

Politecnico di Torino

Master's Degree course in Mechatronic Engineering



**Politecnico
di Torino**

Master's Degree Thesis

An Economic Predictive Control approach for Autonomous EVs in Adaptive Cruise Control scenario

Supervisors

Prof. Michele Pagone

Prof. Stefano Malan

Candidate

Alessandro Gaetano Cannatella

Academic Year 2024/2025

Abstract

Concerns about the effects of climate change have recently driven research across various fields of applications. The automotive transportation sector is one of the most interesting from this point of view: government laws and regulatory policies push the automotive companies to invest in diminishing the emissions of the means of transportation. As matter of facts, the automotive industry has intensified its effort in development of Electric Vehicles (EVs) or Hybrid Electric Vehicles (HEVs) to be compliant with the new era of the transportation sector.

Although the EVs represent a promising solution to ensure a more sustainable form of transportation, their limited driving range is a critical aspect, which calls for further investigation. Despite their higher energy efficiency compared to vehicles equipped with Internal Combustion Engines (ICEs), the storage of electric energy presents greater challenges than conventional fuel storage. This limitation needs to have a thrifty usage of the electric power. Furthermore, the time required to recharge a battery is significantly longer than the time needed to refuel a traditional fuel tank, representing another obstacle to the widespread adoption of EVs across various sectors.

To address these necessities, the following work proposes a Nonlinear Model Predictive Control (NMPC) strategy for the lateral and longitudinal control of a vehicle dynamics and then shifts to an Economic NMPC to achieve simultaneously optimal control performances and energy saving.

Tracking performances, comfort and safety considerations, and energy saving are opposing objectives: often, focusing solely on one objective might cause significant degradation of the other objective performance. Hence, the main objective of the thesis is to demonstrate the effectiveness of the economic approach in finding a compromise in the control action which can reduce the energy consumption and, at the same time, featuring satisfying tracking performances.

The control algorithm is developed in a software environment, employing two main methodologies: a traditional tracking-based NMPC and an Economic NMPC. The controllers are tested in a simulated environment using real-world data for the controlled vehicle. Then the two controllers are compared from the standpoints of tracking performances, passenger safety and comfort and energy consumptions.

Simulation results show the potential of the economic approach in different scenarios, with increasing level of complexity. Additionally, the EMPC simplifies the cost function, maintaining the same constraint set, such that the computational effort of the problem considerably decreases.

Contents

List of Figures	VI
List of Tables	VIII
Symbols	IX
Acronyms	XII
1 Introduction	1
1.1 Problem Statement and Motivation	2
1.2 Objectives	3
1.3 Literature Review	4
1.4 Methodology	7
1.5 Thesis Structure	9
2 Fundamentals of Electric Vehicles	11
2.1 Battery Management System	11
2.2 Inverter and DC-DC converter	12
2.3 Motor Control Unit and Vehicle Control Unit	13
2.4 Thermal Management	14
2.5 Electric Motor and Transmission	15
2.6 On-Board Charger	16
3 Physical Model of the System	17
3.1 Vehicle Parameters	17
3.2 Longitudinal and Lateral Dynamics Model	18
3.2.1 Longitudinal Dynamics	18
3.2.2 Lateral Dynamics	22
3.3 Gearbox Model	26
3.4 Wheel Model	27
3.5 Electric Motor Model	28

3.6	Battery Model	30
3.7	State Space Representation	34
4	NMPC and EMPC Framework	37
4.1	General MPC formulation	39
4.2	Tracking NMPC	40
4.3	Economic Nonlinear MPC Formulation	42
4.4	Optimization variables	43
4.5	Terminal Cost	44
4.6	Constraints	45
4.6.1	Terminal Constraints	49
4.7	Recursive Feasibility and Asymptotic Stability	51
4.7.1	Recursive Feasibility	51
4.7.2	Asymptotic Stability	53
4.8	Design of Stabilizing Term	55
5	Controller Implementation, Simulation Results	59
5.1	NMPC Design	59
5.1.1	NMPC Cost Function	60
5.2	Constraints Formulation	66
5.2.1	Hard Constraint for the NMPC Controller	67
5.3	EMPC Design	72
5.3.1	EMPC Cost Function	72
5.3.2	Hard Constraints for the EMPC Controller	74
5.4	Practical implementation on MATLAB	75
5.4.1	Selecting the Sampling Time	75
5.4.2	Setting of the Solver	76
5.4.3	Numerical Approximation to facilitate the Solver	77
5.5	Tuning and Simulation Results	78
5.6	Simulation on Reference Trajectory 000	79
5.7	Simulation on Reference Trajectory 001	84
5.7.1	NMPC vs EMPC on Reference Trajectory 001	90
5.8	Simulation on Reference Trajectory 002	94
5.9	Control inputs Analysis: NMPC vs EMPC	99
5.10	Controller Design with Averaging Conflicting Objectives Approach	101
5.10.1	Cost Function for selecting the trade-off equilibrium point	102
5.10.2	Cost Function for selecting the weight of the additional term	103
5.10.3	Cost Function for finding the optimal control action	104
5.10.4	Constraints of the Conflicting Objectives NMPC Controller	105
5.10.5	Hard constraints for Conflicting Objectives NMPC Controller	105

CONTENTS

IV

5.10.6 Comparison Between Conflicting Objectives NMPC and NMPC	108
5.10.7 Control Inputs behaviour in decreasing Prediction Horizon length for Conflicting Objectives NMPC	112
6 Conclusion and Further Developments	115
6.1 Further Developments	117
Bibliography	119

List of Figures

3.1	Scheme of the Dynamic Single-Track model, from [13].	22
3.2	Scheme of the reference frames in the Lateral Dynamics Model, from [13].	26
3.3	Efficiency map of the EM.	29
3.4	Experimental Data on Voltage and Resistance for a Single Cell in the Battery Pack, taken from[14].	32
4.1	Terminal constraint effect.	50
5.1	The reference trajectory and corresponding speed profile employed in the first set of simulations, on Reference Trajectory 000. .	81
5.2	NMPC tests on Reference Trajectory 000 for choosing N_p . .	83
5.3	The reference trajectory and corresponding speed profile employed in the second set of simulations, on Reference Trajectory 001.	86
5.4	SOC behaviour and Power Ratio profile for tests in Table 5.12, on Reference Trajectory 001.	92
5.5	Tracking performances comparison for tests in Table 5.12 on Reference Trajectory 001.	93
5.6	The reference trajectory and corresponding speed profile employed in the third set of simulations, on Reference Trajectory 002. .	95
5.7	Tracking performances comparison for tests in Table 5.14 and Table 5.15 on Reference Trajectory 002.	97
5.8	Comfort performances for tests in Table 5.14 and Table 5.15 on Reference Trajectory 002.	98
5.9	Distance between leading and controlled vehicle for tests in Table 5.14 and Table 5.15 on Reference Trajectory 002.	99
5.10	T_{EM} behaviour for tests in Table 5.14 and Table 5.15 on Reference Trajectory 002.	100
5.11	δ_f behaviour for tests in Table 5.14 and Table 5.15 on Reference Trajectory 002.	101

5.12 Distance between vehicle for the Averaging Conflicting Objectives approach (Tests Table 5.20) with $N_p = 20$, on Reference Trajectory 001.	112
5.13 T_{EM} behaviour for tests in Table 5.20 and Table 5.17, on Reference Trajectory 001.	114
5.14 δ_f behaviour for tests in Table 5.20 and Table 5.17, on Reference Trajectory 001.	114

List of Tables

3.1	Nominal parameters of the EV to be controlled: Fiat 500e.	18
3.2	Value of cornering stiffness and load factor, taken from [13]. . . .	24
3.3	Battery Nominal parameters.	33
5.1	Bounds of the inequality constraints for all controllers.	72
5.2	Characteristics of Reference Trajectory 000.	80
5.3	Initial conditions Reference Trajectory 000.	80
5.4	Weights for selecting the prediction horizon length.	82
5.5	Tests for choosing the prediction horizon of NMPC on Reference Trajectory 000.	84
5.6	Characteristics of Reference Trajectory 001.	85
5.7	Initial conditions for all the simulations on Reference Trajectory 001.	87
5.8	Tuning results of the NMPC controller, on Reference Trajectory 001.	87
5.9	Average computational time of NMPC controller for different trials on Reference Trajectory 001.	88
5.10	Tuning results of the EMPC controller on Reference Trajectory 001.	89
5.11	Evaluation of the EMPC Controller's best performances on Reference Trajectory 001.	90
5.12	Resume of bests EMPC tests vs benchmark NMPC on Reference Trajectory 001.	90
5.13	Characteristics of Reference Trajectory 002.	94
5.14	Results of NMPC controller on Reference Trajectory 002. . . .	94
5.15	Performance results of EMPC controller on Reference Trajectory 002.	94
5.16	Tuning results of the NMPC controller with $N_p = 22$, on Reference Trajectory 001.	109
5.17	Tuning results of the controller with the conflicting objectives, $N_p = 22$, on Reference Trajectory 001, tracking focus.	110

5.18	Tuning results of the controller with the conflicting objectives, $N_p = 22$, on Reference Trajectory 001, energy consumption focus	110
5.19	Tuning results of the NMPC controller with $N_p = 20$, on Reference Trajectory 001, tracking focus.	111
5.20	Tuning results of the controller with the conflicting objectives, $N_p = 20$, on Reference Trajectory 001, tracking focus.	111

Symbols

β_f	Slip angle on the front wheels	$\mathcal{U}$	Set of the inputs
β_r	Slip angle on the rear wheels	$\mathcal{X}$	Set of the states
δ_f	Steering angle	$\mathcal{X}_{N_p}$	Domain of attraction
$\Delta\delta_f$	Steering Angle error	$\mathcal{Z}$	Manifold of states and inputs
$\Delta\zeta$	SOC Rate	ω	angular speed (generic)
Δ_a	Acceleration error	ω_ψ	Yaw rate of the vehicle
Δ_d	Safety Distance error	ω_{EM}	angular speed of the EM
Δ_v	Speed error	ω_w	angular speed of the wheels
ℓ	Stage cost (generic)	ψ	Heading of the vehicle (generic)
η	Efficiency (generic)	ρ	Air density
η_b^{conv}	Converter efficiency	τ_{gb}	Gearbox ratio
η_b^{coul}	Coulumbic efficiency of the battery	θ	Road inclination
η_f	Vertical load factor on the front wheels	ξ	Vector of states
η_r	Vertical load factor on the rear wheels	ζ	State of Charge of the battery (SOC)
η_{EM}	Electric motor efficiency	A_f	Frontal surface of the vehicle
η_{inv}	Inverter Efficiency	a_x	Longitudinal acceleration of the vehicle
η_{tr}	Efficiency of the transmission system	a_y	Lateral acceleration of the vehicle
		C_d	Drag Coefficient
		C_f	Cornering stiffness on the front
		C_r	Cornering stiffness on the rear
		e_{ct}	Cross-Track error
		e_h	Heading error
		F	Force (generic)
		f_r	Rolling resistance coefficient

SYMBOLS

F_{aero}	Aerodynamic Drag	P_b^{up}	Maximum value of power request at the battery
F_{roll}	Rolling resistance force	P_b	Power request at the battery
F_{slope}	Gravitational resistance force	P_{EM}	Electric motor power
F_{trac}	Traction force	P_{ref}	Reference Power
F_{yf}	Lateral force on the front axle of the vehicle	Q	Weight matrix of the first 4 states for NMPC
F_{yr}	Lateral force on the rear axle of the vehicle	Q_b	Battery charge
g	Gravitational acceleration	Q_{nom}	Nominal battery capacity
I	Moment of inertia of the vehicle	r	Wheel radius
I_b	Battery current	$R_b^{o,s}$	Internal resistance of the single cell
J	Cost function (generic)	R_b^o	Internal resistance of the total battery
k	Simulation time step	T	Torque (generic)
k_p	Prediction horizon time step	T_w	Torque at the wheels
l_f	Distance CoG - front axle	T_{EM}	EM torque
l_r	Distance CoG - rear axle	T_s	Sampling Time
m	Mass of the vehicle	u	Vector of the control inputs
N_b	Total number of cells in series	u^*	Optimal sequence of control inputs
N_{par}	Number of cells in parallel	u^e	Equilibrium point for the inputs
N_p	Prediction horizon	V	Terminal term of the cost function (generic)
N_{ser}	Number of cells in series	$V_b^{oc,s}$	Open circuit voltage of the single cell
P	Terminal cost weight matrix	V_b^{oc}	Open circuit voltage of the total battery
P_b^{gen}	Power generated by the battery		
P_b^{low}	Minimum value of power request at the battery		

SYMBOLS

v_x	Longitudinal speed of the vehicle	W_h	Weight of the Heading error
		W_{Pow}	Weight of the Power Ratio
v_y	Lateral speed of the vehicle	W_v	Weight of the Speed error
W	Weight matrix (generic)	X	CoG position along the x -axis (generic)
W_{Δ_d}	Weight of the Safety Distance error	x^e	Equilibrium point for the states
$W_{\Delta_{\delta_f}}$	Weight of the Steering Angle error	$x_{1 \rightarrow 4}$	Vector of the first 4 states of the system (generic)
$W_{\Delta_{\xi}}$	Weight of the SOC Rate	x_k	States at time instant k (generic)
W_{Δ_a}	Weight of the Acceleration error	Y	CoG position along the y -axis (generic)
W_{ct}	Weight of the Cross-Track error		

Acronyms

ACC Adaptive Cruise Control	EM Electric Motor
LKA Lane Keeping Assist	MCU Motor Control Unit
MPC Model Predictive Control	VCU Vehicle Control Unit
NMPC Nonlinear Model Predictive Control	PID Proportional-Integrative-Derivative controller
EMPC Economic Model Predictive Control	LQR Linear Quadratic Regulators
LTV-MPC Linear Time-Varying Model Predictive Control	HEV Hybrid Electric Vehicle
CoG Center of Gravity	ICE Internal Combustion Engine
SOC State of Charge	ADAS Advanced Driver Assistance Systems
SOH State of Health	AEB Autonomous Emergency Braking
EV Electric Vehicle	BSM Blind Spot Monitoring
	RNN Recurrent Neural Network
	BMS Battery Management System
	OBC On-Board Charger
	DST Dynamic Single-Track model

Chapter 1

Introduction

In recent years the automotive industry has undergone a phase of great transformation. Growing awareness of the significant environmental impact of the transport sector, the urgent need to reduce greenhouse gas emissions, and the progressive depletion of fossil fuel resources have collectively pushed the industry towards a new direction.

