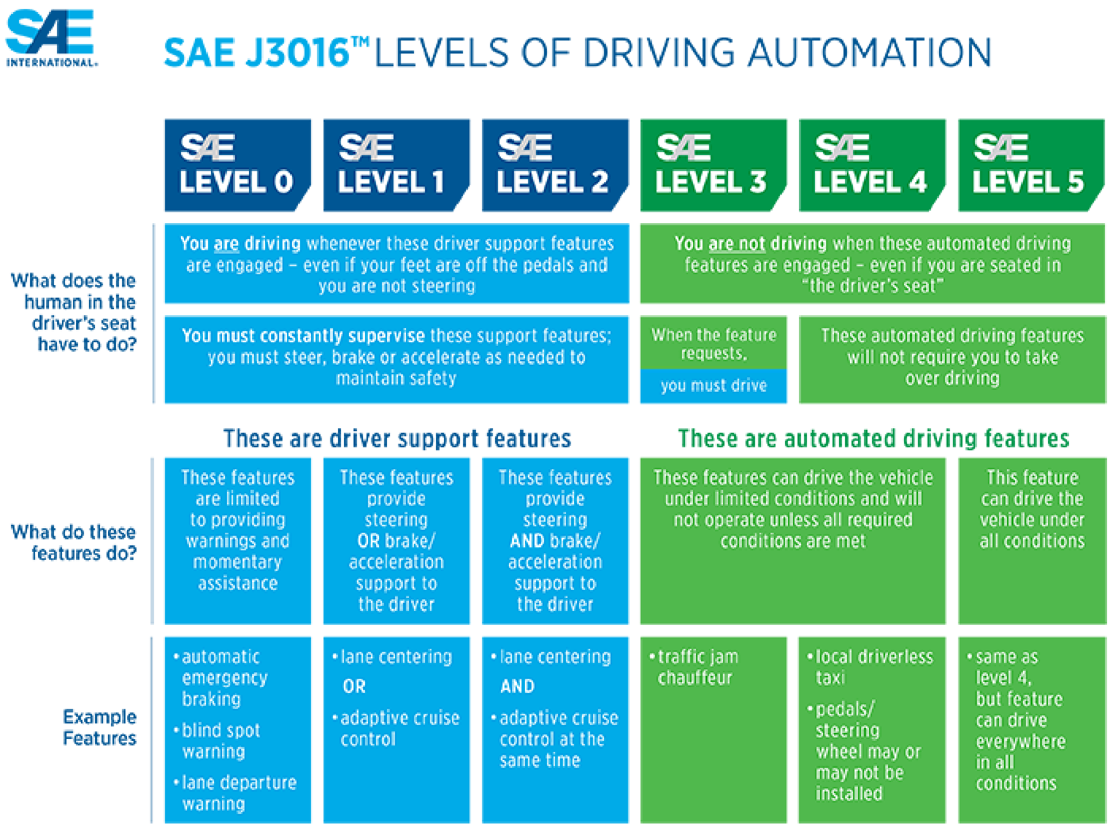

Figure 1.1: SAE's level of automation [2]

At the current moment, it is still a long journey before reaching SAE level 5, if it is attainable, as some experts argue, or even SAE level 4. While commercial solutions are more dispersed between level 2 and level 3, with the majority in the former. Furthermore, in a coarse way, an autonomous driving system can be basically defined by three elements: perception, localization, and control. Each of these has different possible solutions.

Of course, perception is done using a combination of different technologies to be able to correctly detect the road and environmental conditions, e.g., radar, ultrasound sensors, LiDAR, cameras, etc. Some car makers rely on lower cost sensors like cameras rather than LiDARs and compensate for it using machine learning. Majority of available autonomous driving system are intended in the framework of ADAS that can substitute, under certain conditions, the driver for specified functions.

1.2 Autonomous Parking: State of art

Nowadays, many car manufacturer have advanced self-parking technology implemented into a number of their models as part of a wider ADAS system. This technology can be divided into parking assist systems or autonomous parking systems. The difference between these two is that the former aid the driver while he manually parks the car, and that the latter can park the vehicle with little to no aid from the driver. In 2003, Toyota was the first manufacturer to offer an automated parking feature in her hybrid vehicle, the Prius, [3]. After her was Lexus with its Lexus LS sedan in the year 2006, [4]. And others followed them, Ford in 2009, BMW in 2010, etc. With little to no information published by the manufacturers on how their systems works and what kind of topology is used to determine the trajectory and commands, different third parties, from bloggers to independent research centers, classified the available models in the market today. From these classifications, a few models persist on being among the best: BMW Series 5, DS 7, Mercedes-Benz C-Class, KIA EV6, Volkswagen Golf...[5]

1.3 Contribute

The aim of this thesis, realized under the supervision of Prof. Carlo Novara and Eng. Mattia Boggio, is to study the capability of an NMPC-based algorithm (Nonlinear Model Predictive Control) in realizing different parking maneuvers assuming measured vehicle's state and obstacles perception.

Therefore, it has tested the ability of the NMPC controller to generate the proper commands (steering angle and speed of the vehicle) in order to lead the car into the parking zone and reach the target point without trespassing adjacent zones and constraints. Each maneuver is divided into two parts for the purpose of making it possible for the NMPC to succeed in its task. To add, the conversion of the sensors incoming information into constraint and target points for the simulation system, could be assumed as given. The testing of the controller was done using MATLAB® software, and the simulation environment was Simulink®, in which different relative feature of interests can be found.

The thesis has been organized in 4 main sections. The first being about the vehicle model and its use along with the main parameters chosen for it. It is worthy to note that the model used to predict the system behavior is different from the model of this first paragraph, to be able to better represent a real case.
The second paragraph talks about the NMPC, from an introduction to state-of-the-art application passing by the mathematical formulation and the meaning of the different parameters inside the controller.
The third part defines each case study and describes briefly how each case was set up including definition of the constraints and target points.
The fourth and last part deals with how the controller was tuned and how the results were obtained for each case study with a few comments.

Chapter 2

Car Model

This chapter presents the mathematical model used for the Ego vehicle. This model has an important role as it replaces the vehicle in a real life situation and allows to test the NMPC as if it was in real life situation. This section is based on the MATLAB® documentation, [6].
The model used is the "Vehicle Body 3DOF" available in the "Vehicle Dynamics Blockset" in Simulink®. The Vehicle Body 3DOF block implements a rigid two-axle vehicle body model to calculate longitudinal, lateral, and yaw motion. The block accounts for body mass and aerodynamic drag between the axles due to acceleration and steering. It can be used for modeling nonholonomic motion when pitch, roll, and vertical motion are not significant. There are two types of Vehicle track setting for this model, one assuming that forces act along the center line at the front and rear axles and that there is no lateral load transfer, called Single (bicycle), and the other considering forces act at the four vehicle corners, called Dual. The one used during the simulations is the Vehicle Body 3DOF Single Track, as it satisfies the requirements needed. Moreover, the "Axle Forces Setting" is to "External longitudinal Velocity", and therefore assuming the following:

- External longitudinal velocity is in a quasi-steady state
- So, lateral forces are calculated using tire slip angles and linear cornering stiffness

The governing equations for this model are:

- Dynamics:

$$\ddot{x} = 0$$
$$\ddot{y} = -\dot{x}r + \frac{F_{xF} + F_{yR} + F_{y,\ ext}}{m}$$
$$\dot{r} = \frac{aF_{yF} - bF_{yR} + M_{z,\ ext}}{I_{zz}}$$

$$r = \dot{\psi}$$

- External forces

$$F_{xFt} = 0$$

$$F_{yFt} = -C_{yF}\alpha_F\mu_F\frac{F_{zF}}{F_{z\ nom}}$$

$$F_{xRt} = 0$$

$$F_{yRt} = -C_{yR}\alpha_R\mu_R\frac{F_{zR}}{F_{z\ nom}}$$

To maintain pitch and roll equations, the block uses the following equations:

$$F_{zF} = \frac{bmg - (\ddot{x} - \dot{y}r)mh + hF_{x\ ext} + bF_{z\ ext} - M_{y\ ext}}{a+b}$$

$$F_{zR} = \frac{amg - (\ddot{x} - \dot{y}r)mh - hF_{x\ ext} + aF_{z\ ext} + M_{y\ ext}}{a+b}$$

Note: External forces includes both drag and external force inputs so:

$$F_{x,y,z\ ext} = F_{d\ x,y,z} + F_{x,y,z\ ext}$$

$$M_{x,y,z\ ext} = M_{d\ x,y,z} + M_{x,y,z\ ext}$$

- Tire forces

$$\alpha_F = \arctan(\frac{\dot{y} + ar}{\dot{x}}) - \delta_F$$

$$\alpha_R = \arctan(\frac{\dot{y} - br}{\dot{x}}) - \delta_R$$

$$F_{xF} = F_{xFt}\cos(\delta_F) - F_{yFt}\sin(\delta_F)$$

$$F_{yF} = -F_{xFt}\sin(\delta_F) + F_{yFt}\cos(\delta_F)$$

$$F_{xR} = F_{xRt}\cos(\delta_R) - F_{yRt}\sin(\delta_R)$$

$$F_{yR} = -F_{xRt}\sin(\delta_R) + F_{yRt}\cos(\delta_R)$$

Symbols	Variables
$x, \dot{x}, \ddot{x}$	Vehicle's center of mass displacement, velocity and acceleration along the vehicle- fixed x axis
$y, \dot{y}, \ddot{y}$	Vehicle's center of mass displacement, velocity and acceleration along the vehicle- fixed x axis
ψ	Yaw angle
$r, \dot{\psi}$	Yaw rate
F_{xF}, F_{xR}	Longitudinal forces applied to front and rear wheels, along the vehicle-fixed x-axis
F_{yF}, F_{yR}	Lateral forces applied to front and rear wheels, along vehicle-fixed y-axis
$F_{x\ ext}, F_{y\ ext}, F_{z\ ext}$	External forces applied to vehicle CG, along the vehicle-fixed x-, y-, and z-axes
$F_{d\ x}, F_{d\ y}, F_{d\ z}$	Drag forces applied to vehicle CG, along the vehicle-fixed x-, y-, and z-axes
$M_{x\ ext}, M_{y\ ext}, M_{z\ ext}$	External moment about vehicle CG, about the vehicle-fixed x-, y-, and z-axes
I_{zz}	Vehicle body moment of inertia about the vehicle-fixed z-axis
F_{xFt}, F_{xRt}	Longitudinal tire force applied to front and rear wheels, along the vehicle-fixed x-axis
F_{yFt}, F_{yRt}	Lateral tire force applied to front and rear wheels, along vehicle-fixed y-axis
F_{zF}, F_{zR}	Normal force applied to front and rear wheels, along vehicle-fixed z-axis
$F_{z\ nom}$	Nominal normal force applied to axles, along the vehicle-fixed z-axis
a, b	Distance of front and rear wheels, respectively, from the normal projection point of vehicle CG onto the common axle plane
m	Vehicle body mass
h	Height of vehicle CG above the axle plane
α_F, α_R	Front and rear wheel slip angles
C_{yF}, C_{yR}	Front and rear wheel cornering stiffness
μ_F, μ_R	Front and rear wheel friction coefficient
δ_F, δ_R	Front and rear wheel steering angles

Table 2.1: Variables and Symbols used

As for the parameters used for this model:

Vehicle Parameters	Numerical Value
Number of wheels on front axle $[-]$	2
Number of wheels on rear axle $[-]$	2
Vehicle mass [kg]	2000
a [m]	1.4
b [m]	1.4
Vertical distance from center of mass to axle plane [m]	0.35
Initial inertial frame longitudinal position [m]	$x_0 + disttocenter * cos(\psi_0)$

Table 2.2: Vehicle's Longitudinal Parameters

Vehicle Parameters	Numerical Value
Front tire corner stiffness $[\mathrm{N\,rad^{-1}}]$	12000
Rear tire corner stiffness $[\mathrm{N\,rad^{-1}}]$	11000
Initial inertial frame lateral displacement [m]	$y_0 + disttocenter * sin(\psi_0)))$
Initial lateral velocity $[\mathrm{m\,s^{-1}}]$	0

Table 2.3: Vehicle's Lateral Parameters

Vehicle Parameters	Numerical Value
Yaw polar inertia [kgm^2]	4000
Initial yaw angle [rad]	ψ_0
Initial yaw rate [rad s^{-1}]	0

Table 2.4: Vehicle's Yaw Parameters

Vehicle Parameters	Numerical Value
Longitudinal drag area [m^2]	0
Longitudinal drag coefficient [$-$]	0.3
Longitudinal lift coefficient [$-$]	0.1
Longitudinal drag pitch moment [$-$]	0.1
Relative wind angle vector [rad]	[0 : 0.01 : 0.3]
Side force coefficient vector [$-$]	[0 : 0.03 : 0.9]
Yaw moment coefficient vector [$-$]	[0 : 0.01 : 0.3]

Table 2.5: Vehicle's Aerodynamic Parameters

Vehicle Parameters	Numerical Value
Absolute pressure [Pa]	101325
Air temperature [K]	273
Gravitational acceleration [m s^{-2}]	9.81
Nominal friction scaling factor [-]	1

Table 2.6: Environment's Parameters

Vehicle Parameters	Numerical Value
Longitudinal velocity tolerance [m s^{-1}]	0.01
Nominal normal force [N]	5000
Geometric longitudinal offset from axle plane [m]	0
Geometric lateral offset from axle plane [m]	0
Geometric vertical offset from axle plane [m]	0

Table 2.7: Block's Simulation Parameters

Chapter 3

Non-linear Model Predictive Control

In this chapter, the Model Predictive Control (MPC) and subsequently the Non-linear MPC (NMPC) are presented. Initially, an overview of their principles and a showcase of state of art application of the NMPC, then their mathematical formulation is put in focus, lastly, the properties of the properties. This chapter is based on [7] & [8].