The new reality to be faced has driven important changes in the technologies used in the vehicles, with an orientation to the electrification of transportation. The European Union has set important targets about this topic: from the 2035 on, only vehicles running exclusively on CO₂ neutral fuel can be registered, and a reduction of the emissions in the transport sector of 90% within the 2050 has to be achieved [15].

To meet these new requirements and comply with the objectives of the European Union, major manufacturers have increased the production of Electric Vehicles (EVs) and Hybrid Electric Vehicles (HEVs).

This work will focus specifically on Electric Vehicles, examining their potential, limitations, and control strategies to improve performance and reduce energy consumption.

1.1 Problem Statement and Motivation

The Electric Vehicles, despite their environment benefits and some mechanical advantages, have a major challenge: **limited range**. On average an EV can travel for 350-400 *km* with a single charge, while a vehicle with an Internal Combustion Engine (ICE) can overpass 650 *km* of autonomy with a full of fuel. Moreover, it is necessary to consider the slow recharge speed of an EV: the charging lasts hours, while a car equipped with an ICE can be refuelled in some minutes. The limited driving range has slowed down the widespread adoption of EVs, so nowadays a great focus on the **optimization of energy consumption** arises.

Another significant transformation in the automotive industry is the development of Advanced Driver Assistance Systems (ADAS). They encompass a range of technologies designed to enhance the safety and the comfort of the passengers during a travel, improving the overall driving experience by automating some functions of the vehicle.

Examples of ADAS systems are the following:

- **Adaptive Cruise Control (ACC):** This system is slightly different from the classic Cruise Control. The basic Cruise Control helps the vehicle to maintain a desired longitudinal speed of travel without the need, by the driver, to press the accelerator. It is used on long and straight/almost straight path (for example in highways) that are travelled at high speed (above 30 *km/h*). Whereas the ACC combines the functionalities of the Cruise Control with the capacity of maintaining a fixed (pre-defined by the driver) safety distance from the vehicle ahead. The control system will regulate acceleration and deceleration to achieve an optimal balance between the desired travel speed and the safety distance during road operation.
- **Lane Keeping Assist (LKA):** By applying small adjustments to the steering angle, this system helps the vehicle maintain its correct position within the lane during driving.
- **Autonomous Emergency Braking (AEB):** This system autonomously detects the sudden appearance of obstacles in front of the vehicle, alerts the

driver of the danger, and, if the driver does not respond fast enough, the system breaks to avoid or mitigate a collision.

- **Blind Spot Monitoring (BSM):** This system uses sensors to detect the presence of vehicles in the lateral blind spots of the driver during a travel, then warns the driver.
- etc.

Although fully autonomous vehicles are not yet widely available, all the ADAS are steps towards a future autonomous drive that, hopefully, will reduce road accidents and improve drive experience.

The following work will concentrate on the combination of the ACC and LKA, with particular focus on safety and energy efficiency.

1.2 Objectives

The combination of ACC and LKA enables the development of a control system capable of **following a leading vehicle** on a road that is not straight, while maintaining a safe distance from the vehicle ahead. The control system will:

- follow the trajectory of the leading vehicle, using small correction of the Steering Angle.
- maintain a safety distance from the leading vehicle.
- adjust acceleration and deceleration to adapt its own velocity to the speed profile of the leading vehicle.

The following work aims to:

- Analyse the lateral and longitudinal dynamics of an EV and its electric powertrain, highlighting the main nonlinear behaviours of the model, which influence the control system mentioned above.
- Formulate an appropriate NMPC control problem and address it by means of a suitable objective function, considering physical constraints and control input limits.

- **Address the problem with an EMPC approach in order to reach a better optimization of the energy consumption, with tracking performances comparable with the NMPC approach.**
- Implement and simulate the controllers in a software environment. MATLAB will be used to develop the controller, and CasADi [11] functionalities will be employed to solve the constrained optimization problem.

The ACC and the LKA, which are used in the development of the controller in the following, are slightly different from the ones available on the market. The LKA systems usually have a camera to identify the lane, and when the vehicle approaches the boundaries the control action is engaged. In the following design, another vehicle highlights the right trajectory to follow, not the horizontal signage.

Regarding the ACC, the system should be able to guarantee the safety distance from the car ahead and tracking of a reference speed also in the situation in which the vehicle ahead changes. For example, if the controlled vehicle was overtaken the ACC will decelerate to restore a safe distance from this new vehicle in front. Instead, the controller that will be developed should track as faithfully as possible the speed profile of a single leading vehicle, no overtake or other vehicles will enter in the scenario.

Moreover, another design will be carried out. As we will see, the stability of a MPC-based controller is a challenging part of the design, but some techniques can be employed to reach it. Among the classical approaches for studying stability, for both NMPC and EMPC, the using of a terminal cost and a terminal constraint is quite effective. Nevertheless, new procedures are under study, particular interesting is a novel approach proposed in [3]. This last method, with some difference with respect the one presented by Calogero et al. [3], will be employed in the following work to design another controller, named *Conflicting Objectives NMPC*.

1.3 Literature Review

Nowadays, various control strategies are employed in the automotive industry, in particular the Proportional-Integrative-Derivative (PID) controllers and the Linear

Quadratic Regulators (LQR) controllers are widely adopted.

PID controllers are particularly appreciated for their ease of implementation, robustness and reliability. As detailed in [20], the PID controllers are popular in different branches of industry, including automotive applications. Over time, different sets of rules for tuning have been developed to optimize their performances under different operating conditions. They are used in many systems in the automotive domain, such as the speed control in an EM, the basic Cruise Control and the ACC. Reference, in [20] different variants of PID applied to the ACC are listed with their performances.

Despite their effectiveness and simplicity the PID controllers exhibit limitations: they have to be accurately tuned for different scenarios, so **adaptation difficulties** arise. Moreover, they struggle to catch and handle nonlinear effects, which, unfortunately, are common in the dynamics of a vehicle.

Regarding the LQR, it is based on the minimization of a quadratic cost function that balances the different requests of the system: performances and control effort, which are often at odds. The LQR is effectively used for controlling the lateral dynamics of a vehicle and in the yaw control. In its basic form the LQR does not handle constraints on the states or on the control inputs, so it has the same problems of the PID from this point of view. Although LQR strategies rely on a model of the system, allowing for a more accurate representation of its behaviour compared to PID controllers, they are based on the assumption that the system is linear. This assumption does not hold in many real-world applications, where system dynamics are strongly nonlinear.

Due to the need to handle complex and nonlinear model in an effectively way and the presence of physical constraints, the research shifts its attention on MPC strategies. As we said the energy optimization is one of the most important field of research not only in the automotive industry, but in general in the transport sector. The research spans different areas, from powertrain optimization to energy management systems using different types of energy sources. Most of these optimization problems must respect physical **constraints**, both on the system states and on the control inputs provided by the actuators. As highlighted in [17], the high performances of a control system are often linked to the proper handling of such constraints. The capability of the MPC approach to solve **constrained opti-**

mization problems is one of its key features that encourages its adoption. Additionally, Model Predictive Control (MPC) naturally evolves into Nonlinear Model Predictive Control (NMPC), which is capable of catching the system's nonlinear behaviour, provided that the adopted prediction model is sufficiently accurate.

An interesting comparison between PID controller and MPC-based controller is detailed in the particular application of the ACC in [12], where the potential of the MPC comes out. In a simulation environment, Nie and Farzaneh [12] show how the MPC approach can improve the fuel economy (above 10%) with respect to the PID in different driving scenarios.

Another example of MPC-based control strategy is described in [2], the particular application is the power management in power-split HEV. They demonstrate the capability of the MPC to catch the nonlinearities of the plant. Borhan et. al [2] use two slightly different methods to improve the performance of existing commercial power management techniques. The two approaches are: a Linear Time-Varying MPC (LTV-MPC) and a Nonlinear MPC (NMPC). In particular the LTV-MPC shows results comparable with the existing control method, but the linearization cannot capture the complexity of the model in a suitable way, introducing error. Thus, they demonstrate how NMPC can reach a fuel economy improvement with respect the available controller in commercial PSAT and, also, with respect to the LTV-MPC controller. Unfortunately, the computational demand is not a side issue in a sensitive application like the management of a vehicle in operation, so the high computational time of a NMPC approach must be faced, and reduced if possible.

The high computational demand derives from the nonlinear model used by the MPC to infer the optimal control action: the optimization problem and its constraints are based on the model, if it is complex the entire problem is challenging to solve, even for advanced solvers. In order to reduce the complexity of the model to be solved a Recurrent Neural Network (RNN) can be used to approximate the nonlinear model, then the NMPC will use the RNN as the approximation of the real physical model to solve the dynamics on the prediction horizon in an easier way. Starting on this basic idea Pereira et al. [16] use a RNN to model a proton exchange membrane of a Fuel Cell Hybrid Electric Vehicles (FCHEV), then they interface the RNN with a NMPC to create an energy management system. As we

can see in [16] the NMPC approach can reach the desired performances, not only running in a real-time applications, but also using a low-cost hardware.

An interesting area of research is **adaptive control**, which is based on the principle of adjusting controller parameters in real time, following an online evaluation of the system to be controlled. The aim is to follow the changes of the system, although no mathematical model can incorporate neither all the possible mechanisms of the system, nor all the possible disturbances that can occur, some system changes can be evaluated and faced in a proper way. We can use different tunings to enhance the performance in different situations, while some controller are blocked in only a single configuration, which is a trade-off among all the possible setups. To better understand the concept we can think at the load of a vehicle: the total mass of a vehicle can change easily, more passengers or luggages can be loaded or unloaded. The changing of the mass, and the distribution of it, influences the entire dynamics of the vehicle and the energy demand. Here Xin et al. [19] have developed an interesting concept: using a recursive least square algorithm they estimate on-line the mass of a vehicle, so the mass is treated as a variable in the control algorithm. As shown in [19], by using a different tuning strategy to account for variations in mass, they design a controller that reduces additional fuel consumption compared to a deterministic parameter scheme, which cannot adapt to the change of the characteristics of the system since some features of the vehicle, like the mass, are treated as a constant.

1.4 Methodology

The work will focus on the control of an EV, the aim is modelling the car both in its longitudinal and lateral behaviour and from the standpoint of the electric powertrain. Since the primary goal is to develop a control algorithm, the model will be designed to achieve a high level of accuracy while maintaining manageable code complexity and ensuring affordable computational effort. Based on this principle, some characteristics of the system, like the efficiency of the EM, will be taken constant during each optimization problem even if they should be update also within the prediction horizon.

Regarding the model of the vehicle, a Single-Track scheme will be taken as

reference, and some of the relationships will be simplified under the working conditions. However, the main nonlinearities of system will be taken into account and the Single-Track model provides the ideal basis of the dynamics in order to retrieve accurate results. The Single-Track model does not capture some behaviours of the vehicle, like the suspension dynamics, the road-tire interaction phenomena etc., all these phenomena will be ignored consciously, since the aim is not to model perfectly the car in order to develop a control system for each single part of the vehicle, but the development of a control system for the total dynamics of the car.

After the study of the system model, real-world data taken from a Fiat 500e, owned by the Politecnico di Torino, will be used to shift the theoretical development to a more practical and quantitative work.

The working flow has the following steps:

- Lateral and longitudinal dynamics modelling, along with powertrain modelling, are developed with particular consideration for the aspects of the model that affect controller performances and accuracy.
- Designing of a NMPC controller with a *tracking* approach, it will be taken as a benchmark of performance.
- Testing of the NMPC controller on difference scenarios, in order to have an idea of the performances that can be reached with an affordable computational time and the limits of the controller.
- Design of a NMPC controller with an *economic* approach.
- Testing of the EMPC controller on the same scenarios of the NMPC.
- A comparison between the tracking NMPC and EMPC approaches in controller design.
- Design of a NMPC controller and an additional term by means of an Averaging Conflicting approach, as detailed in [3].
- A comparison between the classical tracking NMPC and Averaging Conflicting approaches in controller design.

The implementation phase and simulation phase are developed on MATLAB [18], while the optimization problems are solved by means of CasADi [11] functionalities.

The following scenarios were considered for evaluating system performance:

- **Reference Trajectory 000:** The first scenario is a repetitive path, built in a sinusoid shape. It is travelled at constant longitudinal speed.
- **Reference Trajectory 001:** The second scenario is more realistic, the path resembles a real road in which there are wide-radius curves and extended, nearly straight segments. The trajectory is travelled at different longitudinal speed.
- **Reference Trajectory 002:** The third scenario is similar to the second one, but with a longer distance to be travelled. The longer distance can show how the approach can influence long trips in which the autonomy of the vehicle is fundamental.

1.5 Thesis Structure

The thesis follows a logical flow: starting with the basics knowledge about an EV, then the physical model of the system is explained in detail. After outlining the adopted control theory, the implementation of the controller, for each control strategy (tracking NMPC, EMPC and Averaging Conflicting Objectives), is presented. In the end the results of the simulations are analysed.

The organization of the chapters is the following:

- **Chapter 2 - Fundamentals of Electric Vehicles:** the chapter provides the basics knowledge on an EV, describing its most important components. In this chapter we can understand how the major components of the vehicle work and how they interact, moreover the differences with a vehicle equipped with an ICE are highlighted.
- **Chapter 3 - Physical Model of the System:** in this chapter the physical model of the lateral and longitudinal dynamics of the vehicle is described,

then the relationships with the electric powertrain in terms of rate of State of Charge (SOC) of the battery and power are detailed. In this chapter can be found the equations of the model used in the model predictive approach.

- **Chapter 4 - NMPC and EMPC Framework:** in this chapter theory of the advanced control technique named MPC is resumed. Particular focus is placed on the control approaches adopted: tracking Nonlinear Model Predictive Control (NMPC), Economic Nonlinear Model Predictive Control (EMPC) and Averaging Conflicting Objectives.
- **Chapter 5 - Controller Implementation, Simulation Results:** the chapter focuses on the implementation of controllers, with different approaches, in the software environment MATLAB. Here the cost function design, the constraints formulation and the practical implementation tools and challenges are detailed. Then simulations on different patterns are reported and analysed to understand the different performances of each controller.
- **Chapter 6 - Conclusions and Further Developments:** the final chapter discusses the final results, highlighting strengths and weaknesses of the approaches, and possible future extensions of the work are discussed.

Chapter 2

Fundamentals of Electric Vehicles

As stated in [6], the number of battery-only electric passenger cars in European Union countries grew twelvefold from 2018 to 2023, making them a new reality to be addressed on our roads. The increasing number of EVs, and in general of vehicles that use low-emission fuels, is responding to the demand for sustainable transportation. The shift of the transportation to the market of electric and hybrid vehicles affects both private and public transportation, then the automotive companies are significantly investing towards electric powertrain development.

To provide a basic understanding of these technologies, the main components of an electric vehicle's powertrain are introduced in the following chapter. The reader will gain insight into the interaction of these elements, highlighting the substantial differences compared to vehicles equipped with an Internal Combustion Engines (ICE).

2.1 Battery Management System

The Battery Management System (BMS) continuously monitors the parameters of the battery cells to ensure safety and optimal operations. The BMS supervises functions like:

- Tracking of the State of Charge (SOC) of the battery to compute an estimation of the available energy in-real time.

- Tracking of the State of Health (SOH) of the battery in order to monitor the ageing phenomena of the battery.
- Temperature control, it is essential to ensure safe operations, as both excessively high and low temperatures can lead to premature degradation or even trigger thermal runaway.
- Protection against overcharging and deep discharging.

The BMS monitors also the behaviour of each cells of the battery: Lithium-ion batteries are made up of several cells, they are connected in series or in parallel to built the total battery. Unfortunately, due to tolerance of manufacturing, ageing or different temperatures between cells, each unit can have different speed of charging or discharging, or different voltage level. The different behaviour of the cells can lead to degradation, temperatures increasing, potential swelling and other safety issues. The BMS maintains the balancing between the different cells to avoid arising of these problems.

In the end the BMS ensures an efficient and safe utilization of the battery, while aiming to maximize the component's lifespan.

2.2 Inverter and DC-DC converter

The EV powertrain contains two main power conversion devices:

- **Inverter.**
- **DC-DC converter.**

The battery pack provides electric energy in DC form, but most of the Electric Motor available works only in AC; then the Inverter has the function to convert the electric energy in DC form to energy in AC form. As with any energy conversion between different forms, a portion of energy is inevitably lost (as heat). However, the Inverter has a high efficiency as we can see in Table 3.3, minimizing this losses. The Inverter works in both direction of power flow: when the vehicle needs power to move forward, the power flow goes from the battery to the motor, and then to the wheels; but during a travel also deceleration and braking can occur, in this latter

case the motor acts as a generator providing power to the Inverter that converts the energy from the AC form to DC form to charge the battery. When a braking torque is exploited to recharge the battery pack the vehicle is in *regenerative braking mode*.

The DC-DC converter transforms the high voltage (100-400 V) power provided by the battery pack to a low voltage power (12-24 V) to feed other services of the car such as lights, air conditioning, infotainment devices etc. Basically it is used to step-down the high voltage, i.e. a *buck* converter.

It can be present also a DC-DC *boost* converter, its aim is to step-up the voltage at the output of the battery pack in order to have a power flow at high voltage and low current feeding the EM. The boost converter can be used in some vehicles, but since it is not always present we will not consider it in the prediction model, whereas the buck converter must be taken into account due to its power conversion efficiency: η_b^{conv} .

2.3 Motor Control Unit and Vehicle Control Unit

The Vehicle Control Unit (VCU) and the Motor Control Unit (MCU) works together in an Electric Vehicle to ensure power delivery, efficiency, and overall drive. VCU can be seen as the brain of the vehicle: it works at high level, managing the entire vehicle. It computes the torque demand based on multiple parameters, including driver inputs, and governs the overall vehicle dynamics. These parameters are then used to optimize power delivery for efficiency, ensure traction control, and enhance regenerative braking to maximize energy recovery. The VCU is also involved in the thermal management: it monitors the different part of the power-train to prevent overheating, or engages cooling systems if needed.

Regarding the MCU it can be seen as the muscles of the vehicle: it works only on the motor. Once the VCU determines the appropriate strategy, it communicates a torque request to the MCU. The MCU then translates this command into precise control actions on the EM, ensuring smooth acceleration and deceleration. In this sense the MCU has also a role in the thermal management: it tries to minimize the motor current in order to minimize heat generation [5].

2.4 Thermal Management

The thermal management monitors and controls the temperatures of the critical components of the vehicle, in order to guarantee efficiency, longevity and safety. Power electronic devices, particularly inverters and converters, as well as the battery pack and the electric motor, operate within a specific temperature range that ensures safe and efficient usage of the components. For example, from [8] we can see that the battery has a temperature range of $-30^{\circ}\text{C}/60^{\circ}\text{C}$: it can reach these limits, anyway they are not temperatures of working of the vehicle. Regarding the battery pack, a too high temperature of the cells can lead to degradation or blazing, on the opposite a too low temperature can lead to lack of autonomy.

For the EM a high power demand is satisfied by a high torque, i.e. by a high current. A high current leads to high loss of energy due to the Joule effect, moreover the high rate of changes of the magnetic field, in the rotor or the stator, of the motor caused by the high rotational speed, generates hysteresis losses. In the end, all the losses generate heat, so the temperature of the motor must be controlled to do not overpass the limits: when the EM approaches high temperatures the insulation of the windings can degrade, then short circuits or other faults arise in the motor, additionally the internal resistance increases proportionally to the temperature leading to higher Joule losses.

The power electronic components (inverter and converter), in addition to the Joule losses, generate *switching losses*. They are generated by the transistors inside the devices. At high frequencies the switching losses increase, for instance this situation occurs at high loads or during a fast charge of the battery. Even if the cooling demand is less intensive for power electronic components, however they generate a significant heating.

The thermal management system relies on data collected by sensors embedded within each component to enable real-time monitoring. Its purpose is to maintain optimal operating temperatures through the intelligent use of cooling strategies, such as liquid cooling and forced-air ventilation. This system is fundamental to maintain the vehicle's overall reliability and extending its lifespan.

As observed, the causes of overheating in EVs differ significantly from those in vehicles equipped with an ICE, where the primary sources of heat are mechan-

ical friction and extremely high temperatures generated within the combustion chamber.

2.5 Electric Motor and Transmission

In general the vehicles that use an ICE have as major problem the need to adjust the torque and the rotational speed of the motor before reaching the wheels. The complexity of the transmission system in an ICE vehicle arises from the need to identify the optimal combination of clutch characteristics, gearbox configuration (with multiple transmission ratios), and accelerator control. This coordination is essential to address the following challenges:

- The optimal condition of work for an ICE is only around a determined value of rounds per minute. So, if we want to move the vehicle at a speed that is far from this optimal area we will lose a lot from the point of view of the efficiency. Then we need to adapt the speed of the motor, which is basically always the same for efficiency reasons, to the different speeds at which the wheels can rotate during a road trip.
- The ICE cannot provide high torque at low rotational speed, but it is a common situation during a travel in a vehicle: to start the motion we need a great torque (to overcome the inertia) at practically null speed.

These problems are not present in an Electric Motor:

- an EM has, in general, a large number of areas (or working points) at high efficiency at different rotational speeds.
- an EM can provide a high torque even at very low speed, or at practically null speed.

Thanks to the characteristics of the EM the transmission system of an Electric Vehicle is much simpler than a vehicle with an ICE. In an EV we do not need a complex gearbox with different transmission ratios, just one, to enlarge the torque, is sufficient. This great mechanical simplification leads to higher efficiency in the transmission of the power from the motor to the wheels: less mechanical friction

is present, whereas the efficiency of the transmission system of a vehicle with an ICE rarely reaches the 90%, in an EV easily the 90% is overcome. The vehicle that is taken into account for the simulation part in the following chapters has a transmission efficiency, including the gearbox efficiency, of 97%, as we can see in Table 3.1.

2.6 On-Board Charger

The battery of an Electric Vehicle is recharged using direct current (DC) electrical energy. However, long-distance power transmission through the electrical grid is typically carried out in alternating current (AC) form. As a result, at charging points connected to the grid, the available power is AC. To address this mismatch, the vehicle is equipped with an On-Board Charger (OBC), which converts the AC power from the grid into DC power suitable for charging the battery pack.

The OBC manages the AC-to-DC conversion process safely and efficiently, adapting to different charging levels and monitoring the voltage and current to protect the battery's health [7]. The presence of OBC allows a *slow recharge* during overnight stays at home or other long stops.

On the other hand there is also the need to have a *fast recharge* in some situations: in these cases DC fast charger can be used. They bypass the OBC entirely, delivering DC power directly to the battery at much higher rates, reducing significantly charging time according to [7].

The following work does not focus on the OBC control system or its interaction with other components to ensure the safe and reliable operation of the EV.

Chapter 3

Physical Model of the System

In the MPC approach the model of the system is fundamental: the prediction of the events is based on an approximated model of the real system. Unfortunately, a very precise and exhaustive model will lead to an unaffordable computational demand, moreover it is practically impossible to model a physical system in a perfect manner and anyway unpredictable disturbance can play a role in the real physical system. Hence, the physical model of the system must be a trade-off between precise modelling and simplicity:

- A high-precision model leads to good prediction and then to effective control response. But a overly precise model is unmanageable.
- A simple model leads to manageable computational demand. But a overly simple model will differ significantly from the real system and fail to capture the behaviour of the physical quantities needed for computing the right control action.

3.1 Vehicle Parameters

The vehicle parameters used in the simulations and in the tuning of the controller are taken from a Fiat 500e owned by the Politecnico di Torino, they are shown in Table 3.1.

Parameter	Symbol	Value	Unit
Vehicle mass	m	1400	kg
Moment of inertia (estimated)	I	1867	$kg \cdot m^2$
Distance between CoG and front axle	l_f	1.2	m
Distance between CoG and rear axle	l_r	1.4	m
Frontal surface of the vehicle	A_f	2.15	m^2
Drag Coefficient	C_d	0.33	-
Wheel Radius	r	0.3	m
Transmission ratio of the gearbox	τ_{gb}	9.6	-
Efficiency of the transmission	η_{tr}	0.97	-
Maximum torque provided by the EM	T_{EM}^{max}	280	Nm
Minimum torque provided by the EM	T_{EM}^{min}	-280	Nm

Table 3.1: Nominal parameters of the EV to be controlled: Fiat 500e.

Note 3.1

In Table 3.1 the vehicle mass is equal to 1400 kg , while in [8] the kerb weight is 1290 kg . Adding the weight for a single passengers (80 kg) and the weight for luggages, we round the car's weight to 1400 kg .

3.2 Longitudinal and Lateral Dynamics Model

In the following section, the longitudinal and lateral dynamics of the vehicle are described, and the interaction between these two types of motion is highlighted.

3.2.1 Longitudinal Dynamics

The Longitudinal Dynamics is used to design the Adaptive Cruise Control part of the autonomous drive. The model is based on the Newton's Second Law of Dynamic, in which the mass of the vehicle is considered as concentrated in the Center of Gravity (CoG); therefore all the forces are applied on the CoG. The model can be expressed by the equation:

$$m \cdot \dot{v}_x = F_{trac} - (F_{aero} + F_{roll} + F_{slope}) \quad (3.1)$$

where:

- m is the vehicle mass, measured in $[kg]$;
- v_x is the longitudinal velocity of the vehicle measured in $[m/s]$, so $\dot{v}_x = a_x$ is the longitudinal acceleration of the vehicle measured in $[m/s^2]$;
- F_{trac} is the Traction Force provided by the power train, measured in $[N]$;
- F_{aero} is the Aerodynamic Resistance Force or Drag, measured in $[N]$;
- F_{roll} is the Rolling Resistance Force, measured in $[N]$;
- F_{slope} is the Slope Resistance Force due to road inclination with respect to the horizontal (road's slope), measured in $[N]$.

Traction Force

The input of the longitudinal model is the Traction Force F_{trac} , which is used by the car to move forward; but since the motor can be seen as a torque generator is more convenient use the EM Torque T_{EM} as input. The Traction Force and the torque provided by the motor are directly related, without considering losses:

$$F_{trac} = \frac{T_{EM}\tau_{gb}}{r} \quad (3.2)$$

we must consider the losses due to the transmission system of the vehicle, so the actual relation between the Traction Force and the EM Torque is:

$$F_{trac} = \frac{T_{EM}\tau_{gb}}{r}\eta_{tr} \quad (3.3)$$

where:

- τ_{gb} is the transmission ratio of the gearbox, for a car the gearbox is usually a speed reducer (torque multiplier): $\tau_{gb} > 1$.
- η_{tr} is the efficiency of the transmission system.
- r is the radius of the wheels, measured in $[m]$.

Aerodynamic Resistance

The Aerodynamic Resistance, also called Drag Force (F_{aero}), is caused by the air pushing against the vehicle as it moves forward. This force largely increases with the longitudinal speed of the vehicle, so it is the main opposition to win when the car moves at high speed, like in a highway. The Aerodynamic Resistance Force is modelled by the following equation:

$$F_{aero} = \frac{1}{2} \rho A_f C_d v_x^2 \quad (3.4)$$

where:

- ρ is the air density (typically 1.225 kg/m^3 at sea level);
- A_f is the frontal area of the vehicle, i.e. the surface that the vehicle exposes to the wind in a frontal view, measured in $[m^2]$;
- C_d is the drag coefficient;
- v_x is the longitudinal velocity of the vehicle, measured in $[m/s]$.