3.1 Overview

MPC, also known as receding horizon control, has developed considerably over the last two decades: first becoming popular during the 70s and then attracting huge interest from control theorists during the 90s in the area of NMPC.
MPC does not indicate a specific control strategy, but instead it is the explicit usage of a model of the process to obtain control signal by minimizing an objective function. Different typologies of MPC exists, such as Linear MPC, Adaptive MPC, Gain scheduled MPC, NMPC, etc. Furtheron, a brief introduction of each type mentioned:

Linear MPC: Although most real cases are characterized by non-linear systems, but by different means, like limiting the range of operation, they can be considered as linear. Linearizing a system is done for two main reasons: identification of a linear model based on data is more easy than trying to find the nonlinear model, and linear models yields acceptable results when the plant's state remain in the neighborhood of the operating point, e.g., in the process industries commonly the objective is to preserve a stationary state rather than perform passing from one operation point to another, [9].

Applying an MPC to a linear system or a linearized system simplify the control problem, resulting in a much faster and more robust control scheme.

Adaptive MPC: This type adapts the prediction model in case of changing operating conditions. It uses a fixed model but permits the changing of the parameters of this model according to the operating conditions. Ideally, before making a prediction, the model is updated to fit the current situation. Once done, the model stays the same over the prediction horizon.

Gain-scheduled MPC: Here, linearization of the model happens offline at the operating points of interest and for each operating point a linear MPC is designed, each controller is independent from the other and therefore each have his own set of state variables and constraints. In this approach, an algorithm is needed to switch between the predefined MPCs for different operation conditions.

NMPC: Another case of interest, non-linear systems. Adding to their innate non-linearity, constraints, whether linear or non-linear, on the system dynamics imply tough operating conditions. Hence complicating the operation altogether. Therefore, a non-linear model is used inside the MPC to be able to describe the process adequately and as precise as possible, making it a Non-linear MPC.

So what define a controller as an MPC is its structure, that is:
At each time step:

1. Obtain measurements (or estimates) of the current states of the system

2. Compute an optimal input signal by minimizing the cost function over the prediction horizon by means of the model of the system

3. Implement the first part of the optimal input sequence calculated (receding horizon strategy)

4. Redo again staring from 1.

Therefore MPCs are different in the model used to represent the process, the cost function to be minimized, the optimization strategy, and the optimization startegy. Hence, the computational complexity depends on the chosen model, objective, and constraints. Furthermore, it allows to incorporate constraints whether it is an input, state, and/or output and to manage the trade-off on performance-command effort. Also, it can easily deal with MIMO systems or with systems with very complex dynamics like interactions between inputs and outputs. Moreover, it innately compensates dead time (delay in the response to a control action), therefore it can be used to control processes with long delay times. Thus the resulting controller is an easy-to-implement control law. To add, another prominent features is its preview capability, meaning it can incorporate future reference information into the control problem to improve controller performance making it ideal for autonomous car application. However, MPCs come at a cost. From one side, when the system is characterized by changing dynamics, a huge amount of computations is needed at every sampling point to derive the control law that cannot be done beforehand and taking in account constraints complicates more the situation. Nevertheless, nowadays it is not more an issue. But from the other side, the greatest drawback is finding an appropriate model of the process, because although the design algorithm is based on prior knowledge of the model and it is independent of it, yet the quality of the controller will be directly affected by the inconsistencies between the the real process and the model used. Not mentioning the possibility of finding a local minima in the optimization process instead of the global minima.

Usually, in a control problem the controller formulate the input so the plant's states is as close as possible to a reference.

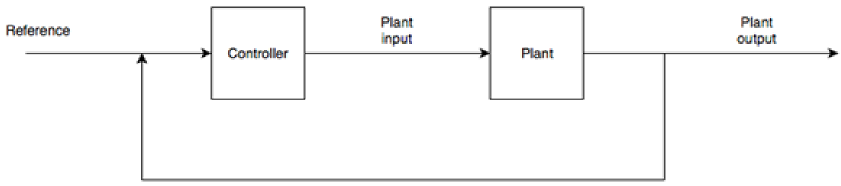

Figure 3.1: Example of a general CL control scheme

While for the MPC, as mentioned before, it uses a model of the process inside the controller in order to anticipate the effects of the possible commands.

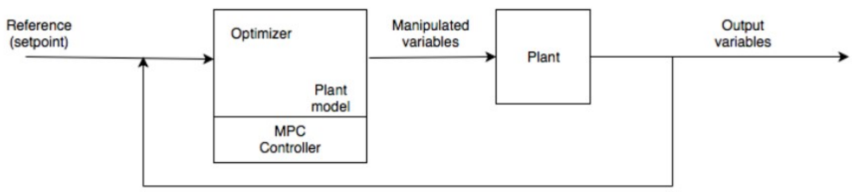

Figure 3.2: Example of a general MPC scheme

The model inside the MPC is not required to be exactly equal to the real world model, it can be simplified, for computational purposes, as long as it fits the criteria. Not to forget, a precise model is rarely easy to obtain. The main goal of the plant model is to be able to mimic the real plant to test control actions, and consequently, determine the optimal one. As a result, the chosen model must describe properly the dynamics of the process to insure proper prediction. Also, it must be simple enough to be implementable.
However the optimizer's purpose is to provide the control sequence that minimizes a cost function taking in account the different constraints on the system. If the cost function is quadratic, its minimum can be obtained as an explicit function (linear) of past inputs and outputs and the future reference trajectory. Instead, in the presence of inequality constraints, the cost function becomes more complex and therefore its minimization requires much higher computational costs. The control action provided by the optimizer is such that it drives the process to accomplish the specified requirements, fulfilling at the same time the specified constraints

Generally, the MPC problem is conceived as solving online a finite horizon open-loop optimal control problem subject to system dynamics and constraints on states and controls. The methodology of all controllers belonging to the MPC family is shown in Fig.3.3.

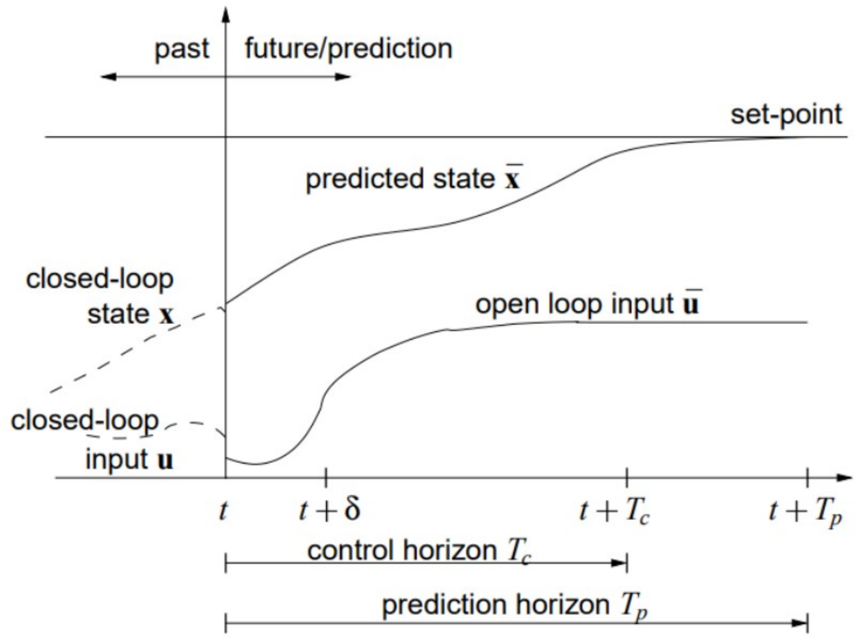

Figure 3.3: Strategy of an MPC

After receiving the measurements at time t, the controller must compute an optimal input signal as mentioned above. This is based on two key operations:

prediction and optimization:

1. The system state and output are predicted over the prediction horizon T_p so the time interval $[t; t + T_p]$
2. A command input is determined such that the predicted output has the desired behavior for $[t; t + T_p]$. The desired behavior is formalized by minimizing the objective function.

In an ideal situation, the predicted system behavior is identical to the real system behavior, but unfortunately this is not possible. Otherwise, the input function found at time t = 0 could be applied to the system for all times $t >= 0$. So a feedback mechanism is needed. This feedback mechanism is based on the idea of remeasuring the state of the system every T_s, also known as δ, i.e. sampling time. Using the new measurements at $t = t + T_s$, the procedure of prediction and optimization is repeated. Furthermore, recalculating the input every T_s allows the inclusion of constraints on the state of the systems and the inputs.
In order to lessen the computational load on the controller, a control horizon is imposed; the input signal can be seen as a vector with an infinite number of elements, making the optimization problem involving an infinite number of decision variables. To solve this problem, the input signal is parameterized.

3.1.1 State of Art

At its early stages, MPC was used for the process industries in chemical and oil domains for their low requirements in terms of processing power, but, recently, by taking a look at the current situation, the fields in which MPC is being applied in are expanding. The main domains are urban life, medical care, power grids, energy, development, aerospace, automotive, power electronics... In recent years, traditional monotonous predictive control algorithms can't meet increasingly complex industrial process requirements any longer. With the development of science and technology and the increased industrial demand, more effort is being put on the development and research on this control method, the number of papers published on this subject almost doubled. Although the theoretical research of predictive control is relatively complete and that the application of MPC is relatively successful, there's still a gap between the industrial development and application and the theory in terms of fulfilling the needs of further expansions and more developed applications. Some of these expansions can be in the domain of large scale systems, fast dynamic systems, and highly nonlinear systems.
The research efforts for the MPC improvements can be summarized in the following way.

- NMPC

  - Linearized nonlinear predictive control: linearizing the nonlinear model and calculating optimizations for the linearized model, while keeping the nonlinear model for the feedback par
  - Non-linear predictive control of hybrid systems: by adding logic variables, the segmented system with the logic switching characteristics is converted into a hybrid logic system, and then the optimal control is designed
  - Multi-model nonlinear predictive control: This type of system introduces multi-model methods into predictive control and uses several different linear models to approximate non-linearity. (A multi-model approach is defined as one in which more than one model-each derived from a different perspective and each is used with different distinct reasoning and strategies)

- Random predictive control model: According to the nature of the research object, it can be divided into two categories: random uncertainty model and random disturbance model

- Predictive Control Applied to More Fields
  - Predictive algorithm combined with other control algorithm (like PID)
  - Industrial Application Extensions: lately, industrial application focused on the integration of economic indicators (a statistic about an economic activity that give insight on economic performance and predictions of future performance)

At this stage, even though model predictive control has had its success and its development, but it is still facing many problems. For the algorithm aspect, this technique still exhibits a heavy toll on the processors therefore it is compatible with environments with high performance computers or slow dynamic processes, limiting its application domain. As from the perspective of the linearity of the process, the MPC is not used in case of highly nonlinear model, and that is largely due to the difficulties in modelling these types of processes precisely. Moreover, the lack of effective algorithms for nonlinear constraints optimization does not help its cause. Lastly, online prediction algorithms are guaranteed performance but at high cost making MPC not adhering to the needs of economic and social development.

3.2 Mathematical formulation

Lets consider a generic MIMO non-linear system:

$$\begin{aligned} \dot{x} &= f(x, y) \\ y &= h(x, u) \end{aligned} \tag{3.1}$$

where

$$x \in \mathbb{R}^n$$

is the state,

$$u \in \mathbb{R}_u^n$$

is the command input, and

$$y \in \mathbb{R}_y^n$$

is the output. Assuming the state is measured in real time with a sampling time T_s. The measurements are:

$$x_k(t_k),\ t_k = T_s k$$

with

$$k = 0, 1, 2...$$

As defined before, the strategy for the NMPC, and any MPC in that matter, consists of prediction and optimization. In the following subsections 3.2.1 and 3.2.2, a detailed showcase of these two operations.

3.2.1 Prediction

At time t, the controller predicts the state and output of the system over the interval $[t,\ t+T_p]$ by the integration of eqn. 3.1 or a model of it. At any time $\tau \in [t,\ t+T_p]$ the predicted output $\hat{y}(\tau)$ is a function of the initial state $x(t)$ and the input signal $\hat{u}(t:\tau)$ ($\hat{u}(t:\tau)$ denotes a generic input signal in the interval $[t,\ \tau]$):

$$\hat{y}(\tau) \equiv \hat{y}(\tau, x(t), \hat{u}(t:\tau))$$

In order to simplify calculations and as stated earlier, the input u is assumed as a constant signal after a time $T_c \in [T_s,\ T_p]$, previously defined as control horizon:

$$u(\tau) = u(t+T_c), \quad \tau \in [t+T_c,\ t+T_p]$$

Note that, as shown in Fig.3.3, u is an open-loop input, since its value in the interval $[t, t+T_p]$ does not depend on the value assumed instant by instant by the state x in that interval.