Most attention is typically directed towards the vehicle's dimensionless drag coefficient, C_d , which represents its aerodynamic efficiency.

Rolling Resistance

The Rolling Resistance Force (F_{roll}) is related to deformation of the tires as they roll on the road surface, the interaction tire-road generates an opposition to the forward motion of the car. Physically, tire deformation alters the pressure distribution across the contact patch between the tire and the road surface. This irregular distribution leads to energy dissipation, which ultimately manifests as a resistive force opposing the vehicle's motion. This force is typically modelled as:

$$F_{roll} = mg f_r \cos(\theta) \quad (3.5)$$

where:

- g is the gravitational acceleration, measured in $[m/s^2]$;

- f_r is the rolling resistance coefficient, it depends on several factors as tire type and material, tire pressure and road surface.
- θ is the road inclination with respect to the horizontal.
 - $\theta > 0$ indicates an uphill,
 - $\theta < 0$ indicates a downhill,

anyway the value of the rolling resistance doesn't change in the two cases (it derives from the cosine in the expression, which is an even function).

Slope Resistance

The Slope Resistance, also known as Gravitational Resistance, represents the component of the vehicle's weight that acts along the direction of motion travelling on an inclined road. This force can oppose to the motion, or can help the motion, or it can be not present based on the inclination of the road. Hence, F_{slope} plays a significant role in determining the required Traction Force needed to travel on a road, especially if it is extremely steep. When a vehicle travels on a road with an inclination of θ (according to the sign convention defined in the previous section), the gravitational force can be decomposed in two components:

- A vertical component, that is perpendicular to the road. It contributes to the normal force on the wheels.
- A longitudinal component, that acts on the direction that is parallel to the road: it opposes to the vehicle motion in case of uphill roads and it assists the vehicle motion in downhill scenarios.

The Slope Resistance corresponds to the longitudinal component of the gravitational force, and it is given by:

$$F_{slope} = mg \sin(\theta) \quad (3.6)$$

To simplify the simulation we will **consider a road with no inclination**, i.e. $\theta = 0$, so the Slope Resistance is null and the Rolling Resistance is constant.

3.2.2 Lateral Dynamics

The Lateral Dynamic Model is taken from [13], as we will see is related to the Longitudinal Dynamics by means of the longitudinal acceleration a_x .

The Lateral Dynamics formulation is based on the Dynamic Single-Track (DST) model. The DST model is characterized by a Single-Track (two wheels: one rear wheel, one front wheel), which is equivalent to a vehicle with four wheels, where the left part is equal to the right part as shown in Figure 3.1.

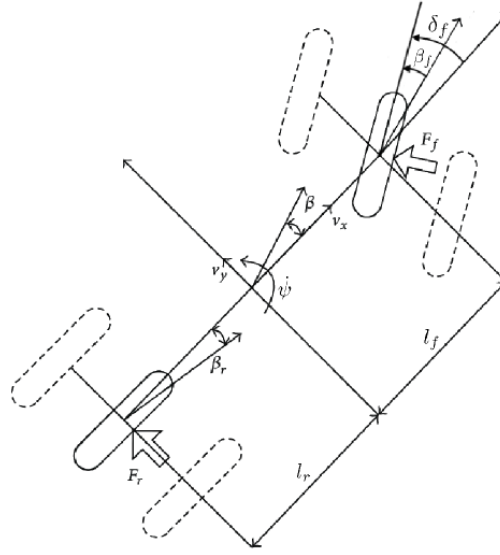


Figure 3.1: Scheme of the Dynamic Single-Track model, from [13].

The model has six states:

- $x_1 = X$.
- $x_2 = Y$.
- $x_3 = \psi$.
- $x_4 = v_x$.
- $x_5 = v_y$.
- $x_6 = \omega_\psi$.

The following equations describe the model:

$$\dot{x}_1 = \dot{X} = v_x \cos(\psi) - v_y \sin(\psi) \quad (3.7a)$$

$$\dot{x}_2 = \dot{Y} = v_x \sin(\psi) + v_y \cos(\psi) \quad (3.7b)$$

$$\dot{x}_3 = \dot{\psi} = \omega_\psi \quad (3.7c)$$

$$\dot{x}_4 = \dot{v}_x = v_y \omega_\psi + a_x \quad (3.7d)$$

$$\dot{x}_5 = \dot{v}_y = -v_x \omega_\psi + \frac{2}{m}(F_{yf} + F_{yr}) \quad (3.7e)$$

$$\dot{x}_6 = \dot{\omega}_\psi = \frac{2}{I}(l_f F_{yf} - l_r F_{yr}) \quad (3.7f)$$

where:

- X, Y : are the coordinates of the vehicle CoG in an inertial reference frame, both measured in $[m]$.
- ψ : is the heading of the vehicle, measured in $[rad]$.
- $\omega_\psi = \dot{\psi}$ is the yaw rate, measured in $[rad/s]$.
- $\vec{v}$: is the velocity vector in the inertial reference frame, measured in $[m/s]$, as shown in Figure 3.2.
 - v_x : is the longitudinal speed of the car, i.e. the $\vec{v}$ component along the longitudinal axis.
 - v_y : is the lateral speed of the car, i.e. the $\vec{v}$ component along the transverse axis.
- a_x : is the longitudinal acceleration in the inertial reference frame, measured in $[m/s^2]$. It is computed as indicated in (3.1), and relates Lateral and Longitudinal Dynamics.
- δ_f : is the Steering Angle, measured in $[rad]$.
- β : is the vehicle slip angle, i.e. the angle between the vehicle longitudinal axis and the velocity, measured in $[rad]$. β_f for the front wheels, β_r for the rear wheels.

- m, I : are mass in $[kg]$ and moment of inertia in $[kg \cdot m^2]$ of the vehicle.
- l_f : is the distance between CoG and front axle of the vehicle, measured in $[m]$.
- l_r : is the distance between CoG and rear axle of the vehicle, measured in $[m]$.
- C_f, η_f : are front cornering stiffnesses in $[N/rad]$ and front vertical load factor. Then $c_f = C_f \eta_f$, the values of these parameters are shown in Table 3.2.
- C_r, η_r : are rear cornering stiffnesses in $[N/rad]$ and rear vertical load factor. Then $c_r = C_r \eta_r$, the values of these parameters are shown in Table 3.2.
- F_{yr} and F_{yf} are the lateral forces exchanged between tire and road in $[N]$, respectively on the rear axle and on the front axle.

Parameter	Symbol	Value	Unit
Front cornering stiffness	C_f	$20 \cdot 10^3$	N/rad
Rear cornering stiffness	C_r	$20 \cdot 10^3$	N/rad
Front load factor	η_f	1.35	-
Rear load factor	η_r	1.0	-

Table 3.2: Value of cornering stiffness and load factor, taken from [13].

The following linear (for constant v_x) model is considered for the lateral forces:

$$F_{yf} = -c_f \beta_f \quad (3.8a)$$

$$F_{yr} = -c_r \beta_r \quad (3.8b)$$

$$\beta_f = \frac{v_y + l_f \omega_\psi}{v_x} - \delta_f \quad (3.8c)$$

$$\beta_r = \frac{v_y - l_r \omega_\psi}{v_x} \quad (3.8d)$$

Note 3.2

Clearly, in real conditions v_x is not exactly constant, but during a single prediction horizon, which is in the order of magnitude of tens of ms , it can be considered constant. Moreover, as said before, it is not possible to have a perfect model, some approximation must be done.

Note 3.3

Since the value of the tire slip angles β are approximated (there is no $\arctan(\cdot)$) in the relation, the model is quite precise for small values of the steering angle: $\tan(\beta) \approx \beta$ for $\delta_f \rightarrow 0$ ($\delta_f < 20^\circ$, i.e. $\delta_f < 0.35 \text{ rad}$).

Note 3.4

If $\delta_f = 0$ for all the time, then we are on a straight road. In this scenario Y coordinate of the CoG, v_y and ψ will never change.

Note 3.5

When $v_x \rightarrow 0$ then $F_{yf}, F_{yr} \rightarrow \infty$. This explosion of values is related to the model: the Single-Track model assumes that the vehicle is not still, but in a quite high range of velocity. A safe limit of $v_{x,safe} = 5 \text{ m/s}$ is used to avoid blow up of the lateral forces' value in the simulation.

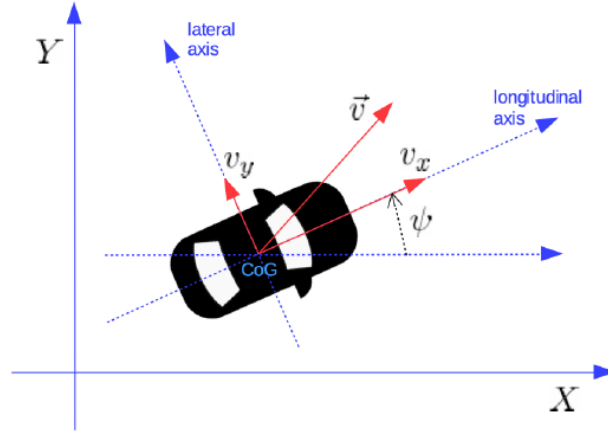


Figure 3.2: Scheme of the reference frames in the Lateral Dynamics Model, from [13].

3.3 Gearbox Model

As said, the motor can be seen as a torque generator, but for practical reasons (cost of production, dimension of the motor itself, weights etc.), in general a motor has small dimension and *small inertia*. In this way it can provide a small torque, but it can run at very high speed. Unfortunately, those are not the characteristics that we need. Therefore, an additional component is necessary to extract a *high* torque value from the motor: the vehicle's gearbox.

The gearbox has two aims:

- Speed matching: since the motor runs at high velocity, that is not comparable with a normal speed of a car, we need to reduce the rotational speed of the motor (ω_{EM}) before it reaches the wheels (ω_w), ensuring the most efficient power transfer.
- Torque amplification: since the small inertia of the motor cannot provide the high torque needed to a vehicle to move at high speed (for example in highways) or win the opposition of the force in uphill situation, the low-torque output of the motor (T_{EM}) must be enlarged before it reaches the wheels (T_w).

These two aims are reached using the gearbox, which has two parameters of in-

terest:

- The gearbox ratio τ_{gb} : it is a multiplication factor, by which is enlarged the value of the output torque of the motor, and by which is reduced the value of the rotational speed of the EM. This opposite behaviour is clearly linked to the principle of energy conservation.
- The gearbox efficiency or transmission efficiency η_{tr} : in any case a loss is present in the drivetrain, and it plays an important role in the power management of the total system.

In the end, the relationships between the EM angular velocity ω_{EM} and torque T_{EM} , and the wheel angular velocity ω_w and the wheel torque T_w are:

$$\omega_w = \frac{\omega_{EM}}{\tau_{gb}} \quad (3.9)$$

$$T_w = T_{EM} \cdot \tau_{gb} \cdot \eta_{tr}^{\text{sign}(T_{EM})} \quad (3.10)$$

In (3.10) must be taken into account both the acceleration and the braking condition: in the first case the EM that provides a torque to the system, as a result of losses, a slightly lower torque reaches the wheels. In the other way around, i.e. braking condition, the braking torque transmitted from the wheels to the motor will be slightly lower.

3.4 Wheel Model

The wheels of a vehicle are not perfectly rigid; they undergo deformation, which, as said, affects the vehicle's motion. This influence involves many factors, but to make the system manageable they are not considered except for the Rolling Resistance Force through the coefficient f_r . Therefore, the radius of the wheels is fixed, and it is a simple scaling factor between rotational quantities and linear quantities, in particular:

$$F_w = \frac{T_w}{r} \quad (3.11)$$

$$v_x = \omega_w \cdot r \quad (3.12)$$

In the end, replacing (3.10) and (3.9) respectively in (3.11) and (3.12):

$$F_{trac} = F_w = \frac{T_{EM} \cdot \tau_{gb} \cdot \eta_{tr}^{\text{sign}(T_{EM})}}{r} \quad (3.13)$$

$$v_x = \frac{\omega_{EM}}{\tau_{gb}} \cdot r \quad (3.14)$$

3.5 Electric Motor Model

The power provided by the EM is given by a Torque component (T_{EM}) and a rotational speed component (ω_{EM}):

$$P_{EM} = T_{EM} \cdot \omega_{EM} \quad (3.15)$$

The motor can provide this power thanks to the battery, the reserve of energy of the vehicle, but the power provided by the battery to the motor (P_b) does not coincide with P_{EM} : the efficiency of the EM and the efficiency of the Inverter must be taken into account.

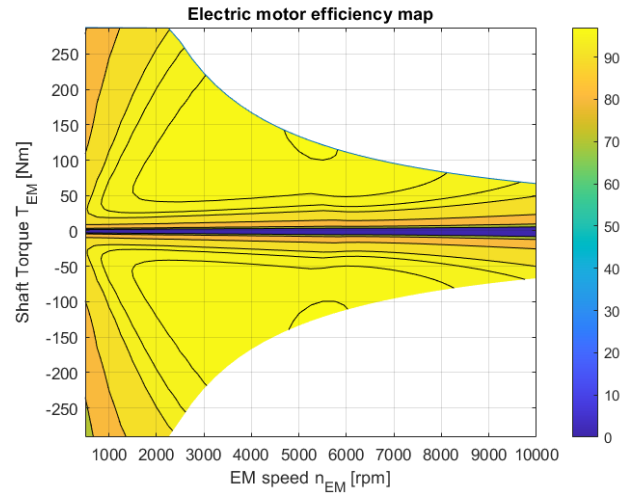
$$P_b = \frac{P_{EM}}{[\eta_{EM}(\omega_{EM}, T_{EM}) \cdot \eta_{inv}]^{\text{sign}(P_{EM})}} \quad (3.16)$$

where:

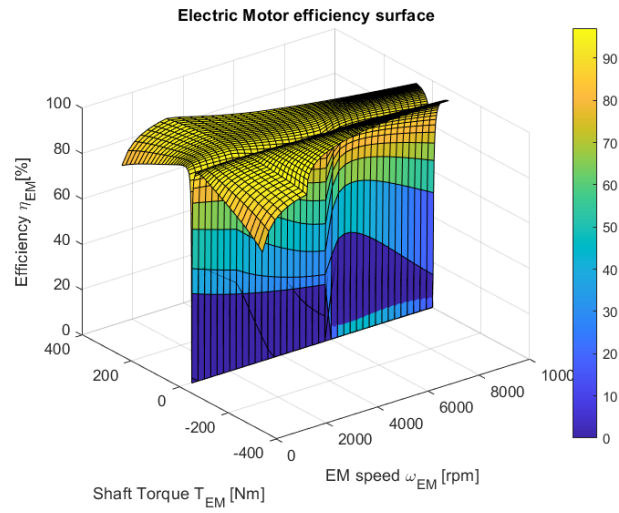
- $\eta_{EM}(\omega_{EM}, T_{EM})$ is the efficiency of the EM, which depends on the value of the angular velocity and the torque of the motor.
- η_{inv} is the efficiency of the inverter.

As explained in Section 3.3, the efficiencies have differing impacts during braking and acceleration modes, it is again highlighted in (3.16).

Regarding the value of $\eta_{EM}(\omega_{EM}, T_{EM})$ it is computed from the efficiency map of the motor, which is shown in Figure 3.3.



(a) 2D representation of the efficiency map of the EM.



(b) 3D representation of the efficiency map of the EM.

Figure 3.3: Efficiency map of the EM.

3.6 Battery Model

The model for the battery dynamics is taken from [4].

The battery model takes as input the power requested by the motor P_{EM} to satisfy the torque request T_{EM} and the efficiency of the motor $\eta_{EM}(n_{EM}, T_{EM})$, then returns as output the power generated by the battery P_b^{gen} and the SOC at the current time instant. It's important to notice that $P_b \neq P_b^{gen}$.

As detailed in [4], the SOC is defined as the ratio between the battery charge Q_b and the nominal battery capacity Q_{nom} :

$$SOC = \zeta = \frac{Q_b}{Q_{nom}} \in [0, 1] \quad (3.17)$$

The dynamic equation that represents the SOC rate with the battery current is obtained by differentiation and it includes the physical quantities of interested:

$$\dot{\zeta} = -\frac{\eta_{b,1}(P_b) \cdot I_b}{Q_{nom}} \quad (3.18)$$

where:

- P_b is the power requested at the battery taking into account the motor efficiency η_{EM} and the inverter efficiency η_{inv} , as detailed in Section 3.5.
- $\eta_{b,1}(P_b)$ is the Coulumbic efficiency, it quantifies a fraction of the current that, unfortunately, is lost during both battery discharge and battery charge.

$$\eta_{b,1} = \begin{cases} \frac{1}{\eta_b^{coul}} & \text{if } P_b > 0 \text{ discharge phase.} \\ \eta_b^{coul} & \text{if } P_b < 0 \text{ charge phase.} \end{cases} \quad (3.19)$$

Normally, a battery pack is made by N_{ser} cells connected in series and N_{par} cells connected in parallel, then we have a total of $N_b = \frac{N_{ser}}{N_{par}}$ connected in series in the total battery.

Each cells is modelled as an ideal voltage source $V_b^{oc,s}$ with a series output resistance $R_b^{o,s}$, then the total voltage and the total resistance of the battery will be: $V_b^{oc} = N_b V_b^{oc,s}$ and $R_b^o = N_b R_b^{o,s}$. In real-world batteries the values of V_b^{oc} and R_b^o

depend on SOC . In order to make the model as realistic as possible the value of $V_b^{oc}(\zeta)$ and $R_b^o(\zeta)$ will be computed by polynomial fitting of a set of experimental data, taken from [14]. The interpolation is made by a third order polynomial and it is used to evaluate the voltage and the resistance at each time instant based on the value $\zeta(k)$. The resulting interpolated behaviour is shown in Figure 3.4.

Taking into account the Joule losses the power generated by the battery is:

$$P_b^{gen} = V_b^{oc} I_b - R_b^o I_b^2 \quad (3.20)$$

Then, the actual power delivered from the generated power must take into account also the losses due to the power converter, which interfacing the battery with the DC bus, i.e. the efficiency of the power converters $\eta_{b,2}$:

$$\eta_{b,2} = \begin{cases} \frac{1}{\eta_b^{conv}} & \text{if } P_b > 0 \text{ discharge phase.} \\ \eta_b^{conv} & \text{if } P_b < 0 \text{ charge phase.} \end{cases} \quad (3.21)$$

The power generated can be also expressed as:

$$P_b^{gen} = \eta_{b,2} \cdot P_b \quad (3.22)$$

By comparing (3.22) and (3.20), and solving for the battery current, we obtain:

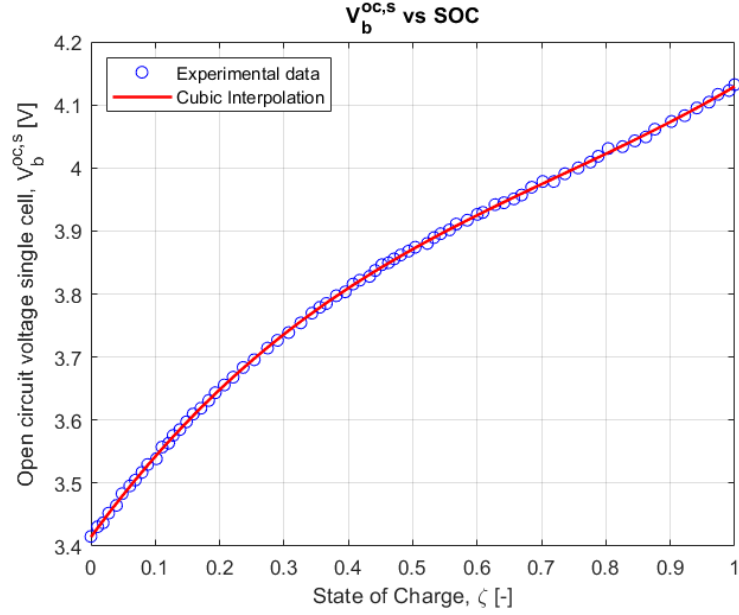
$$\begin{aligned} I_b &= \frac{1}{2R_b^o(\zeta)} \left(V_b^{oc}(\zeta) - \sqrt{V_b^{oc^2}(\zeta) - 4R_b^o(\zeta) \cdot P_b^{gen}} \right) = \\ &= \frac{1}{2R_b^o(\zeta)} \left(V_b^{oc}(\zeta) - \sqrt{V_b^{oc^2}(\zeta) - 4R_b^o(\zeta) \cdot \eta_{b,2} \cdot P_b} \right) \end{aligned} \quad (3.23)$$

The dynamical model of the battery's SOC is:

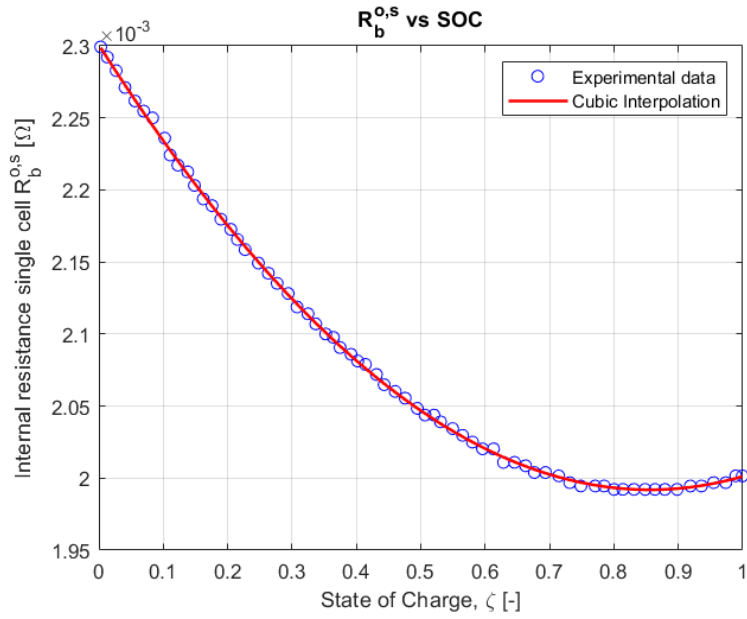
$$\dot{\zeta} = -\frac{\eta_{b,1}(P_b)}{Q_{nom}} \cdot \frac{1}{2R_b^o(\zeta)} \left(V_b^{oc}(\zeta) - \sqrt{V_b^{oc^2}(\zeta) - 4R_b^o(\zeta) \eta_{b,2} \cdot P_b} \right) \quad (3.24)$$

Note 3.6

The dynamical model of the battery assumes the form: $\dot{\zeta} = f_{c,b}(\zeta, P_b)$, where P_b is linked to P_{EM} as expressed in (3.16).



(a) Open circuit voltage for single cell.



(b) Internal resistance for single cell.

Figure 3.4: Experimental Data on Voltage and Resistance for a Single Cell in the Battery Pack, taken from[14].

According to [4], the model (3.24) forces an upper bound on the value of P_b , that is:

$$P_b \leq \eta_b^{conv} \frac{V_b^{oc}(\zeta)^2}{4R_b^o(\zeta)} = \bar{P}_b \quad (3.25)$$

This upper bound still depends on ζ , a complexity difficult to manage. Typically, an upper bound independent of ζ is chosen to simplify the model, and to be sure that in each situation the power request is feasible, i.e. the battery can provide it, this bound is chosen as:

$$\bar{P}_b^{spec} < \min_{\zeta \in [\underline{\zeta}, \bar{\zeta}]} \bar{P}_b \quad (3.26)$$

where $\underline{\zeta}$ and $\bar{\zeta}$ are the minimum and maximum value of the SOC admissible.

For simplicity the lower and upper bound of P_b are chosen as:

$$P_b^{up} = \min_{\zeta \in [\underline{\zeta}, \bar{\zeta}]} \bar{P}_b \quad \text{and} \quad P_b^{low} = -P_b^{up} \quad (3.27)$$

in the simulations of Chapter 5.

The Battery Nominal parameters are listed in Table 3.3.

Parameter	Symbol	Value	Unit
Number of cells in series	N_{ser}	108	-
Number of cells in parallel	N_{par}	1	-
Total number of battery cells connected in series	$N_b = \frac{N_{ser}}{N_{par}}$	108	-
Nominal battery capacity	Q_{nom}	60	Ah
Coulombic efficiency of the battery	η_b^{coul}	0.95	-
Conversion efficiency of the battery	η_b^{conv}	0.97	-
Inverter efficiency	η_{inv}	0.97	-
Upper bound on the battery power	P_b^{up}	132.71	kW
Lower bound on the battery power	P_b^{low}	-132.71	kW

Table 3.3: Battery Nominal parameters.

3.7 State Space Representation

The overall dynamics of the system can be described by a nonlinear function that characterizes its time evolution:

$$\dot{\mathbf{x}} = f(\mathbf{u}, \mathbf{x}) \quad (3.28)$$

From the general expression in (3.28) a *State Space Representation* of the model can be retrieved. For the State Space Representation, the following seven quantities are chosen as system states:

$$\mathbf{x} = \begin{bmatrix} X \\ Y \\ \psi \\ v_x \\ v_y \\ \omega_\psi \\ \zeta \end{bmatrix} \quad (3.29)$$

Each component in the vector state $\mathbf{x}$ has a clear physical meaning:

- $x_1 = X$ is the position along the x -axis of CoG of the vehicle, measured in $[m]$.
- $x_2 = Y$ is the position along the y -axis of CoG of the vehicle, measured in $[m]$.
- $x_3 = \psi$ is the heading of the vehicle, i.e. it is the angle between the longitudinal axis of the vehicle and the x -axis, positive in counter-clockwise direction, measured in $[rad]$.
- $x_4 = v_x$ is the longitudinal velocity of the vehicle, measured in $[m/s]$.
- $x_5 = v_y$ is the lateral velocity of the vehicle, measured in $[m/s]$.
- $x_6 = \omega_\psi$ is the yaw rate of the vehicle, measured in $[rad/s]$.
- $x_7 = \zeta$ is the SOC of the battery, which is dimensionless.

The system's behaviour is influenced by two *control inputs*, which are the quantities that the controller can change in order to guide the vehicle in the desired trajectory:

$$\mathbf{u} = \begin{bmatrix} T_{EM} \\ \delta_f \end{bmatrix} \quad (3.30)$$

Also the control inputs have a physical meaning:

- $u_1 = T_{EM}$ is the torque provided by the EM, measured in $[Nm]$, to the vehicle. T_{EM} is the value before the enhancement, and the consequently losses, of the gearbox. In simple terms, is the input that directly influences the vehicle's longitudinal acceleration and, consequently, its longitudinal speed.
- $u_2 = \delta_f$ is the Steering Angle of the vehicle, measured in $[rad]$. It is the control input associated with the vehicle's heading and enables it to stay on track.

The state equations are derived from the Lateral Model in Section 3.2.2, where the equations from (3.7a) to (3.7f) represents the first six states.

Moreover, the expression of the longitudinal acceleration in (3.7d) is taken from the Longitudinal Model detailed in Section 3.2.1 leading to a final expression of the kind $a_x = f(T_{EM}, m)$:

$$a_x = \frac{T_{EM} \frac{\tau_{gb} \eta_{tr}}{r} - (F_{aero} + F_{slope} + F_{roll})}{m} \quad (3.31)$$

in (3.31) the extended expression of $F_{aero}, F_{slope}, F_{roll}$ are not shown to do not make heavy the expression and to highlight the relationship between a_x and the control input u_1 . Regarding the second input u_2 , in (3.8a) is shown how the lateral force on the frontal axle depends on the Steering Angle: $F_{yf}(u_2)$.