3.2.2 Optimization

The aim here is to generate, at each $t = t_k$, an input signal $u^*(t:\tau)$ such that the prediction

$$\hat{y}(\tau, x(t), \hat{u}^*(t:\tau)) \equiv \hat{y}(\hat{u}^*(t:\tau))$$

has the desired behavior for $\tau \in [t,\ t+T_p]$. In order to formulate the desired behavior, the objective function, J, is defined:

$$J(\hat{u}^*(t:t+T_p)) \doteq \int_t^{t+T_p} (\|\tilde{y}_p(\tau)\|_Q^2 + \|\tilde{u}(\tau)\|_R^2) d\tau + \|\tilde{y}_p(t+T_p)\|_P^2 \tag{3.2}$$

where

$$\|\tilde{y}_p(\tau)\| \doteq r(\tau) - \hat{y}(\tau)$$

is the predicted tracking error, and $r(\tau) \in \mathbb{R}_y^n$ is the reference. The symbol $\|.\|_X$ is the weighted vector norms and their integrals are square signal norms. Hence, the optimal input signal $\hat{u}^*(t:\tau)$ is the one minimizing the objective function $J(\hat{u}^*(t:t+T_p))$. In other words, minimizing the three terms composing J:

- $\|\tilde{y}_p(\tau)\|_Q^2$: the tracking error square norm, i.e., the difference instant by instant between the reference signal and the predicted output.
- $\|\tilde{u}(\tau)\|_R^2$: allows to manage the trade-off between performance and command activity
- $\|\tilde{y}_p(t+T_p)\|_P^2$: gives importance to the final tracking error

Note:

The square weighted norm of a vector $\nu \in \mathbb{R}^n$ is:

$$\|\nu\|_Q^2 \doteq \nu^T Q \nu = \sum_{i=1}^{n} q_i \nu_i^2,\ Q = diag\{q_1, ..., q_n\} \in \mathbb{R}^{n\times n}\ /\ q_i \geq 0.$$

The values assigned to the matrices Q, R and P are fundamental for the NMPC design, since the controller is optimized through them. Indeed, depending on the values assigned to the matrices, more or less importance can be given to each of the three terms in the objective function.

And therefore, the general formulation of the MPC become:

$$u^*(t:t+T_p) = arg \min_{u(.)} J(u(t:t+T_p)) \tag{3.3}$$

subject to the following constraints:

$$\begin{aligned} \dot{\hat{x}}(\tau) &= f(\hat{x}(\tau), u(\tau)),\ \hat{x}(t) = x(t) \\ \hat{y} &= h(\hat{x}(\tau), u(\tau)) \end{aligned} \tag{3.4}$$

with,

$$\begin{aligned} &\hat{x}(\tau) \in X_c,\ \hat{y}(\tau) \in Y_c,\ u(\tau) \in U_c \\ &u(\tau) = u(t+T_c),\ \tau \in [t+T_c,\ t+T_p] \end{aligned}$$

To solve this problem on-line, at each sampling time, an efficient numerical algorithm is needed, since its formulation is in general non-convex. Moreover, the objective function $J(u(\cdot))$ is a function of vector with an infinite number of elements, $u(\cdot)$; hence, the optimization involves an infinite number of decision variables.
To overcome this problem, a suitable parametrization of the input signal u is taken. In particular, the prediction interval $[t, t+T_p]$ is divided into sub-intervals $[t+\tau_i, t+\tau_{i+1}] \subset [t, t+T_p]$, $i \in \{1,2,\ldots,m\}$. Then, u is assumed constant on each sub-interval, so that the optimization problem reduces to a finite-dimension problem. The parameter m is to be set. In some cases, the choice m=1 gives satisfactory results with a reduced computational effort, but it can be not sufficient. With m=1, the command input is constant for the whole prediction interval T_p. Since, with the parametrization, the input signal is represented by the finite dimension matrix $C = [c_1, c_2, ..., c_m] \in \mathbb{R}^{n_u \times m}$, the optimization problem is reformulated as:

$$C^* = arg \min_{C \ in\mathbb{R}^{n_u \times m}} J(C) \tag{3.5}$$

subject to:

$$\begin{aligned} &\dot{\hat{x}}(\tau) = f(\hat{x}(\tau), u(\tau)),\ \hat{x}(t) = x(t) \\ &\hat{y} = h(\hat{x}(\tau), u(\tau)) \end{aligned} \tag{3.6}$$

with,

$$\begin{aligned} &\hat{x}(\tau) \in X_c,\ \hat{y}(\tau) \in Y_c,\ u(\tau) \in U_c \\ &u(\tau) = u(t+T_c),\ \tau \in [t+T_c,\ t+T_p] \end{aligned}$$

The resulting optimal solution to the minimization problem $u^*(t : t+T_p)$, computed at time t, is, as mentioned before, an open-loop input because it is not dependent on $x(\tau)$ (with $\tau > t$). On that account, applying the signal for the whole $[t, t+T_p]$ interval does not implement any feedback that is necessary for precision and could decrease errors and disturbance effects. This is where the Receding Horizon Strategy is implemented, in order to obtain a feedback control algorithm and a closed-loop behavior instead of an open-loop one. Simply put, this strategy consists of:

1. At time t_k, the optimal input is computed via optimization as described earlier.
2. Only the first input value $u^*(t_k)$ is implemented and it is kept constant for $t \in [t_k, t_{k+1}]$, so until the next sampling time.
3. Steps 1 and 2 are repeated at each t_k.

Applying this strategy means the optimal input is calculated instant by instant while keeping the input signal constant for the sampling time T_s, after that, a new optimal input is computed by optimization depending on the current state. This strategy ensures the passage to a closed-loop control scheme taking into consideration the progression of the plant underlining uncertainty errors and unpredicted disturbances, as shown in Fig.3.4.

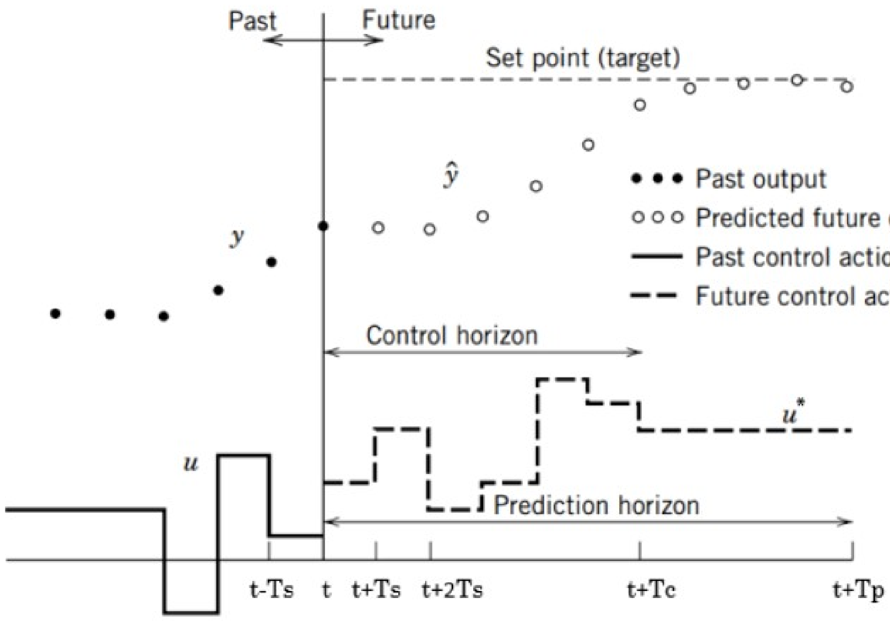

Figure 3.4: MPC with receding horizon strategy

3.2.3 Parameters choice

A design phase of an MPC consists of acting on the parameters by a trial-and-error process in simulation, aiming to obtain the best configuration in terms of results and computational load. These parameters are sampling time, prediction horizon, control horizon, and weight matrices.

Sampling time T_s: This parameter depicts at which rate the controller executes the control algorithm. Setting this parameter is not always an option as it comes given in many situations. In the cases where its value is up for the designer, it should be kept in mind to choose a value that is sufficiently small to deal with the plant dynamics (Nyquist-Shannon sampling theorem) and not too small to avoid numerical problems and slowing down the controller. Because, if it is too big, the controller won't be able to deal with a disturbance fast enough, and on the contrary, if it is too small, it causes an excessive computational load.

Prediction horizon T_p: As discussed earlier, the MPC predicts the future plant outputs and states and finds the optimal sequence of control inputs to have the desired behavior. So the prediction horizon determines how far the controller predicts. This parameters is not normally imposed, but must be a trade-off between sufficiently large to increase the closed loop stability properties and cover the significant dynamics of the system and not too large to avoid reducing the short time tracking accuracy and wasting computational power in planning. Ideally, the prediction horizon should be infinite but the solution wouldn't be fast.

Control horizon T_c: While for the control horizon it establishes the time where the input signal is a variable and after which it is considered constant. So it specify for how many sampling time the controller can set the input. Subsequently, a small control horizon involves fewer combination and so fewer computations for the optimization. And while a large value improves the performance, it also increase the complexity of the algorithm. It is worthy to note that small values of T_c don't effect significantly the performance.

Weight matrices Q, R, and P: The optimization process is regulated and controlled through the design of the three diagonal square matrices Q, R and P. Indeed, the tuning of each diagonal element of these matrices, allows to find a suitable trade-off between performances and command activity:

- Elements of Q affect directly the tracking error at each sampling time and they regulate the optimization of the system state
- Elements of R affect directly the command effort and they regulate the optimization of the input
- Elements of P are related only to the final term of the tracking error minimization and they regulate the optimization of the system output

The dimensions of the matrices are the system order. Each diagonal element sets a certain weight (or penalty) to the associated variable to be optimized. The higher the weight the more importance assumes the variable during the optimization process.
The initial choice of the elements of the matrices can be chosen according to the following criteria:

$$q_{ii} \begin{cases} 1 & \text{in the presence of requirements on } x_i \\ 0 & \text{otherwise} \end{cases}$$

$$r_{ii} \begin{cases} 1 & \text{in the presence of requirements on } x_i \\ 0 & \text{otherwise} \end{cases}$$

$$p_{ii} \begin{cases} 1 & \text{in the presence of requirements on } x_i \\ 0 & \text{otherwise} \end{cases}$$

After that, an adjustment is due depending on the system requirements and the results of simulations in the trial-and-error process of finding the proper values, knowing that an increase of q_{ii} and p_{ii} causes a decrease in the energy of $x_i,\ y_i$, reducing oscillations and convergence time, and an increase of r_{ii} causes a decrease in the energy of u_i, reducing command effort and energy consumption.

3.3 NMPC properties

3.3.1 Stability

One of the main problems of this control technique is that with a finite prediction and control horizons, the predicted open and the resulting closed-loop behaviour is in general different. Consequently, there is no guarantee that the closed-loop system will be stable. As described above, in order to improve the stability properties, a sufficiently large T_p must be chosen. Then, the most intuitive way to achieve stability is the use of an infinite prediction horizon. Indeed, in the nominal case feasibility at one sampling instances also implies feasibility and optimality at the next sampling instances. This follows from Bellman's Principle of Optimality [10]: the input and state trajectories computed as the solution of the NMPC optimization problem at a specific instance in time, are in fact equal to the closed-loop trajectories of the nonlinear system, i.e. the remaining parts of the trajectories after one sampling instance are the optimal solution at the next sampling instance. This fact also implies closed-loop stability.
However, the use of an infinite prediction horizon is impossible from a computational point of view. For this reason, it is necessary to enforce the closed-loop stability using a finite T_p. The simplest possibility is to add a so called zero terminal equality constraint at the end of prediction horizon, i.e. to add the equality constraint:

$$\hat{x}(t + T_p, x(t), \hat{u}) = 0$$

to optimization problem. This leads to stability of the closed-loop, if the optimal control problem possesses a solution at t = 0, since the feasibility at one time instance does also lead to feasibility at the following time instances and a decrease in the value function. One of the main problem of the zero terminal equality constraint is that the system must be brought to the origin in finite time. Additionally, from a computational point of view, an exact satisfaction of a zero terminal equality constraint does require an infinite number of iterations in the nonlinear programming problem. In order to overcome the problems due to the use of a zero terminal constraint, the so called terminal region constraint can be used:

$$\hat{x}(t + T_p) \in \Omega$$

where $\Omega \in \mathbb{R}^n$ is a bounded, closed and connected set. If the terminal region Ω is suitably chosen, then stability of the closed-loop system can be guaranteed[11].

3.3.2 Robustness

In real-world applications, the exact plant model is seldom known. This means that an approximated model $\hat{f}, \hat{h}$ is used for control design, instead of the real model f, h (this holds for any method). In general, this is not a problem since, thanks to the receding horizon strategy, standard NMPC is inherently robust; this means that it is characterized by a good robustness properties. If this property needs to be improved, different techniques can be implemented:

- min-max NMPC
- H_∞
- Parameterized NMPC controller

However these techniques are not widely used since they require a high computational effort and thus cannot be applied to problems where a small T_s is required.

Chapter 4

Case study

Here, firstly, a preliminary section is showcased talking about the setup of the simulation's environment. This includes the mathematical formulation and calculation needed to complete the simulation, along with the parameters considered for the car's body, the coordinates of a few points with particular importance, and the implementation of the constraints. Then, each tested type of parking is described. Those are parallel parking, perpendicular parking in reverse, and angle parking.

4.1 Simulation's environment

4.1.1 Parking spot general shape

For the purpose of generalising, a normal parking spot was considered, not extreme or special cases. A normal parking space can be described as an area enclosed or unenclosed, of sufficient size to park vehicles, with guarantying entrance and exit of the vehicle via a driveway connecting it to a public area, [12]. So, in this case it could be stated that a general parking spot is an empty rectangular spot delimited by a sidewalk from one side and enclosed between two other parking spots, considered occupied, and have one free side connected to a public area, a street lane fro example, which in turn is delimited from the opposite side by barrier or opposite-direction lane, considered here a barrier for the sake of simplicity. Therefore, a parking spot can be visualized as shown in Fig.4.1, 4.2 & 4.3. On top of that, the points $E\&I$ are defined as the corners of the parking spot.