Regarding the last state, it can be shown that is related to the first control input. From (3.24) it's clear that the rate of the State of Charge, $\dot{\zeta}$, is a function of the State of Charge, ζ , and the battery power, P_b . Then the expression of the battery current is directly related to the EM Torque, that is u_1 :

$$P_b = \frac{P_{EM}}{[\eta_{EM}(\omega_{EM}, \mathbf{u}_1) \cdot \eta_{inv}]^{\text{sign}(P_{EM})}} = \frac{\mathbf{u}_1 \cdot \omega_{EM}}{[\eta_{EM}(\omega_{EM}, \mathbf{u}_1) \cdot \eta_{inv}]^{\text{sign}(\mathbf{u}_1 \cdot \omega_{EM})}} \quad (3.32)$$

Hence, $P_b(u_1)$.

In the end, the total State Space Representation is:

$$\dot{\mathbf{x}} = \begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \\ \dot{x}_4 \\ \dot{x}_5 \\ \dot{x}_6 \\ \dot{x}_7 \end{bmatrix} = \begin{bmatrix} x_4 \cos(x_3) - x_5 \sin(x_3) \\ x_4 \sin(x_3) + x_5 \cos(x_3) \\ x_6 \\ x_5 x_6 + \frac{\mathbf{u}_1 \tau_{gb} \eta_{lr}}{r} - \frac{(F_{aero} + F_{slope} + F_{roll})}{m} \\ -x_4 x_6 + \frac{2}{m} (F_{yf}(\mathbf{u}_2) + F_{yr}) \\ \frac{2}{J} (l_f \cdot F_{yf}(u_2) - l_r \cdot F_{yr}) \\ - \frac{\eta_{b,1}(P_b(u_1)) \cdot I_b(x_7, u_1)}{Q_{nom}} \end{bmatrix} \quad (3.33)$$

To do not make too heavy the expression not all the quantities are expressed in details explicitly. Nevertheless, it can be appreciated how the control inputs affect the states.

Chapter 4

NMPC and EMPC Framework

Most control strategies simply provide a control action in order to react to current errors. The typical controllers, such as PID or pole placements, use a *fixed* control law, which is computed a priori. The MPC is different: it actively foresees the system behaviour and plans an *optimal* control action, over a prediction horizon. Its major characteristic is the possibility to *choose* the best possible control action among a span of possibilities.

Unfortunately, this great flexibility and the powerful optimization method, generate a more challenging study of the asymptotic stability with respect to simpler method of control. However, under certain conditions, the MPC can guarantee asymptotic stability.

As the name of the approach specifies, the method of control is strictly related with the *model* of the physical phenomenon: we cannot predict the evolution of the system if the model is not accurate, or worst not available. Nowadays, the macroscopic mechanics and dynamics are detailed and well structured: the models are deterministic, so we can use them for a predictive approach.

In general, a control system can be configured as either an *open-loop* or a *closed-loop* scheme. For simple systems an open-loop configuration is sufficient to control the plant. However, the dynamics of a vehicle in operation represents a complex system, for which a closed-loop control strategy is required.

In control theory, the implementation of a closed-loop system is intended to counteract potential disturbances and compensate for model inaccuracies. The

loop closure in a MPC approach slightly differs from other control strategies, as it is based on the *receding horizon principle*. Basically, the receding horizon principle uses a moving time window during the operation to be sure of catching the actual evolution of the system instead of the evolution described by the mathematical-physical model, which may lack disturbance modelling or may neglect certain dynamical phenomena.

The basic idea of the receding horizon principle, as expressed in [17] is the following: at the time step k the system is in state x_k , then the optimization starts. Using the initial state x_k we can predict the following N_p steps, using an optimization criterion to choose the best possible evolution of the system over the prediction horizon. Once the optimal input sequence u^* is identified, corresponding to the states sequence x^* that satisfies both the performances and constraints, we apply the first control action of the sequence: $u^*(1)$. The remaining control actions $u^*(2), \dots, u^*(N_p)$ are discarded. Applying only $u^*(1)$, the plant evolves to the state x_{k+1} . As the system evolves to the next step $k+1$, new measurements provide update informations on the system, which are used as initial conditions for the successive prediction window, and the iterative procedure starts again according to the receding horizon principle.

By executing only the first input of the optimal sequence and starting a new optimization with updated informations, MPC effectively establishes a closed-loop structure, enabling the controller to adapt to system changes, disturbances and modelling errors. Otherwise, if we apply the whole sequence u^* may happen, and of course it does, that the system evolution differs from the predicted behaviour, leading to a situation in which $x_{actual}(k+i) \neq x^*(k+i)$ due to the unmodelled phenomena.

We can see as the receding horizon principle applied to a MPC approach generates a controller which combines a prediction-based planning of the control action with a continuous update and correction of the prediction based on the 'feedback' response.

The MPC controllers have several variants of algorithm. In this work we will deal with a high nonlinear phenomena, so the basic Linear MPC is not effective, we need to approach the problem with a Nonlinear MPC, and then an Economic approach of the NMPC. The following chapter introduces the general NMPC for-

mulation of a problem, outlines its characteristics and highlights the differences compared to an Economic NMPC approach.

Note 4.1

From now on, we always consider *discrete-time systems*. Controllers, and in general computers, always work in discrete time. Even if the steps are small, we must be able to deal with a discrete time representation of a physical system.

4.1 General MPC formulation

For MPC the primary applications are stabilization and tracking problems. In tracking, the task is to determine a control action u such that the states of the system x follow a given reference x_{ref} as close as possible. The reference can be constant over time or can change over time.

Consider the discrete-time system:

$$\mathbf{x}_{k+1} = f(\mathbf{x}_k, \mathbf{u}_k, \mathbf{d}_k) \quad (4.1)$$

where:

- $f(\cdot)$ is a nonlinear function.
- $\mathbf{x} \in \mathbb{R}^n$ is the state vector of the system.
- $\mathbf{u} \in \mathbb{R}^m$ is the vector of inputs of the system.
- $\mathbf{d} \in \mathbb{R}^p$ is the disturbance vector of the system.

In the following work the disturbances are not considered, situations as a sudden burst of wind are not modelled in the State Space Representation of Chapter 3; therefore, the dependence on $\mathbf{d}$ is ignored. Of course they occur in nature, then disturbances can influence the plant; however, the implementation of the controller is not thought for applications in difficult conditions of driving, then we can neglected the major disturbances. Anyway, in a real world application, the disturbances must be considered.

This simplification leads to the following form of the prediction model in discrete-time:

$$\mathbf{x}_{k+1} = f(\mathbf{x}_k, \mathbf{u}_k) \quad (4.2)$$

Assumption 4.1

The generic nonlinear function $f(\cdot)$ is locally continuous and differentiable.

Assumption 4.1 does not hold for the expressions of the State Space Representation presented in Section 3.7, but with some mathematical trick during the implementation of the controller the assumption can be effectively enforced, as we will see in Section 5.4.

Note 4.2

The assumption 4.1 can seem restrictive, but are rare the cases in which the nature is discontinuous and however some practical adjustments can be used to avoid the discontinuity.

4.2 Tracking NMPC

The traditional framework typically prioritizes regulation or reference tracking.

- **Regulation:** In general, regulation refers to a control system whose objective is to drive the system output towards a pre-defined value following a perturbation. The pre-defined value in most of the cases is zero or a constant value.
- **Reference Tracking:** The tracking, instead, refers to a control system whose aim is to track a reference value as faithfully as possible. The reference value can be constant, but the case in which it varies over time is more interesting.

The regulation is not really interested for our particular case, so we will concentrate on the reference tracking case.

The criterion used to drive the controller is a cost function, in which the different terms penalize the divergence from the desired value. It can happen that

the trajectory of the states, or some of them, is exactly equal to the desired value at some time step k . This lucky case is taken into account in the formulation of the cost function, then we want to have a mathematical form that can penalize the deviations from the desired value and cut out the penalty if the trajectory is exactly what we want to reach.

The cost function that can satisfy these two conditions often take the forms of a quadratic relation, convex if possible, in which the difference between the reference value and the actual value of states or control input of interest guides the penalties: if the difference is not zero the penalty is present in the cost function, if the difference is not present, i.e. the variable is exactly equal to the desired value, the penalty is removed.

Then the algorithm will minimize this cost function at each step k to find an optimal solution for the situation.

A general example of a stage cost for tracking can be the following:

$$\ell(x, u) = \sum_{k_p=0}^{N_p-1} (\|x_{k_p} - x_{ref}\|_Q^2 + \|u_{k_p} - u_{ref}\|_R^2) \quad (4.3)$$

where Q, R are positive diagonal matrices, containing suitable weights, that penalize deviations from reference states and inputs, respectively. The behaviour of the control system that uses (4.3) depends on weighting matrices Q and R :

- Choosing $q_{ii} \gg r_{ii}$ results in aggressive control with the objective of *fast* tracking. If we have a fast tracking of the reference, the value of the control input can assume values that differ significantly from the reference, or the states may exhibit significant oscillations in order to reach the reference as quickly as possible.
- Choosing $r_{ii} \gg q_{ii}$ results in *slow* tracking of the reference of the states, in fact we want to be closer to a reference behaviour of the control input. More time is needed for reaching the desired configuration, but smooth behaviours of the physical quantities are often detected.

Note 4.3

The inclusion of a term that tracks the reference input in (4.3) indicates a desire to impose a reference behaviour on the input as well. However, this is not a mandatory procedure for tracking. Alternatively, limits on the input effort can be imposed as constraints of the optimization problem.

The asymptotic stability of this type of algorithm is difficult to study, but some tools can be used to force stability or, at least, to push the algorithm in the right direction. In particular **stabilizing terminal conditions** are effective for this purpose, the two presented in the following sections are:

- **Terminal cost**, detailed in Section 4.5.
- **Equilibrium endpoint constraint or terminal constraint**, detailed in Section 4.6.1.

4.3 Economic Nonlinear MPC Formulation

Economic Nonlinear MPC is a specialized strategy within the broader framework of NMPC. The main idea of a NMPC is to penalize the distance from a reference or a pre-defined equilibrium using a suitable cost function as we have seen in Section 4.1, whereas the economic approach is based on another principle. Since we can formulate the expression of the cost function to address the changing of all kind of quantities, we can use it to approach the minimization of energy consumption, instead of focusing only on the tracking of a reference, such a viewpoint characterizes the Economic NMPC framework.

We do not abandon the concept of reference tracking or the objective of driving the system towards an equilibrium. This remains a point of paramount importance in controller design: a controller that fails to achieve reference behaviour or desired performance would be meaningless. But now we aim to exploit the interplay between system dynamics and an alternative minimization approach to drive the system towards the reference trajectory, without prioritizing its tracking from the outset. Basically we want to follow the reference as an implicit consequence of our new point of view on the cost function.

Based on this idea, we can formulate the general cost function of Economic NMPC as follows:

$$J(x, u) = \left(\sum_{k_p=0}^{N_p-1} \ell_o(x_{k_p}, u_{k_p}) \right) + V_o(x_{N_p}) \quad (4.4)$$

where:

- $\ell_o(x_{k_p}, u_{k_p})$ is the economic stage, it expresses the quantity we are going to minimize as a function of the state x_{k_p} and the input u_{k_p} .
- $V_o(x_{N_p})$ is an offset cost, it is related to system stability. It will be the terminal cost.

4.4 Optimization variables

In an optimization problem, the *optimization variables* are those variables that can be manipulated in order to determine the optimal solution. Basically, they are the parameters that we can change for moving inside the space of admissible solutions in order to find the optimal or suboptimal one. When solving a prediction model, there are two main approaches for selecting the optimization variables:

- **Explicit prediction form:** we arrange the prediction model in order to have an explicit relation between the states, the inputs and the initial conditions of the states. In this way the future states are explicitly expressed in function of the past inputs and the initial state:

$$\mathbf{x} = \Phi \mathbf{x}_0 + \Gamma \mathbf{u} \quad (4.5)$$

This method is suitable when dealing with a linear model: matrices Φ and Γ depend on the linear representation of the model. Although any model can be linearized, this process may introduce significant difficulties in accurately catching the actual behaviour of the underlying phenomena. Moreover, in this approach, the problem is strongly simplified: the optimization variables are often only the control inputs $\mathbf{u}$, while the states are computed

as a linear combination of the initial states and the possible control inputs. For a simple model, which can be linearized with a satisfying accuracy, and for limited computational resources, it is an effective approach.

- **Implicit prediction form:** for complex system, for which the linearization is not a reliable option, the implicit prediction form is employed: there is not an explicit expression for computing the states from the inputs, but the model relates states $\mathbf{x}$ and inputs $\mathbf{u}$ in an implicit form, as seen in (4.2). In this approach, both states and control inputs are optimization variables. Their evolution is dynamically constrained by the implicit expression of the model and not just computed from ones others. The problem gains complexity from the point of view of the constraints, but this form provides greater flexibility and additional degrees of freedom in the search for an optimal or suboptimal solution.

In the following work, and in general, the implicit prediction form is the approach adopted in the development of the MPC controller. It means that the states: coordinates of position of CoG of the vehicle, heading, longitudinal and lateral speed, yaw rate and SOC, are treated as optimization variables in addition to the inputs and the dynamic of the system is incorporated as equality constraints in the optimization problem.

4.5 Terminal Cost

The best case of a predictive controller is given when we can predict all the future, i.e., the prediction horizon goes from $k_p = 0$ to $k_p = \infty$. Clearly, predicting too far into the future becomes impractical. Moreover, in the framework of the receding horizon principle, such long-term predictions are essentially useless. Thus, the cost function is evaluated on a *finite*-horizon, which becomes the prediction horizon.

In order to have a satisfying approximation of the infinite horizon with a finite number of steps, we can add the *terminal cost*. The strategy of adding the terminal cost is referred to as a *quasi-infinite* horizon, as explained in [10]. In general, the terminal cost can approximate the behaviour of the cost function *beyond* the

prediction horizon. To have this approximation, the weight P of the terminal cost is chosen as solution of the Riccati Equation.