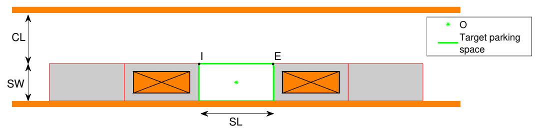

Figure 4.1: Parking spot example 1

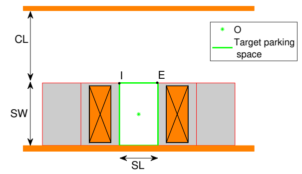

Figure 4.2: Parking spot example 2

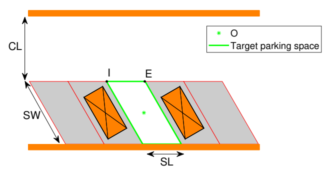

Figure 4.3: Parking spot example 3

For what follows, regarding the parking spot description, the following nomenclature has been adopted:

- Target parking space: the central parking space that is empty.
- SL: Parking slot length.
- SW: Parking slot width.
- CL: Distance between frontiers on the other side of the road.

4.1.2 Dimensions of the car model

Vehicles, both ego and parked, is modelled with a rectangular form with the following dimensions:

- Rear overhang, m: Distance of the vehicle which extend beyond the wheelbase at the rear, $m = 1$m.
- Front overhang, n: Distance of the vehicle which extend beyond the wheelbase at the front, $n = 0.9$m.
- Wheelbase, l: Distance between the two axles, $l = 2.8$m.
- Track, $2b$: Distance between hub flanges on the same axle, $2b = 1.8$m.

As shown in the Fig.4.4 , the points $A, B, C, \&D$ are considered as the ego vehicle corners, starting from A as the front left corner and continuing clockwise, naming them in alphabetical order.

4.1.3 Modelization of the parking spot

This was done by considering a Cartesian reference frame centered in the center of the rectangle, O, representing the target parking space and another relative Cartesian reference frame attached to the ego vehicle, centered in its center, G, with x in the direction of the length of the vehicle and y in the direction of the width of the vehicle. These references will be indexed 0 and 1, respectively.
Now, the coordinates of the points $A, B, C, D, E, \&I$ will be calculated using the transformation matrices, [13]. References are distant by $\vec{OG}$ and rotated by angle ψ. The distance vector between the two references is:

$$\vec{OG} = \begin{bmatrix} x + \frac{l+n-m}{2}\cos\psi \\ y + \frac{l+n-m}{2}\sin\psi \\ 0 \end{bmatrix}$$

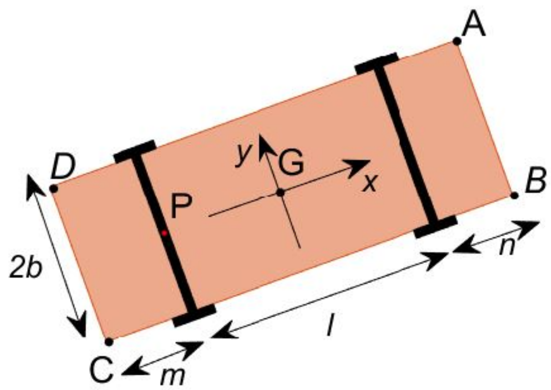

Figure 4.4: Car's body with dimensions

Therefore, the transformation matrix of reference frame 1 with respect to reference frame 0, ${}^{0}\hat{T}_1$:

$$
\begin{aligned}
{}^{0}\hat{T}_1 &= I \cdot \hat{Tras}(\vec{OG}) \cdot \hat{Rot}(z, \psi) \\
&= \begin{bmatrix} \cos\psi & -\sin\psi & 0 & x + \frac{l+n-m}{2}\cos\psi \\ \sin\psi & \cos\psi & 0 & y + \frac{l+n-m}{2}\sin\psi \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}
\end{aligned}
\tag{4.1}
$$

Inverting, to obtain the transformation matrix of reference frame 0 with respect to reference frame 1:

$$
{}^{1}\hat{T}_0 = \begin{bmatrix} \cos\psi & \sin\psi & 0 & -x\cos\psi - y\sin\psi - \frac{l+n-m}{2} \\ -\sin\psi & \cos\psi & 0 & x\sin\psi - y\cos\psi \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}
\tag{4.2}
$$

These two transformation matrices are then used to calculate the coordinates of points, that are later used to set the constraints inside the NMPC.

Knowing the coordinates of A with respect to reference 1:

$$^{1}\hat{A} = \begin{bmatrix} \frac{l+m+n}{2} \\ -b \\ 0 \\ 1 \end{bmatrix}$$

The coordinates of A in reference 0 by:

$$\begin{aligned} ^{0}\hat{A} &= {}^{0}\hat{T}_1 \cdot {}^{1}\hat{A} \\ &= \begin{bmatrix} x + (l+n)\cos\psi - b\sin\psi \\ y + (l+n)\sin\psi + b\cos\psi \\ 0 \\ 1 \end{bmatrix} \end{aligned}$$

In the same way the coordinates of $B, C, \&D$ are calculated:

$$^{0}\hat{B} = \begin{bmatrix} x + (l+n)\cos\psi + b\sin\psi \\ y + (l+n)\sin\psi - b\cos\psi \\ 0 \\ 1 \end{bmatrix}, \quad ^{0}\hat{C} = \begin{bmatrix} x - m\cos\psi + b\sin\psi \\ y - m\sin\psi - b\cos\psi \\ 0 \\ 1 \end{bmatrix}$$

$$^{0}\hat{D} = \begin{bmatrix} x - m\cos\psi - b\sin\psi \\ y - m\sin\psi + b\cos\psi \\ 0 \\ 1 \end{bmatrix}$$

As for the points $E\&I$, the process is the opposite of that before. The coordinates in reference 0 are easily determined and so:

$$^{0}\hat{E} = \begin{bmatrix} \frac{SL}{2} \\ \frac{SW}{2} \\ 0 \\ 1 \end{bmatrix}, {}^{0}\hat{I} = \begin{bmatrix} \frac{-SL}{2} \\ \frac{SW}{2} \\ 0 \\ 1 \end{bmatrix}$$

Hence,

$$\begin{aligned} ^{1}\hat{E} &= {}^{1}\hat{A}_0 \cdot {}^{0}\hat{E} \\ &= \begin{bmatrix} \frac{SL}{2}\cos\psi + \frac{SW}{2}\sin\psi - x\cos\psi - y\sin\psi - \frac{l+n-m}{2} \\ -\frac{SL}{2}\sin\psi + \frac{SW}{2}\cos\psi + x\sin\psi - y\cos\psi \\ 0 \\ 1 \end{bmatrix} \end{aligned}$$

And,

$$
\begin{aligned}
{}^{1}\hat{I} &= {}^{1}\hat{A}_0 \cdot {}^{0}\hat{I} \\
&= \begin{bmatrix} -\frac{SL}{2}\cos\psi + \frac{SW}{2}\sin\psi - x\cos\psi - y\sin\psi - \frac{l+n-m}{2} \\ \frac{SL}{2}\sin\psi + \frac{SW}{2}\cos\psi + x\sin\psi - y\cos\psi \\ 0 \\ 1 \end{bmatrix}
\end{aligned}
$$

Finally, the frontiers of the parking spot remains to be formulized. This is done using a heaviside function, when the parking borders are vertical and horizontal, as in the perpendicular and perpendicular parking, and using a sigmoid function and an inclined line when the parking borders are inclined, as in angle parking. The heaviside function is a step function, [14], described in the following way:

$$
heaviside(x) = \begin{cases} 0 & \text{for } x < 0 \\ 1 & \text{for } x \geq 1 \end{cases}
$$

In order to modelize the frontiers of the parking spot, the function heaviside was used in the following way:

$$
h(x) = (heaviside(x - \frac{SL}{2}) - heaviside(x + \frac{SL}{2})) \times SW + \frac{SW}{2} \tag{4.3}
$$

An example of the function h($\cdot$) is shown in Fig.4.5, the parameters were set as: $SL = 6.2$ & $SW = 3.1$.

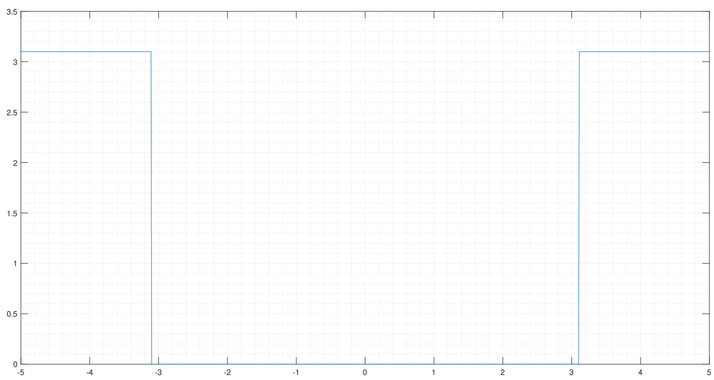

Figure 4.5: Example of the heaviside function h($\cdot$)

While the sigmoid function is a mathematical function that uses the exponential function and have a characteristic "S"-shaped curve or sigmoid curve, it is displayed

in Fig.4.6.
It is written in the following way:

$$S(x) = \frac{1}{1+e^{-x}} \tag{4.4}$$

Applying some transformations on the the sigmoid function it can be used to

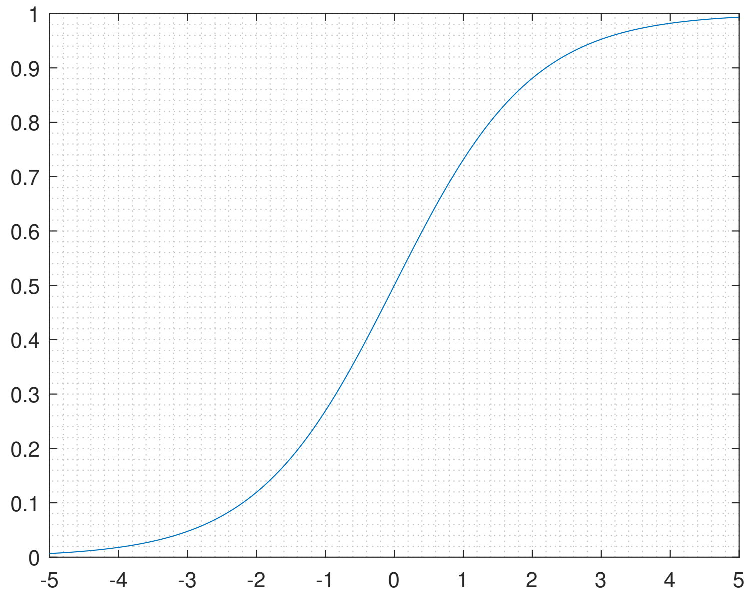

Figure 4.6: Example of the sigmoid function S(·)

formulize the left border of the parking spot:

$$sig(x) = SW \times \frac{1}{1+e^{a(x+\frac{SL}{2})}} - \frac{SW}{2} \tag{4.5}$$

Moreover, the second inclined frontiers of the parking spot can be modelized using a simple line equation

$$y(x) = a \times x + a \times \frac{SL}{2}; \tag{4.6}$$

where a depicts the slope of the parking's side frontiers and $SL\&SW$ are the dimensions of the parking spot as seen in Fig.4.3.

4.1.4 Constraints

One of the key features of the NMPC is the possibility to impose constraints on the control process. Example of a constraint F definition in the NMPC is in the form of $F \leq 0$. Therefore, if the value A must be lower than a certain threshold a, the constraint F will be written in the following way:

$$A \leq a \Rightarrow F = A - a \leq 0$$

Constraints for two maneuvers: Parallel and Perpendicular parking

And henceforth, for this case, the conditions, that the vehicle should respect, are:

1. The car remains under the upper barrier.
2. The car respects the frontiers of the target parking space, i.e., the heaviside function defined in eq.4.3.

Using the convex properties, this can be translated to:

1. The necessary and sufficient constraint for the first constraint is to keep the ordinate of the 4 corners of the car under the upper barrier.
2. As for the second condition, two constraints are needed:
   - The ordinate of the 4 corners of the car remain above the heaviside function $h(\cdot)$ described in the previous paragraph.
   - The two points E & I remain outside the ego vehicle.

As for the ultimate constraint, according to [15], it can be formulated as:

$$\begin{aligned} |{}^1E_x| &\geq \frac{l+m+n}{2}, \text{ when } |{}^1E_y| \leq b \\ |{}^1I_x| &\geq \frac{l+m+n}{2}, \text{ when } |{}^1I_y| \leq b \end{aligned} \tag{4.7}$$

Mathematically speaking, it would be written as:

$$FA1 = {}^0A_y - (CL + \frac{SW}{2}) \tag{4.8}$$

$$FB1 = {}^0B_y - (CL + \frac{SW}{2}) \tag{4.9}$$

$$FC1 = {}^0C_y - (CL + \frac{SW}{2}) \tag{4.10}$$

$$FD1 = {}^0D_y - (CL + \frac{SW}{2}) \tag{4.11}$$

$$FA2 = h({}^0A_x) - {}^0A_y \tag{4.12}$$

$$FB2 = h(^0B_x) - {}^0B_y \tag{4.13}$$

$$FC2 = h(^0C_x) - {}^0C_y \tag{4.14}$$

$$FD2 = h(^0D_x) - {}^0D_y \tag{4.15}$$

Using the conditional statements if to impose the constraints relating to E & I:

$$FE2 = \frac{l+m+n}{2} - |\,^1E_x| \tag{4.16}$$

$$FI2 = \frac{l+m+n}{2} - |\,^1I_x| \tag{4.17}$$

Constraints of the third maneuver: Angle parking

And subsequently, the conditions this car must respect are:

1. The car remains under the upper barrier.
2. The car respect the frontiers of the target parking space, i.e., the sigmoid function defined in eq.4.5 and the inclined line defined in eq.4.6.

This can be translated to:

1. The necessary and sufficient constraint for the first constraint is to keep the ordinate of the 4 corners of the car under the upper barrier.
2. As for the second condition, two constraints are needed:
   - The ordinate of the 4 corners of the car remain above the sigmoid function $sig(\cdot)$, and below the inclined line $y(\cdot)$ described in the previous paragraph.
   - The point E remain outside the ego vehicle.

In this case, constraints on the two corner points (E&I) was not needed to be set because the sigmoid function (eq.4.5) englobe the point I, and the inclined line function (eq.4.6) passes through the point E.
Mathematically speaking, it would be written as:

$$FA1 = {}^0A_y - (CL + \frac{SW}{2}) \tag{4.18}$$

$$FB1 = {}^0B_y - (CL + \frac{SW}{2}) \tag{4.19}$$

$$FC1 = {}^0C_y - (CL + \frac{SW}{2}) \tag{4.20}$$

$$FD1 = {}^0D_y - (CL + \frac{SW}{2}) \tag{4.21}$$

$$FA2 = {}^0A_y - y({}^0A_x) \tag{4.22}$$

$$FB2 = {}^0B_y - y({}^0B_x) \tag{4.23}$$

$$FC2 = {}^0C_y - y({}^0C_x) \tag{4.24}$$

$$FD2 = {}^0D_y - y({}^0D_x) \tag{4.25}$$

$$FA3 = sig({}^0A_x) - {}^0A_y \tag{4.26}$$

$$FB3 = sig({}^0B_x) - {}^0B_y \tag{4.27}$$

$$FC3 = sig({}^0C_x) - {}^0C_y \tag{4.28}$$

$$FD3 = sig({}^0D_x) - {}^0D_y \tag{4.29}$$

In addition, some constraints were imposed on the car via the controller, e.g., max steering angle of $\pm\frac{\pi}{4}$rad, because of physical limitations (wheel hub), and a maximum speed of $\pm 2\,\mathrm{m\,s^{-1}}$, since in a parking maneuver the speed does not exceed this limit.

4.1.5 Simulation's Parameters

As what concerns the parametrization of $u(\cdot)$, the piece-wise approach is adopted here with $m = 2$. As it was found that $m = 1$ does not guarantee optimal results in all cases. Therefore, $u(\tau)$ $s.t.$ $\tau \in [t, t + T_p]$ take two constant values:

$$u(\tau) = \begin{cases} u_1 & \text{for } \tau \in [t, t + \alpha T_p] \\ u_2 & \text{for } \tau \in [t + \alpha T_p, t + T_p] \end{cases}$$

The parameters discussed earlier where set as shown in Tab.4.1. The states of

Parameters	Numerical Value
$T_{simulation}$ [s]	25
T_s [s]	0.05
T_p [s]	10
T_c [s]	T_p
α	0.6

Table 4.1: Adopted Parameters

the ego vehicle (x, y, ψ) are located at the center of rear axle, where x & y are the abscissa and ordinate of the point P (in [m]), respectively, and ψ is the direction of the vehicle along its x axis, also known as yaw angle, in [rad], (see Fig 4.4).

After doing the work described in section 4.1, the code is ready for the different case studies.

4.2 Case Study 1: Parallel parking

The first and most important maneuver is the parallel parking for what it brings of challenge and its high level of difficulty. It was thought that succeeding in this maneuver will enable the control to surpass any other maneuvers. The maneuver is divided in two parts. In the first part, the goal of the controller is to take the car after the parking space staying on the street, as shown in Fig.4.8, while in the second part the goal is to enter the parking spot. Dividing the process into this two parts allows to back up the car inside the parking space, which is proven to be a better approach to parallel parking.

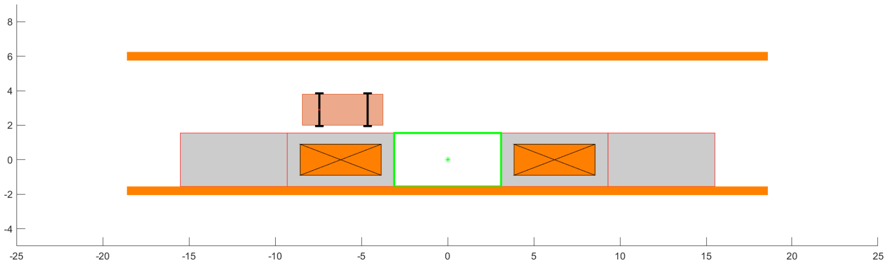

Figure 4.7: Start point of part 1

For this maneuver, the following parameters were considered: The starting point of part 1 is:

$$\begin{bmatrix} -1.2 \times SL \\ SW - \epsilon_1 \\ 0 \end{bmatrix}$$

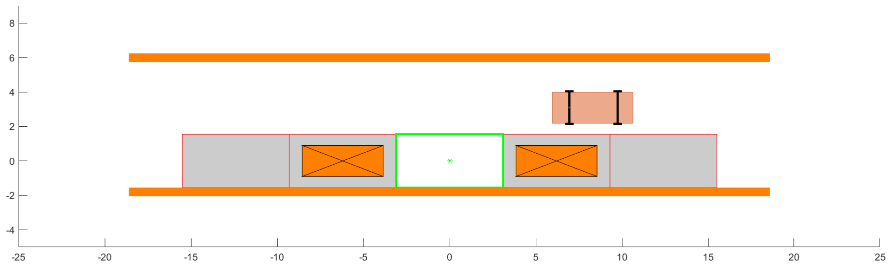

Figure 4.8: Target point of part 1 / Start point of part 2

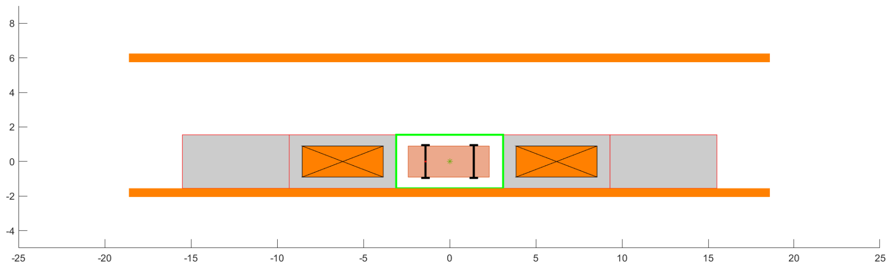

Figure 4.9: Target point of part 2

ϵ_1 is a small parameter added to not have the target coordinate equal to the initial coordinate. Its value is equal to $0.2m$.
And the target point of part 1 , also known as starting point of part 2:

$$\begin{bmatrix} 1.12 \times SL \\ SW \\ 0 \end{bmatrix}$$

Parameters	Numerical Value in [m]
SL	6.2
SW	3.1
CL	4.2

Table 4.2: Parameters of Maneuver 1

and the target point of part 2:

$$\begin{bmatrix} -\frac{l}{2} \\ 0 \\ 0 \end{bmatrix}$$

4.3 Case Study 2: Reverse perpendicular parking

The second maneuver is a very common one. It is found in vast open parking spaces, in places like commercial center. In order to be on the same level, this maneuver also is divided into two parts, so to be on a par with the previous one. The first is equivalent to that of the parallel parking, while the second part is to enter the parking spot reversing. Parking in reverse allows a simple exit of the vehicle when it is time to take the road again. For this maneuver, the following parameters were considered:

Parameters	Numerical Value in [m]
SL	3.2
SW	5.2
CL	6

Table 4.3: Parameters of maneuver 2

The starting point of part 1 is:

$$\begin{bmatrix} -2 \times SL \\ \frac{SW}{2} + 1.5 \\ 0 \end{bmatrix}$$

And the target point of part 1 , also known as starting point of part 2:

$$\begin{bmatrix} SL \\ \frac{SW}{2} + 2 \\ 0 \end{bmatrix}$$

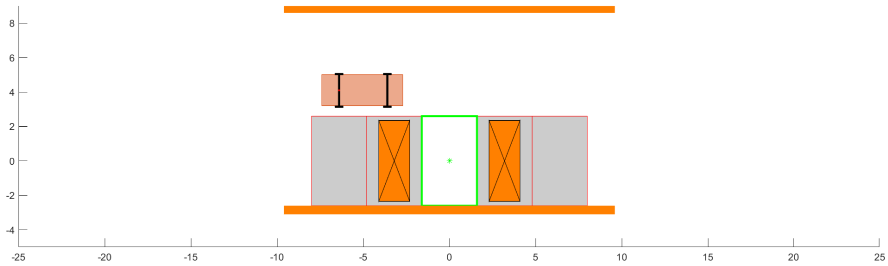

Figure 4.10: Start point of part 1

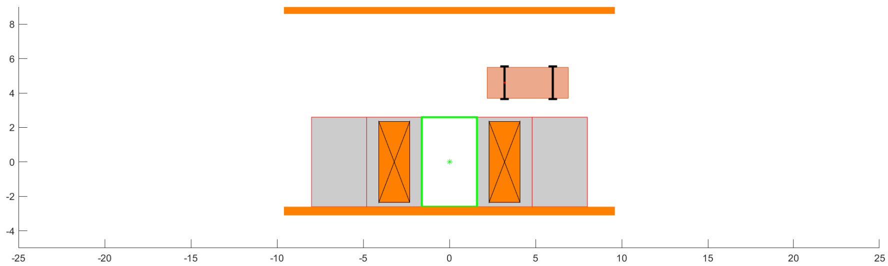

Figure 4.11: Target point of part 1 / Start point of part 2

and the target point of part 2:

$$\begin{bmatrix} 0 \\ -\frac{SW}{2} + m + \epsilon_2 \\ \frac{\pi}{2} \end{bmatrix}$$

Here, ϵ_2's role is to add a safety distance. Its value is set to $0.1m$

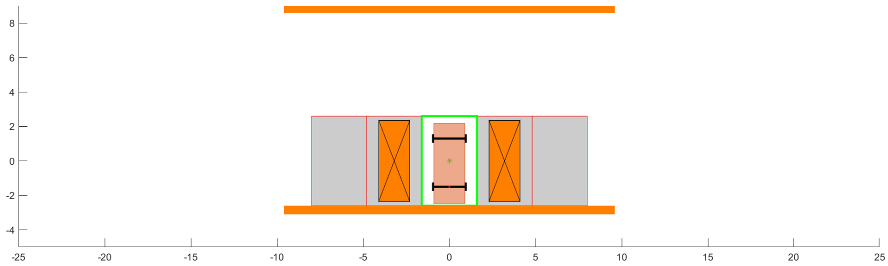

Figure 4.12: Target point of part 2

4.4 Case Study 3: Angle parking

This third maneuver is a also very common one. It is used to save space and because of their simplicity to park not requiring vehicles to make sharp turns, which in turns lower the chance of a collision during entrance and exit of the vehicle. Because of its simplicity and the fact that the parking spots are usually layed out in order to facilitate entrance and exit of the vehicle, this maneuver will not be divided in two parts as the previous ones, and will be instead done in one complete movement. In angle parking, parking spaces are inclined a set angle, θ that can differ in each case. In our case, θ is set equal to 60°.

For this maneuver, the following parameters were considered:

Parameters	Numerical Value in [m]
SL	3.5
SW	6
CL	4

Table 4.4: Parameters of Maneuver 3

The starting point is:

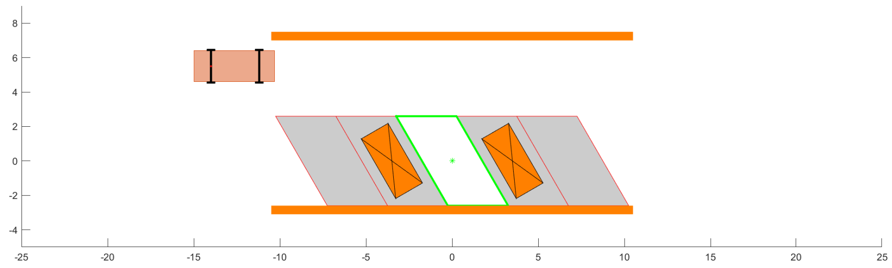

Figure 4.13: Start point

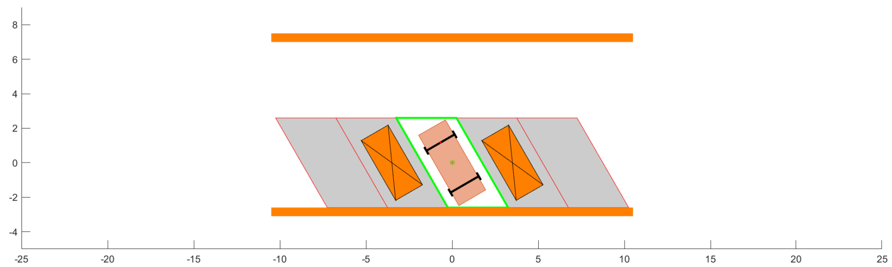

Figure 4.14: Target point

$$\begin{bmatrix} -4 \times SL \\ \frac{SW}{2} + 2.5 \\ 0 \end{bmatrix}$$

And the target point is:

$$\begin{bmatrix} -\frac{l+n-m}{2} \times \cos(\theta) \\ -\frac{l+n-m}{2} \times \sin(\theta) \\ -\theta \end{bmatrix}$$

The target point is determined considering the center of mass of the vehicle coinciding with the origin of reference O and aligned with the parking spot.