In our case, the terminal cost is always a stabilizing term, but its weight is chosen after a trial-and-error procedure.

The presence of the terminal cost penalizes the states far from the stable equilibrium or the reference value, enforcing the evolution of the system towards a stable area. A terminal cost must be added in (4.3):

$$J(x, u) = \left(\sum_{k_p=0}^{N_p-1} \ell(x_{k_p}, u_{k_p}) \right) + V(x_{N_p}) \quad (4.6)$$

where $\ell(x_{k_p}, u_{k_p})$ is the stage cost, as expressed in (4.3), and $V(x_{N_p})$ is the terminal cost. A simple example of terminal cost can be:

$$V(x_{N_p}) = \|x_{N_p} - x_{ref}\|_P^2 \quad (4.7)$$

where P is the matrix of the weights of the elements of $V(x_{N_p})$.

Note 4.4

In [10] can be found a more detailed explanation of the computation of the terminal cost, through a linear quadratic approach. However, this method is computationally demanding and deviates from the goal of designing a controller with an intuitive formulation. Therefore, it is preferable to use a simpler idea of the terminal cost, that can engage a real situation as we are approaching to address.

4.6 Constraints

As said, an important feature of the MPC is the capability to deal with constrained optimization problems: it is well known that all real-world control systems have an associated set of constraints. A simple control design approach can proceed ignoring these constraints, but according to Goodwin et al. [17] is generally true that higher levels of performance are associated with the constraint boundaries,

then it becomes essential to explicitly account for them in the control design.

In [17] are expressed four main methods to deal with constraints in control system design:

- **Cautious**

In the cautious approach the performance requirements are relaxed until all the constraints are met in any situation, it is a conservative approach that allows the use of standard unconstrained strategies for the design. The approach can be carried out with a rigorous and linear analysis of the problem, but the best performances are never reached in order to satisfy the constraints even if they are not an insurmountable obstacle.

- **Serendipitous**

In the serendipitous approach, occasionally violation of the constraints are allowed. Once again, they do not address the constraints directly: no special precautions to handle the boundaries are implemented. In some cases, system performance remains acceptable even in the presence of constraint violations, but reaching high performances without facing directly the constraints can have a negative effect on the stability of the system, a risk that is not always acceptable.

- **Evolutionary**

In the evolutionary approach, as the name suggests, there is an evolution of the controller during the design. The first part of the design is made with an unconstrained approach to the problem, then, in the second part, the designer adds modifications and embellishments to avoid the negative consequences of the constraints on the performances through an iteratively trial-and-error procedure. In this second part, the aim is to avoid appearance of negative performances directly related to constraints. For example, anti-windup methods are among the modifications introduced at this stage. But, it can happen that the modifications cannot avoid the negative consequences of the constraints.

- **Tactical**

The tactical approach is the MPC approach: the constraints are incorpo-

rated in the formulation of the problem, and the controller is thought to deal with them from the beginning. The most used method to implement this approach is to set the problem as a constrained optimization problem. This method allows for effective control over the behaviour of the plant, even though the solution to the optimization problem is always a trade-off between performance and constraint satisfaction. Although, solving a constrained optimization problem is not an easy task, it is the most effective approach.

The tactical approach is used in the following work.

The constraints can be added at the optimization problem in two forms:

- **Equality constraints.**
- **Inequality constraints.**

Equality constraints

An equality constraint enforces an *exact* relationship, they are typically expressed as:

$$c(\mathbf{x}_k, \mathbf{u}_k) = 0 \quad (4.8)$$

The function $c(\cdot)$ can be complex, but, in general, it is a common practice to express the equality constraints in a matrix form, so it is a linear combination of states and inputs. In most of the cases, the equality constraints are used to impose a quantity equal to a reference $c_{ref}(\mathbf{x}_{ref}, \mathbf{u}_{ref})$, in this case it is useful to express the constraints in the following form:

$$c(\mathbf{x}_k, \mathbf{u}_k) - c_{ref}(\mathbf{x}_{ref}, \mathbf{u}_{ref}) = 0 \quad (4.9)$$

This latter formulation is the most common in the solvers available, especially because they cannot handle directly an equality constraints. The practical implementation of this type of constraints include the usage of (4.9) in an inequality formulation in which lower and upper bound are equal to zero, i.e.:

$$0 \leq c(\mathbf{x}_k, \mathbf{u}_k) - c_{ref}(\mathbf{x}_{ref}, \mathbf{u}_{ref}) \leq 0 \quad (4.10)$$

The equality constraints are particularly useful for the equations describing the dynamical model: we must be sure that the system evolves according to the prediction model, so we express this need in the form of an equality constraint.

Note 4.5

Although the purpose of equality constraints is to enforce exact equivalence, they are often challenging to handle in practice during the resolution of an optimization problem, particularly from the numerical point of view. Thus, tolerance for respecting the equality constraints (and also inequality constraints) are used.

Inequality constraints

The exact relations are not the only type of constraints that we can impose: upper and lower boundaries are common, especially for actuators. In these cases, the inequality constraints are suitable for expressing the bounds, the typical form they take is the following:

$$c_x^{min} \leq c(\mathbf{x}_k) \leq c_x^{max} \quad (4.11)$$

$$c_u^{min} \leq c(\mathbf{u}_k) \leq c_u^{max} \quad (4.12)$$

Once again, this formulation can be easily expressed in matrix form, which can be handled by a solver.

The two mentioned classes of constraints can be rewritten as:

$$c_i^{min} \leq c_{ineq,i}(\mathbf{x}_k, \mathbf{u}_k) \leq c_i^{max} \quad i = 1, \dots, q \quad (4.13)$$

$$0 \leq c_{eq,j}(\mathbf{x}_k, \mathbf{u}_k) \leq 0 \quad j = 1, \dots, p \quad (4.14)$$

Then the matrix form is straightforward:

$$\begin{bmatrix} c_1^{min} \\ \vdots \\ c_q^{min} \\ 0 \\ \vdots \\ 0 \end{bmatrix} \leq \begin{bmatrix} c_{ineq,1}(\mathbf{x}_k, \mathbf{u}_k) \\ \vdots \\ c_{ineq,q}(\mathbf{x}_k, \mathbf{u}_k) \\ c_{eq,1}(\mathbf{x}_k, \mathbf{u}_k) \\ \vdots \\ c_{eq,p}(\mathbf{x}_k, \mathbf{u}_k) \end{bmatrix} \leq \begin{bmatrix} c_1^{max} \\ \vdots \\ c_q^{max} \\ 0 \\ \vdots \\ 0 \end{bmatrix} \quad (4.15)$$

4.6.1 Terminal Constraints

As said in Section 4.2, a method for forcing asymptotic stability is the utilization of the reasonable requirements of an *equilibrium endpoint constraint* or *terminal constraint*. As detailed in [10], a terminal constraint is a simple way of constructing a **stabilizing terminal condition**. The idea is straightforward: since the optimization problem can be constructed with suitable constraints that respect both physical and logical limits, it is also possible to incorporate stability-related constraints. If we want to converge to the value $x_{ref}(k)$ at the end of the prediction horizon, we can impose it as an equality constraint, in order to push the optimization problem in the "right" direction:

$$x_u(k+1, x_0) = f(x_u(k, x_0), u(k)) \quad (4.16)$$

where $x_u(\cdot)$ is the state derived applying the control action u , and $f(\cdot)$ is the non-linear prediction model. Practically, adding this constraint, we are optimizing only on trajectories of the states that, starting from the position x_0 , are able to converge to the reference, through an evolution compatible with the prediction model $f(\mathbf{x}, \mathbf{u})$, this is the recursive feasibility property (see Section 4.7.1). Clearly, the initial position x_0 must belong to the set of the feasible states, called *Domain of Attraction* (see Definition 4.1), otherwise all the constraints are meaningless.

Note 4.6

We are not exactly following the reasoning behind (4.16) in the controller implementation. We will not use the prediction model to find a suitable trajectories of the state, it will be too demanding. Instead, we assume that the trajectory of the leading vehicle is consistent with the prediction model, so we will impose a terminal constraint to converge to it.

Note 4.7

We are strictly related to the method of resolution of the optimization problem and, in most of the cases, it needs an *initial guess* of the solution. The initial guess becomes the starting point of the optimization algorithm; if it is too far from a feasible trajectory, we risk not finding a solution to the problem.

In Figure 4.1 we can see the practical effect of the terminal constraint.

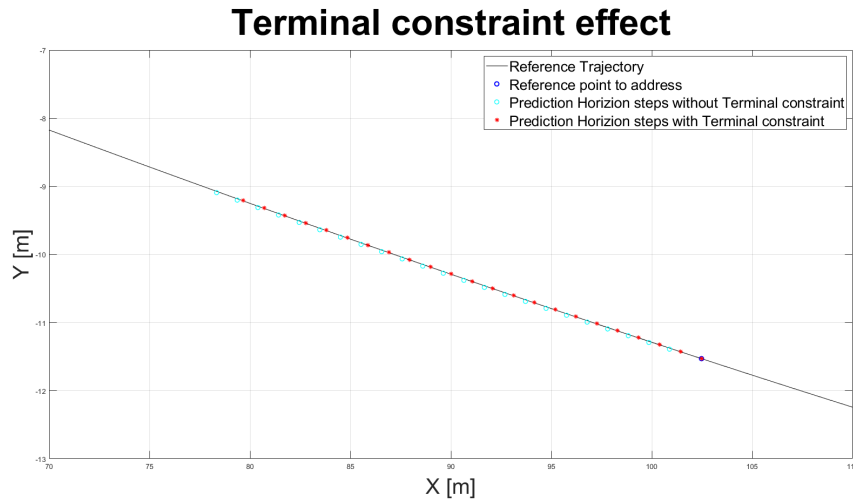


Figure 4.1: Terminal constraint effect.

We can see that the evolution of the state on the prediction horizon is forced to be closer to the reference trajectory, in order to try to respect the terminal constraint.

4.7 Recursive Feasibility and Asymptotic Stability

The recursive feasibility and the asymptotic stability are two main aspects in the MPC approach: the first one is related to the solvable characteristics of the problem, the second one ensures the approaching of the equilibrium point over time.

For both recursive feasibility and asymptotic stability, the concept of Domain of Attraction is important.

Definition 4.1 (Domain of Attraction) *The domain of attraction, or feasible region, for a tracking MPC controller is the set to which belongs all the initial states such that, starting from them the system can converge to the desired value, without violate the constraints.*

$$\mathcal{X}_{N_p}(x_{ref}) = \{x \in \mathcal{X} \mid \exists \mathbf{u} \in \mathcal{U}^{N_p} \text{ s.t. } x(k_p) \in \mathcal{X}, u(k_p) \in \mathcal{U}, k_p = 0, \dots, N_p - 1 \\ \text{and } x(N_p) = x_{ref}\}$$

where x_{ref} is the reference value to be achieved [9].

Therefore, the states that belong to the Domain of Attraction are the initial points for which the tracking problem is recursively feasible and asymptotically stable. Practically, if we start from a point of the domain of attraction, the problem can be solved. The same concept can be applied at an Economic MPC framework.

4.7.1 Recursive Feasibility

A feasible solution is a solution that respect all the constraints of the optimization problem. The problem of an iterative algorithm is the possibility to do not find a feasible solution over time, i.e., even if at the first time step a feasible solution exists, we are not sure that for all time steps a feasible solution can be found. As explained in [10], the MPC approach can face this problem: a trajectory towards the equilibrium point may include points that violate the constraints, i.e., unfeasible points. In this prospective, the recursive feasible property is fundamental for an iterative algorithm, like a NMPC controller. The recursive feasibility property ensures that, if the initial value of the state x belongs to a recursively feasible set, which is a subset of the feasible solution and it is invariant for the NMPC feedback

law, then the NMPC closed loop will generate an admissible solution for all future time steps [10]. Therefore, if the property is valid, we are sure that if we starts from a feasible solution for the first time step, in all the successive future time steps a feasible solution can be found. Practically, we are sure that the controller does not enter in a region in which is impossible to find an admissible solution.

Note 4.8

The recursive feasibility does not imply the asymptotic stability.

Theorem 4.1 (Recursive Feasibility of MPC) *Consider the following MPC problem:*

$$(\mathbf{x}^*, \mathbf{u}^*) = \arg \min_{x \in \mathcal{X}, u \in \mathcal{U}} J(x, u) \quad (4.17)$$

s.t.

$$x_{k_p+1} = f(x_{k_p}, u_{k_p}), k_p = 0, \dots, N_p - 1 \quad (4.18a)$$

$$x_0 = x, \quad (4.18b)$$

$$x_{k_p} \in \mathcal{X}, k_p = 0, \dots, N_p - 1 \quad (4.18c)$$

$$u_{k_p} \in \mathcal{U}, k_p = 0, \dots, N_p - 1 \quad (4.18d)$$

$$x_{N_p} \in \mathcal{X}_f \quad (4.18e)$$

where the terminal constraints set is a control invariant set, $\mathcal{X}_f = \{x^e\}$ corresponding to u^e . If

1. $x^e \in \mathcal{X}, u^e \in \mathcal{U}$ and, $\mathcal{X}, \mathcal{U}$ are compact,
2. The function $f(x, u)$ is continuous,
3. The optimization problem is feasible for the initial state x_0 .

Then the MPC optimization problem remains feasible for all subsequent time steps [10].

Note 4.9

The notation (x^e, u^e) indicates an equilibrium point.

4.7.2 Asymptotic Stability

In every control strategy the stability is a point of paramount important, if the algorithm is not stable we are not sure that in every situation it can handle the plant. The stability property for the classic MPC approach are studied through Lyapunov functions, but, as said in [9], for the Economic MPC approach this technique is not applicable, at least under the same condition of the classic MPC.

Then, for studying stability property of an Economic NMPC, the knowledge of the dissipativity of the system is necessary.