Chapter 5

Implementation and Results

This section first presents all the steps that were taken for the tuning of the NMPC. Then, starting from this "suitable" configuration, the results obtained considering the scenarios described in the previous section are shown. The tuning was based on the most challenging part of all the maneuvers discussed, that is the second part of the parallel parking maneuver. More than 50 trials were implemented in order to obtain the proper parameters, taking into account the cost function J, the overall time, and the stability and smoothness of the process. As for the results, they are reported by showing the final position of the vehicle along with the trajectory of the center of the rear axle and time evolution of the vehicle's body, the cost function evolution, the command inputs, and the variation of the error, i.e., the difference between the target coordinates and final coordinates.
Along the results, the Monte Carlo campaign conducted is showcased. The Monte Carlo campaign was only implemented on the first maneuver due to it being considered the most challenging and the most difficult one between all three and therefore the most significant.
Note, the third error $\Delta\psi$ is normalized by dividing with π.

5.1 Tuning

As mentioned earlier, the tuning process was done considering only the second part of the first case study: parallel parking. The studied maneuver concerns backing the car from after the parking spot inside it. The table 5.1 shows the first few matrices considered. Indeed, the first choice of the $Q, P, \&R$ is rather trivial and direct, their development is done taking a close look at the resulting simulation.

Trial	Q	R	P	Comment
1	100*diag([1 1 1])	0.05*eye(2)	1000*diag([1 5 5])	Success
2	100*diag([1 1 1])	0.05*eye(2)	500*diag([1 5 5])	Failed
3	100*diag([1 1 5])	0.05*eye(2)	500*diag([1 5 5])	Failed
4	100*diag([1 1 5])	0.05*eye(2)	750*diag([1 5 5])	$\psi_{final} >>$
5	100*diag([1 1 10])	0.05*eye(2)	750*diag([1 5 5])	Failed
6	100*diag([1 1 10])	0.05*eye(2)	500*diag([5 10 1])	Success
7	100*diag([1 1 10])	0.05*eye(2)	500*diag([5 10 5])	Success
8	100*diag([1 1 10])	0.05*eye(2)	500*diag([5 10 1])	Success(mediocre)
9	100*diag([1 1 1])	0.05*eye(2)	500*diag([5 10 1])	Failed
10	100*diag([1 1 1])	0.05*eye(2)	500*diag([5 10 5])	$\psi_{final} >>$
11	100*diag([1 1 1])	0.05*eye(2)	2500*diag([5 10 5])	Failed

Table 5.1: Tuning process part 1

It was very clear that trials 4 and 8 were the best results with the lowest computational effort, so it only makes sense to try to improve them. Starting by trial 4:
It is noticeable that the inclination is the imperfection, so p_{33} is increased as a first approach:

Trial	Q	R	P	Comment
12	100*diag([1 1 5])	0.05*eye(2)	750*diag([1 5 7])	No improvements
13	100*diag([1 1 5])	0.05*eye(2)	750*diag([1 5 10])	Same results as trial 4

Table 5.2: Tuning process part 2

Increasing p_{33} proved to be useless. So it was decided to put effort on improving q_{33} instead (see table 5.3).
While trial 14 ended with the car still inclined relatively high enough, trial 15 was of better results where y was the farthest from the target value at $|y_{final} - y_{target}| = 0.27$, making it the configuration with the best parameters regarding the the improvements of trial 4.

Trial	Q	R	P	Comment
14	100*diag([1 1 7])	0.05*eye(2)	750*diag([1 5 5])	Success
15	100*diag([1 1 7])	0.05*eye(2)	750*diag([3 5 5])	Success
16	100*diag([1 1 5])	0.05*eye(2)	750*diag([3 5 5])	Failed
17	100*diag([1 1 7])	0.05*eye(2)	750*diag([2 5 5])	Success
18	100*diag([1 1 7])	0.05*eye(2)	750*diag([2 5 7])	Success
19	100*diag([1 1 7])	0.05*eye(2)	750*diag([2 5 10])	Success
20	100*diag([1 1 7])	0.05*eye(2)	750*diag([2 5 15])	Failed
21	100*diag([1 1 7])	0.05*eye(2)	750*diag([2 5 12])	Success
22	100*diag([2 2 7])	0.05*eye(2)	750*diag([2 5 12])	Failed

Table 5.3: Tuning process part 3

As for trial 8:

Trial	Q	R	P	Comment
23	100*diag([2 1 10])	0.05*eye(2)	500*diag([5 10 1])	Success(mediocre)
24	100*diag([2 1 10])	0.05*eye(2)	500*diag([7 10 1])	Success(mediocre)
25	100*diag([3 5 10])	0.05*eye(2)	500*diag([7 10 1])	Success
26	100*diag([5 5 10])	0.05*eye(2)	500*diag([7 10 1])	Failed
27	100*diag([5 5 10])	0.05*eye(2)	500*diag([5 10 1])	Success(mediocre)
28	100*diag([5 7 10])	0.05*eye(2)	500*diag([5 10 1])	Success(mediocre)

Table 5.4: Tuning process part 4

While only trial 26 failed, the majority of the rest delivered mediocre results. The best results so far were reached in trial 25. Although trying to improve them led to worse results or in the case of trial 26 to a failure.

Trial	Q	R	P	Comment
29	100*diag([5 7 10])	0.5*eye(2)	500*diag([5 10 1])	No improvements
30	100*diag([5 7 10])	0.05*eye(2)	750*diag([5 10 1])	No improvements
31	300*diag([5 7 10])	0.05*eye(2)	750*diag([5 10 1])	Failed

Table 5.5: Tuning process part 5

In Tab.5.5, the focus was on the magnitude of the matrices. It was proven to be insignificant. So, at the end of this tuning process, the best configuration is the one obtained in trial 25. For clarity, it is reported in Tab.5.6. It has been proven that this configuration is capable to reach the target while having to perform complicated maneuvers mitigating all the different constraints, described in the section4, especially in the case of parallel parking. Furthermore, it proves again its effectiveness and potential with the Monte Carlo.

	Q	R	P
Optimal NMPC configuration	100*diag([3 5 10])	0.05*eye(2)	500*diag([7 10 1])

Table 5.6: Optimal NMPC configuration

5.2 Case study 1: Parallel Parking

First, the results of the parallel parking are showcased, followed with a description of the Monte Carlo campaign conducted, ending with the presentation of its results.

5.2.1 Results

The first maneuver consists of delivering the vehicle to post parking space. It is shown in Fig.5.1 for the trajectory, Fig.5.2 for the cost function and the error signal in Fig.5.3. The red line depicts the trajectory of the center of the rear axle, the green line frame the target parking spot, and the magenta rectangles show the time evolution of the vehicle during its trajectory.

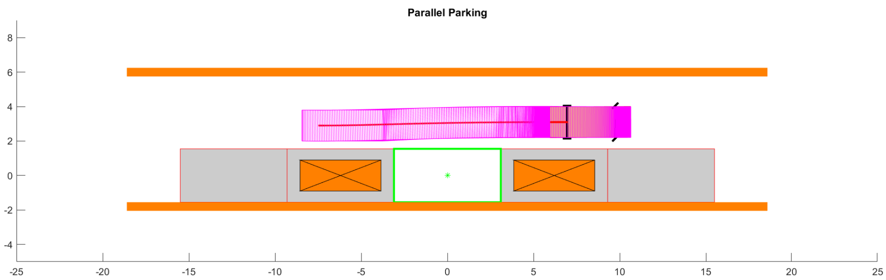

Figure 5.1: Trajectory of part 1

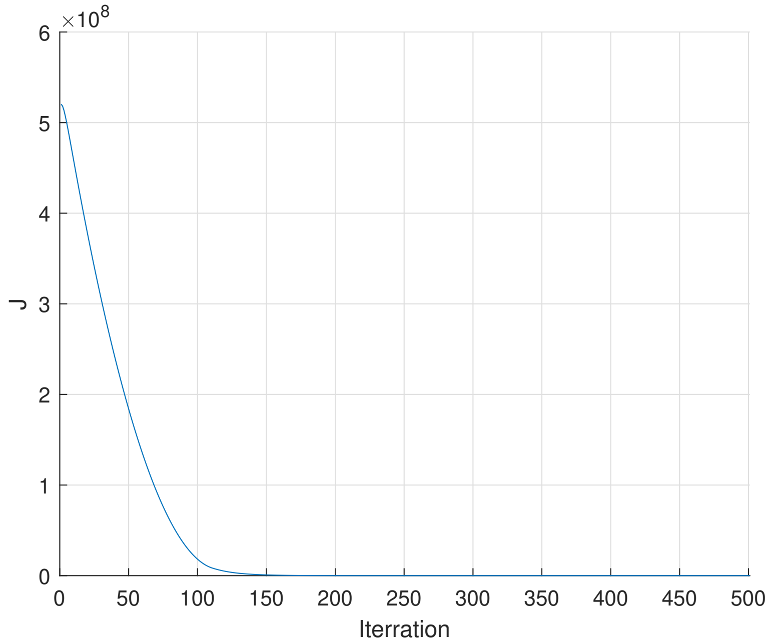

Figure 5.2: Cost function of part 1

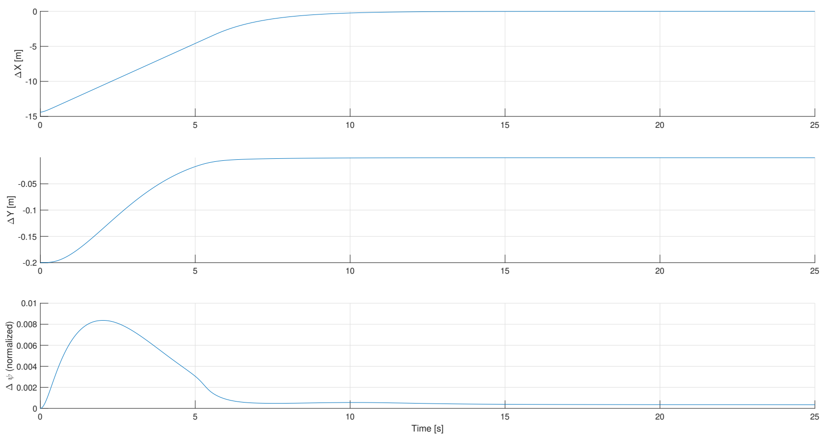

Figure 5.3: Error signal of part 1

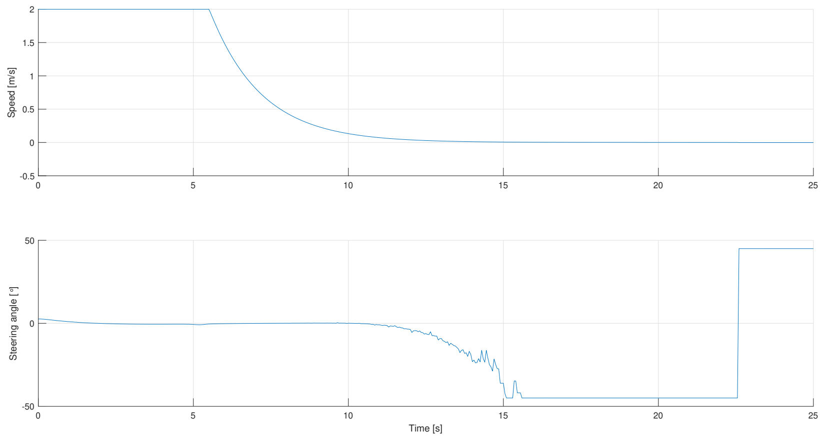

Figure 5.4: Commands signal of part 1

While for the second part:

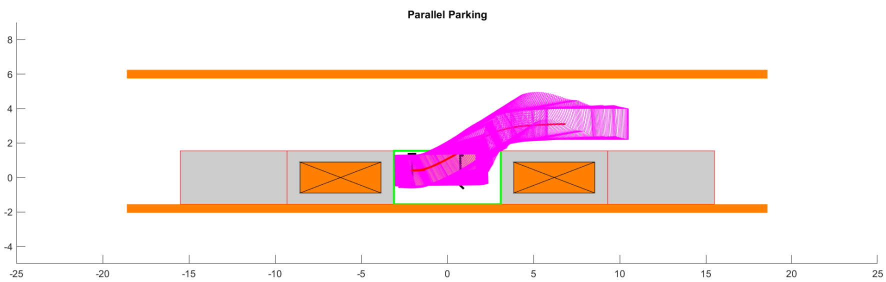

Figure 5.5: Trajectory of part 2

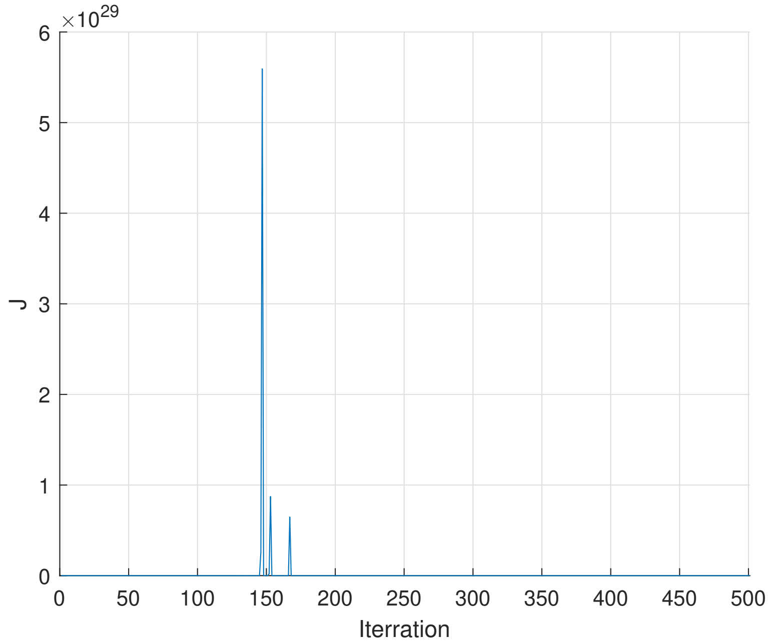

Figure 5.6: Cost function of part 2

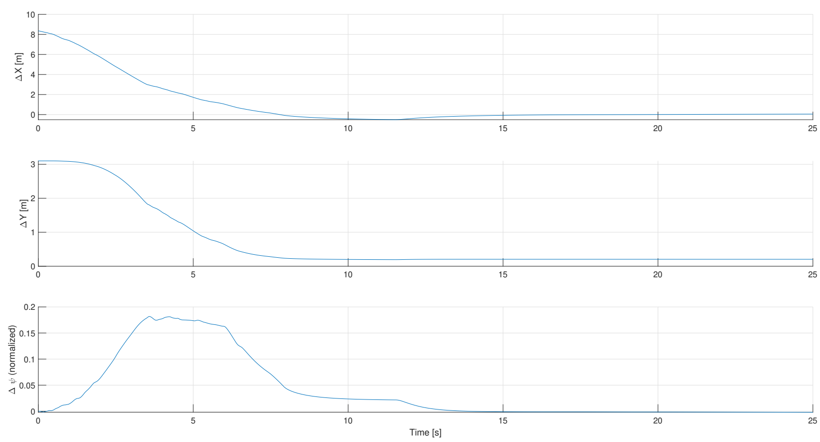

Figure 5.7: Error signal of part 2

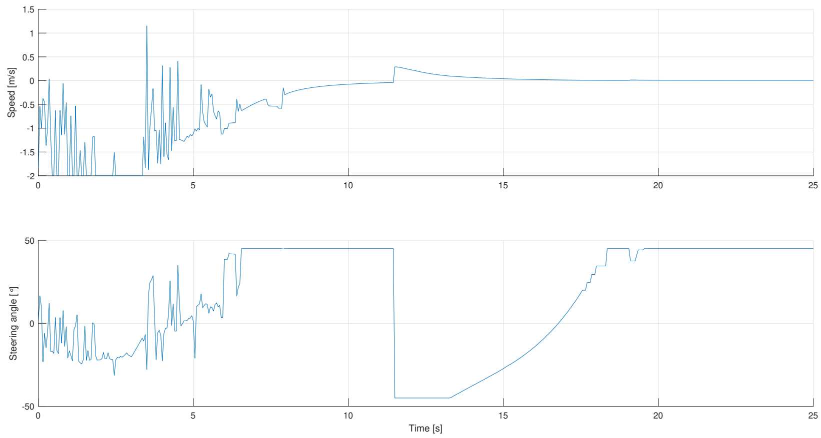

Figure 5.8: Commands signal of part 2

While the first part has a more declining exponential cost function, the second part instead shows a much elevated order of magnitude with high peaks cost function. That is because of the nature of both parts where the latter demands to get farther and then closer to the target while the other have a more direct approach. To add, these peaks represent the instant the vehicle approaches the border of the adjacent parking spots.
The final errors of both these simulations:

Tracing error	Part 1	Part 2
Δx [m]	-0.0008	-0.01
Δy [m]	-0.0004	0.08
$\Delta \psi$ (normalized)	0.0003	0.0098

Table 5.7: Tracking error of maneuver 1

5.2.2 Monte Carlo

Monte Carlo comes to verify the robustness of the controller in all cases. The proposed approach was to create a vector of random numbers of dimensions 100×3, the number are limited in a certain interval, and add each row to the initial position vector and execute a simulation. At the end, of each simulation, the success or failure of the simulation is signaled by changing the value of the corresponding row to 0 or 1, respectively, in a "error vector". For the first part, the variation of $x_{initial}$ and $y_{initial}$ was $[-0.5; 0.5]$ and for ψ is $[-0.2; 0.2]$.

As for the second part, using the Monte Carlo of the first, it was possible to deduce that an interval of ± 0.5 is only needed for $x_{initial}$, while for $y_{initial}$ an interval of 0.2 was enough, and for ψ an interval of ± 0.05.

It was found that the first part had a success rate of 100% and the controller was able to always get the car to the target point with an interval of ± 0.5 for x, ± 0.2 for y, and ± 0.05 for ψ. While the second part was found to have a success rate of $92\% - 93\%$. This success rate means succeeding in reaching the vicinity of the target point with out touching trespassing adjacent parking spot and having the car completely inside the target parking space.

5.3 Case study 2: Perpendicular Parking

The results of part 1:

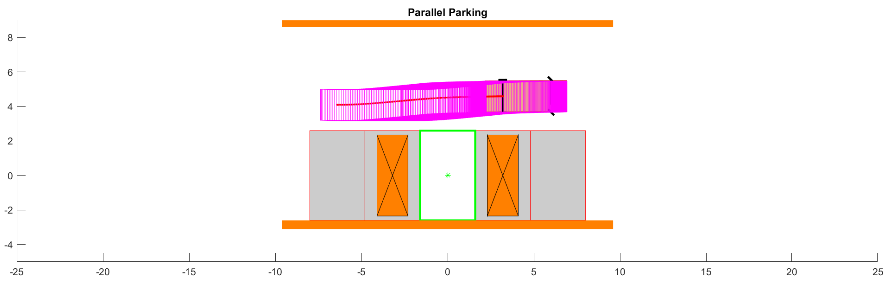

Figure 5.9: Trajectory of part 1

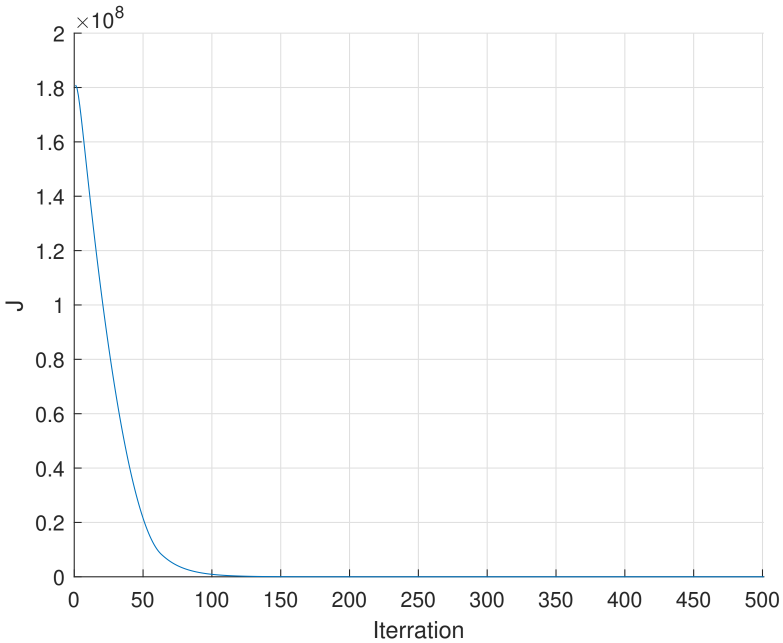

Figure 5.10: Cost function of part 1

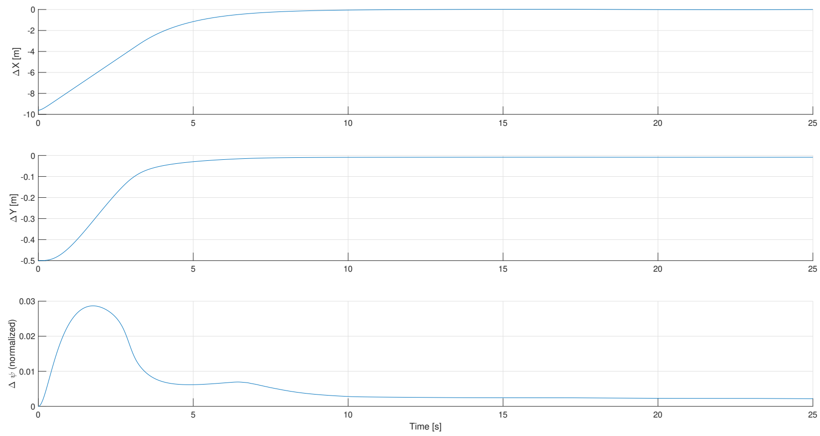

Figure 5.11: Error signal of part 1

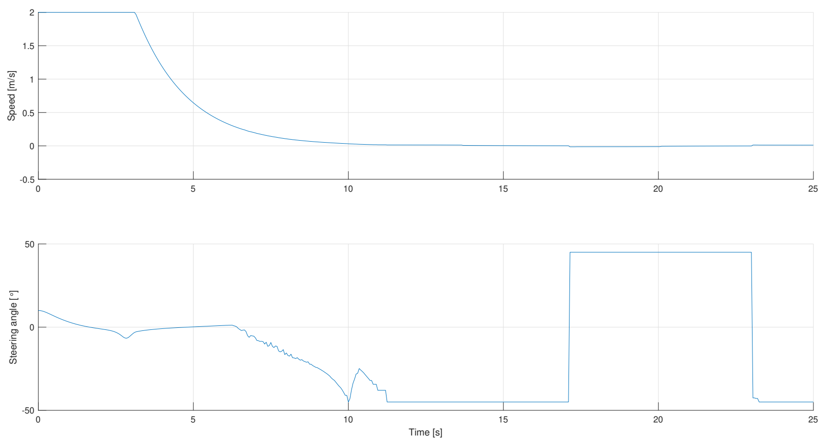

Figure 5.12: Commands signal of part 1

Considering it was a simple process, it was expected to be direct and simple, as it can be seen by the results.

While for the second part:

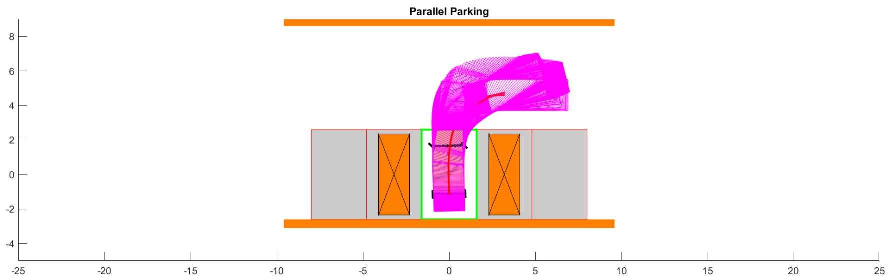

Figure 5.13: Trajectory of part 2

It is worthy to note that even though how trivial and limited the controller is but in this maneuver the controller puts the car in a convenient position allowing him to enter. This position is above the E point at approximately 45°.

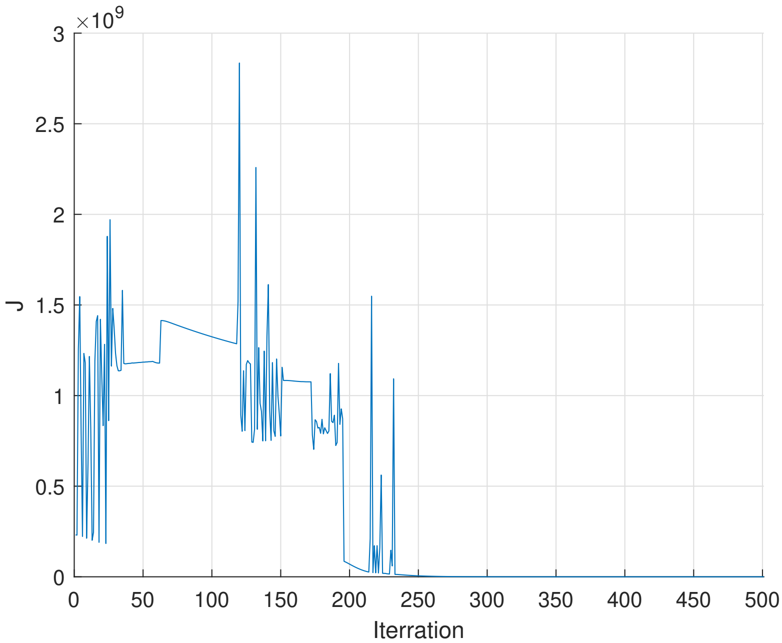

Figure 5.14: Cost function of part 2

The plot of the cost function shows high variation and different peaks. In addition to the analysis done before on J, that is still valid here, this also comes back to the trajectory of the vehicle, where it backed and advanced many times in order to get to a spot, mentioned earlier, that allowed the entrance to the parking spot.

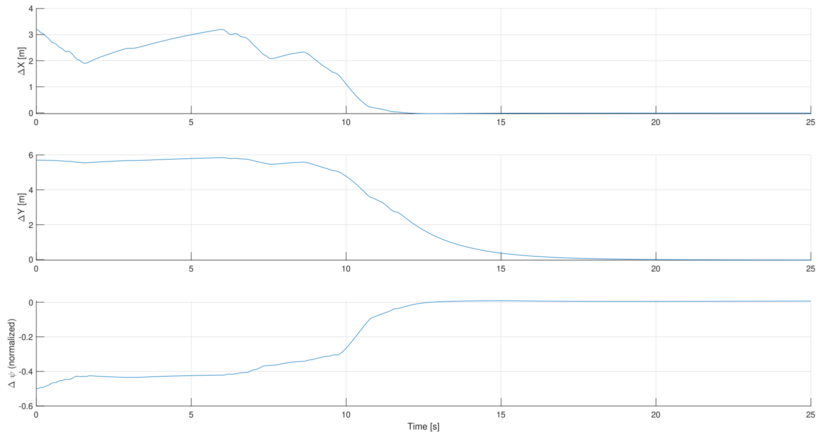

Figure 5.15: Error signal of part 2

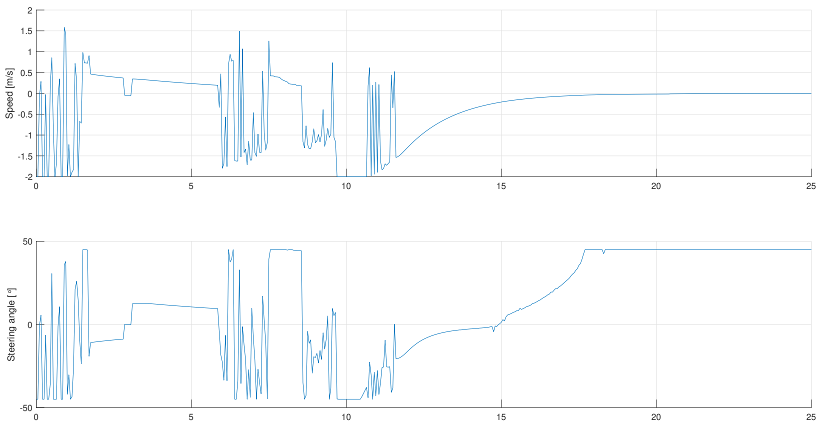

Figure 5.16: Commands signal of part 2

The final errors of both these simulations:

Tracing error	Part 1	Part 2
Δx [m]	-0.0002	-0.008
Δy [m]	-0.0084	-0.03
$\Delta\psi$ (normalized)	-0.002	0.0068

Table 5.8: Tracking error of maneuver 2

5.4 Case study 3: Angle Parking

The results:

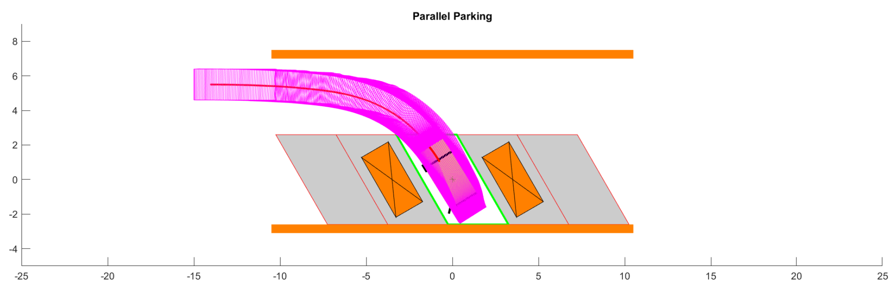

Figure 5.17: Trajectory maneuver 3

Results of this maneuver and expectations for this maneuver to run smoothly match. Moreover, it is worthy to highlight the fact that the smooth constraints and its simplicity give the controller an advantage over the different maneuvers.

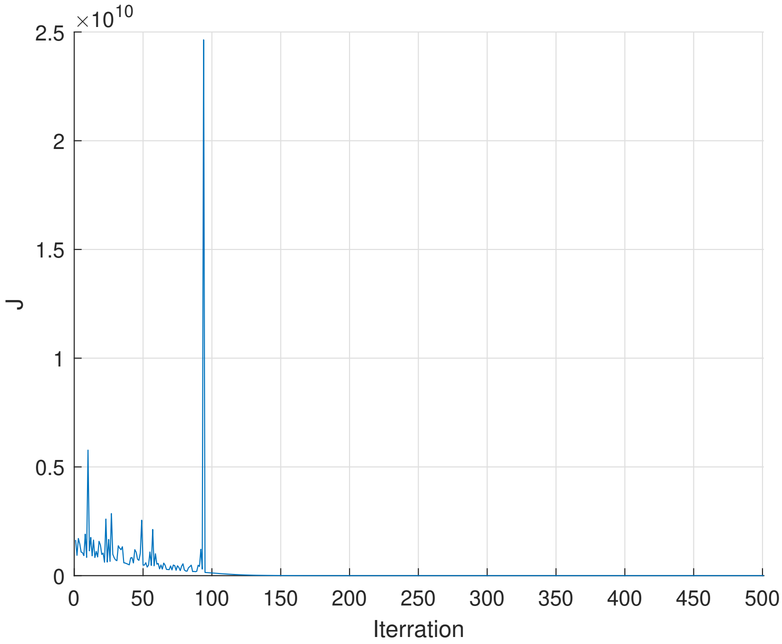

Figure 5.18: Cost function of maneuver 3

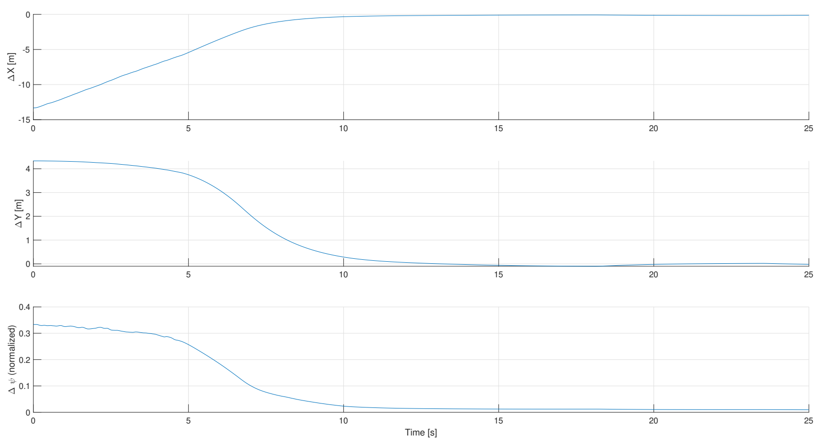

Figure 5.19: Error signal of part 1

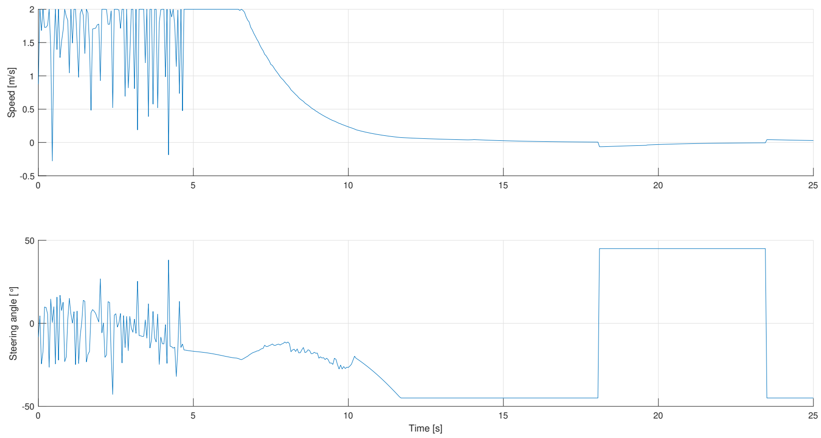

Figure 5.20: Commands signal of part 1

The final errors of this simulation:

Tracing error	Maneuver 3
Δx [m]	-0.13
Δy [m]	-0.021
$\Delta\psi$ [rad]	0.0095

Table 5.9: Tracking error of maneuver 3

Chapter 6

Conclusion

Autonomous driving has the potential nowadays to significantly impact our society in the upcoming years. On the positive ends, the number of vehicle crashes could be reduced and mobility will be more accessible for those currently unable to drive. In this scope, research efforts are being invested more and more by automotive companies and academic institutions in order to develop vehicles with a high level of autonomy.

Automated parking system is an autonomous car-maneuvering system that moves a vehicle into a parking spot whether it is a parallel, perpendicular or angle parking. Its main advantages reside in enhancing the comfort and the safety of driving in constrained environments. In this context, the thesis addressed the autonomous parallel, perpendicular and angle parking via a Nonlinear Model Predictive Control. In particular, the aim was to develop a general and unique NMPC framework capable of performing all the parking manoeuvres. To attain that, several tests were implemented, each presenting a different configuration of the parameters characterizing the NMPC. At the end of this fine-tuning operation, the configuration defined in table 5.6 proved to be the best one in terms of robustness and high success rate in each type of parking maneuvers taken into account. Indeed, providing different initial positions and targets during a Monte Carlo campaign, it was always able to generate the optimal trajectory and commands, guiding the ego vehicle into the parking zone. Note that this set-up was successful in entering the parking zone without trespassing into the adjacent parking spots region, regardless of the type of parking scenario considered.

Further studies and tests should be established taking into account more hindrance, such as smaller parking spaces on one hand, and autonomous vehicle perception (involving the collection of data from vehicle sensors and the processing of this data into an understanding of the space around the vehicle) on the other hand.

In conclusion, Autonomous Parking Systems offer many advantages in reducing the chances of collisions when driving. The primary benefits of automated parking systems compared to conventional parking methods are the fact of needing less land area, smaller building volume, high efficiency and reduction of traffic jams on the streets. This technology ensures safe parking and contributes greatly to a faster, more convenient and hassle-free parking experience. For all these reasons and thanks to well-known environments and relatively low risks, autonomous parking may be the first fully autonomous application in the near future. In this regard, the thesis demonstrated the effectiveness of using the Nonlinear Model Predictive Control, hinting that it could become a key technology in the framework of autonomous guidance.

Bibliography

[1] K. Shirokinskiy, W. Bernhart, and S. Keese. *ADVANCED DRIVER-ASSISTANCE SYSTEMS: A UBIQUITOUS TECHNOLOGY FOR THE FUTURE OF VEHICLES*. Mar. 2021. URL: https://www.rolandberger.com/en/Insights/Publications/Advanced-Driver-Assistance-Systems-A-ubiquitous-technology-for-the-future-of.html (cit. on p. 1).

[2] SAE International. *SAE International Releases Updated Visual Chart for Its "Levels of Driving Automation" Standard for Self-Driving Vehicles*. Dec. 2018. URL: https://www.sae.org/news/press-room/2018/12/sae-international-releases-updated-visual-chart-for-its-%E2%80%9Clevels-of-driving-automation%E2%80%9D-standard-for-self-driving-vehicles (cit. on p. 2).

[3] CNN. *Toyota unveils car that parks itself*. 2003. URL: http://edition.cnn.com/2003/TECH/ptech/09/01/toyota.prius.reut/index.html (cit. on p. 3).

[4] Joe Wiesenfelder. *The Lexus LS460: It Really Does Park Itself*. 2006. URL: https://www.cars.com/articles/the-lexus-ls460-it-really-does-park-itself-1420663331314/ (cit. on p. 3).

[5] Edward Cook. *Top 10: Best Self-Parking Cars in 2022*. 2022. URL: https://www.evanshalshaw.com/blog/best-self-parking-cars/ (cit. on p. 3).

[6] The MathWorks Inc. *Vehicle Body 3DOF*. URL: https://it.mathworks.com/help/vdynblks/ref/vehiclebody3dof.html?s_tid=srchtitle_Vehicle%20Body%203DOF_1 (cit. on p. 5).

[7] Carlo Novara. «Lecture 10: Nonlinear model predictive control». In: *Nonlinear control and aerospace applications* (cit. on p. 9).

[8] Lars Grüne and Jürgen Pannek. *Nonlinear Model Predictive Control*. London, UK: Springer Science & Business Media, 2011 (cit. on p. 9).

[9] Eduardo F. Camacho and Carlos Bordons Alba. *Model Predictive Control*. Second. London, UK: Springer Science & Business Media, 2013 (cit. on p. 9).

[10] Richard E. Bellman. *Dynamic Programming*. Princeton, USA: Princeton University Press, 1957 (cit. on p. 20).

[11] Rolf Findeisen and Frank Allgöwer. «An Introduction to Nonlinear Model Predictive Control». In: *21st Benelux Meeting on Systems and Control*. 2002, pp. 1–141 (cit. on p. 21).

[12] Law Insider Inc. *Parking Space definition*. URL: `https://www.lawinsider.com/dictionary/parking-space` (cit. on p. 22).

[13] Andrea Trivella. «Transformation Matrices: Translations and Rotations». In: *Numerical Modelling and Simulation* (cit. on p. 24).

[14] The MathWorks Inc. *heaviside*. URL: `https://it.mathworks.com/help/symbolic/heaviside.html?s_tid=srchtitle_heaviside_1` (cit. on p. 27).

[15] Bai Li, KexinWang, and Zhijiang Shao. «Time-Optimal Maneuver Planning in Automatic Parallel Parking Using a Simultaneous Dynamic Optimization Approach». In: *IEEE TRANSACTIONS ON INTELLIGENT TRANSPORTATION SYSTEMS* 17 (Nov. 2016), pp. 3263–3274 (cit. on p. 29).