Definition 4.2 (Dissipativity and strictly Dissipativity) *A control system is dissipative with respect to a supply rate $s : \mathcal{X} \times \mathcal{U} \rightarrow \mathbb{R}$ if there exists a function $\lambda : \mathcal{X} \rightarrow \mathbb{R}$ such that:*

$$\lambda(f(x, u)) - \lambda(x) \leq s(x, u) \quad (4.19)$$

$\forall (x, u) \in \mathcal{Z} \subseteq \mathcal{X} \times \mathcal{U}$.

If in addition $\sigma : \mathcal{X} \rightarrow \mathbb{R}_{\geq 0}$ positive definite exists such that:

$$\lambda(f(x, u)) - \lambda(x) \leq -\sigma(x) + s(x, u) \quad (4.20)$$

Then the system is said to be strictly dissipative.[1]

The physical interpretation of Definition 4.2, according to [9], is that **a dissipative system cannot create energy** by itself, it can only dissipate it or store it. While a strictly dissipative system cannot produce energy, and, moreover, it dissipates some energy even if no energy is provided to the system.

Consequently, a system capable of dissipating energy can converge to a stable equilibrium; trivially, if no energy is supplied, the system will eventually deplete its energy reserve, making further evolution impossible and resulting in a zero-energy state.

A vehicle in movement, our study case, is a strictly dissipative system. Since we consider the opposition of the Drag Force, the system will dissipate energy even if no energy is provided to the system, i.e. even if the driver does not accelerate, the vehicle will gradually lose kinetic energy due to Aerodynamic Drag.

If we define the function $s(x, u)$ as the amount of energy supplied by u and x we can have:

$$s(x, u) = \ell_o(x, u) - \ell_o(x^e, u^e) \quad (4.21)$$

where x^e, u^e is an equilibrium point [10].

Given the definition of $s(x, u)$ we can search for the function λ . According to [1] and [9] exists a function $\lambda : \mathcal{X} \rightarrow \mathbb{R}$ that we can use for defining a *rotated cost function*:

$$L(x, u) = \ell_o(x, u) + \lambda(x) - \lambda(f(x, u)) \quad (4.22)$$

The rotated cost function has a particular characteristic: it reaches its minimum under the same constraints of $\ell_o(x, u)$, i.e. at the equilibrium point (x^e, u^e) , but it will behave as a Lyapunov function; therefore, the stability can be studied. Consequently we need to add also the rotated terminal cost at the optimization problem of the rotated cost function:

$$\tilde{V}_o(x) = V_o(x) + \lambda(x) - V_o(x^e) - \lambda(x^e) \quad (4.23)$$

As stated in [9], we have built an auxiliary optimization problem, which has a different cost function, but the same constraints. In the end, the original optimization problem and the auxiliary optimization problem have the same solution. Additionally the following theorem is valid.

Theorem 4.2 (Asymptotic Stability of EMPC with Terminal Constraints) *Consider the Economic MPC optimization problem [9] [10]:*

$$\min_{x \in \mathcal{X}, u \in \mathcal{U}} \left(\sum_{k_p=0}^{N_p-1} \ell_o(x(k_p), u(k_p)) \right) + V_o(x_{N_p}) \quad (4.24)$$

s.t.

$$x_{k_p+1} = f(x_{k_p}, u_{k_p}), k_p = 0, \dots, N_p - 1 \quad (4.25a)$$

$$x_0 = x, \quad (4.25b)$$

$$x_{k_p} \in \mathcal{X}, k_p = 0, \dots, N_p - 1 \quad (4.25c)$$

$$u_{k_p} \in \mathcal{U}, k_p = 0, \dots, N_p - 1 \quad (4.25d)$$

$$x_{N_p} \in \mathcal{X}_f \quad (4.25e)$$

Assume that

1. The system is strictly dissipative at the equilibrium point $(x^e, u^e) \in \mathcal{X} \times \mathcal{U}$, with a storage function $\lambda : \mathcal{X} \rightarrow \mathbb{R}$ bounded from below with $\lambda(x^e) = 0$.
2. The terminal region (which determines the terminal constraints equalities) $\mathcal{X}_f \subseteq \mathcal{X}$ is compact, and x^e is interior to $\mathcal{X}_f$.
3. (Basic Stability Assumption) There exists a control law $\mu_{N_p} : \mathcal{X}_f \rightarrow \mathcal{U}$ such that:

$$V_o(f(x, \mu_{N_p})) = V_o(x) - \ell_o(x, \mu_{N_p}) + \ell_o(x^e, u^e), \forall x \in \mathcal{X}_f \quad (4.26)$$

with $V_o(x^e) = 0$.

4. The optimization problem is recursively feasible.

Then, the equilibrium point x^e is asymptotically stable for the closed-loop system.

Both Theorem 4.1 and Theorem 4.2 are useful in the practical implementation, particularly in clarifying the types of constraints that must be incorporated into the optimization problem.

4.8 Design of Stabilizing Term

As said, the stability of a MPC-based controller is not easy to study, but some strategies can be employed to achieve it. We have already seen the terminal cost

and the terminal constraint, but a more interesting approach is detailed in [3]. Calogero et al.[3] propose a procedure to design a stabilizing term, the main objective of their idea is to ensure closed-loop stability of an EMPC approach, whereas the impact on the performance of the controller is minimal. Their constructive procedure is used in the following work to face the design of an additional term of the stage cost for a new controller, we name this third controller *Conflicting Objectives NMPC* in the following sections.

The procedure is based on three main steps:

1. Computation of an equilibrium trade-off point (x_s, u_s) of the system to be controlled. The equilibrium trade-off point is computed by means of a constrained optimization problem, which cost function is composed by two parts: a tracking/regulation part and an economic part,

$$\ell(x, u) = \ell_r(x, u) + \ell_e(x, u) \quad (4.27)$$

The aim of the MPC-based control is to steer the system towards this equilibrium trade-off point. Hopefully, in the neighbourhood of (x_s, u_s) , there exists another optimal point that achieves a lower value of the cost function.

2. Then a stabilizing term is inserted to ensure closed-loop stability of the system. This term is a positive definite function of the following kind:

$$\alpha(z) = a \cdot \chi(z - z_s), \quad \chi : \mathcal{Z} \rightarrow \mathbb{R}_{>0} \quad (4.28)$$

where the weight $a \in \mathbb{R}_{>0}$. $\mathcal{Z}$ is the manifold of the states and the inputs of the system. $\alpha(z)$ is designed to push the system to a stable area of the manifold $\mathcal{Z}$ when the conflict objectives of the controller leads the plant to instability, thus:

- high value of a indicates that the controller is far from the stable area, therefore a strong action is necessary to push the system to a stable region.
- Conversely, when a is low, the controller is already progressing towards a stable area, requiring minimal adjustments to support its tra-

jectory.

The main focus of the procedure detailed in [3] is how to compute the weight a . In order to have x_s as an asymptotically stable equilibrium point of the closed-loop system to be controlled, the weight a must satisfy:

$$a > \frac{\ell(x_s, u_s) - \ell(x, u)}{\chi(z - z_s)}, \forall (x, u) \in \mathcal{Z} \setminus \{(x_s, u_s)\} \quad (4.29)$$

3. Then an augmented cost function is used to compute the final optimal control action in the MPC-based approach:

$$\bar{\ell}(x, u) = \ell(x, u) + \alpha(x, u) \quad (4.30)$$

Remark 4.1 *The optimal value of a that minimizes the influence of α on the original stage cost ℓ is given by:*

$$a^* = \sup_{(x, u) \in \mathcal{Z} \setminus (x_s, u_s)} \frac{\ell(x_s, u_s) - \ell(x, u)}{\chi(z - z_s)} + \varepsilon \quad (4.31)$$

where $\varepsilon > 0$ is arbitrarily small. [3]

By means of the *maximization problem* expressed in Remark 4.1 the optimal value of the weight a can be computed.

Additionally, an interesting characteristic of a^* arises:

Remark 4.2 *Consider (4.31) and let $\beta(x, u) = \frac{\ell(x_s, u_s) - \ell(x, u)}{\chi(z - z_s)}$. If*

$$\ell(z) = \ell(z_s) + O(\chi(z - z_s)) \text{ as } z \rightarrow z_s \quad (4.32)$$

then a^ is finite.*[3]

In Remark 4.2 are expressed the condition to avoid blow up of the weight of the stabilizing term.

The procedure described above is quite particular: it involves *three* constrained optimization problems, two minimization problems and one maximization problem, to determine the control action. The computational demand of this novel

procedure is certainly higher compared to other MPC-based methods, but it ensures stability. However, for simple systems, the increased computational effort can be managed, and the stability guarantees are highly appreciated.

As previously mentioned, we will employ this procedure to design another controller for our problem, with a key difference compared to the method described in [3]. In the present work, the term derived from the three-steps procedure outlined above will be treated as a simple **additional term** of the stage cost. Although the stability features of this addition to the stage cost have yet to be demonstrated, it offers one way to apply this interesting procedure to a complex dynamics such as the longitudinal-lateral dynamics of a vehicle.

Chapter 5

Controller Implementation, Simulation Results

In the following, the controllers will be implemented: the first one with the tracking approach, the second one with the economic approach. The tracking NMPC is taken as reference: all the practical problem and the implementation issues are first of all solved on this controller, all solutions are then replicated in the other controllers, i.e. the EMPC controller. Clearly, the economic approach has some differences with the tracking approach, then some particular implementation issues are different with respect to the NMPC, and they are treated ad hoc.

Finally the three-steps procedure seen in Section 4.8 is employed, with some differences, to design a third controller. Once again, the implementation precautions are detailed.

5.1 NMPC Design

The design of the controller starts with the definition of the cost function of the optimization problem: it should balance tracking performances, passengers' safety and comfort. After the testing of NMPC controller the tracking part will be eliminated and replaced with an Economic part. Therefore, the design of the NMPC is fundamental to find possible issues or limits of the model or of the implementation method.

5.1.1 NMPC Cost Function

The cost function to be minimized consists of three main parts, along with two additional terms introduced for technical reasons.

- **Lane Keeping Assist**

The first part of the cost function regards the Lateral Dynamics of the system, so the Lane Keeping Assist part of the controller is:

$$J_{LKA} = \sum_{k_p=0}^{N_p-1} W_{ct} \cdot e_{ct}^2 + W_h \cdot e_h^2 \quad (5.1)$$

where:

- i) $e_{ct} = (Y_r - Y_a) \cos(\psi_r) - (X_r - X_a) \sin(\psi_r)$: **Cross-Track error**. It is the orthogonal projection of the distance on the direction of motion.
 - X_r, Y_r, ψ_r : reference pose. For tracking, a reference is required; during development, the pose serves as that reference (X,Y coordinates in a 2D reference frame and the heading of the vehicle) of the front axle of a *leading vehicle*.
 - X_a, Y_a, ψ : actual front axle pose of the controlled vehicle. It is important to translate the actual pose of the CoG of the vehicle in the pose of the front axle of the vehicle in order to be compliant with the meaning of the Cross-Track error and its definition. The geometric relationships between the front axle and the CoG position of the vehicle are detailed in (5.2).

$$X_a = X + l_f \cos(\psi), \quad Y_a = Y + l_f \sin(\psi) \quad (5.2)$$

For convention:

- $e_{ct} > 0$: The vehicle is on the left with respect to the reference trajectory $\implies \delta_f < 0$ as control action in order to approach the reference trajectory (the steering wheel turns to the right from the driver's point of view).

- $e_{ct} < 0$: The vehicle is on the right with respect to the reference trajectory $\implies \delta_f > 0$ as control action in order to approach the reference trajectory (the steering wheel turns to the left from the driver's point of view).
- ii) W_{ct} : weight of the Cross-Track error.
- iii) $e_h = \psi_r - \psi$: **Heading error**. It's important to notice that the difference between the two angles (in radians) can go outside the interval $[-\pi, \pi]$, so we need to normalize the difference in this interval in the practical implementation. We have the same convention of sign used for the Cross-Track error.
 - $e_h > 0$: The vehicle is on the left with respect to the reference trajectory.
 - $e_h < 0$: The vehicle is on the right with respect to the reference trajectory.
- iv) W_h : weight of the Heading error.

- **Adaptive Cruise Control**

The second part of the cost function regards the Longitudinal Dynamics of the system, so the Adaptive Cruise Control part of the controller is:

$$J_{ACC} = \sum_{k_p=0}^{N_p-1} W_v \cdot \Delta_v^2 + W_{\Delta_d} \cdot \Delta_d^2 \quad (5.3)$$

where:

- i) $\Delta_v = v_x(k_p) - v_{ref}(k)$: **Speed error**. It is the difference between the actual velocity of the CoG (v_x) of the vehicle at the current time step of the prediction horizon and the reference velocity of the leading vehicle (v_{ref}).
- ii) W_v : weight of the Speed error.
- iii) $\Delta_d = d(k_p) - d_{s,min}(k)$: **Safety Distance error**. It is the difference between the actual distance between vehicles at the current prediction horizon time step and the minimum safety distance computed online:

- $\Delta_d > 0 \implies d(k_p) > d_{s,min}(k)$: we are respecting the minimum safety distance.
- $\Delta_d < 0 \implies d(k_p) < d_{s,min}(k)$: we are not respecting the minimum safety distance.

The minimum safety distance should be computed as the distance covered by the controlled vehicle in 1 second, which is the typical reaction time of a human. Unfortunately, the using of a variable of the optimization problem, i.e. $v_x(k)$, will overload the solver: we need to know a priori the minimum safety distance to address the whole optimization problem. To deal with this issue we can think that, likely, the longitudinal speed of the controlled vehicle will be close to the speed of the leading vehicle at each time instant, otherwise the controller does not track the speed. Based on this reasonable consideration we can use the speed of the leading vehicle at the time k , instead of the actual speed of the controlled vehicle, to compute the minimum safety distance to be respected. Then: $d_{s,min}(k) = v_{ref}(k) \cdot t_{react}$, where $t_{react} = 1\text{ s}$ (anyway the reaction time can be changed to promote the safety of the passengers). The distance $d(k)$ between the vehicles is computed as the Euclidean norm between the front axle of the controlled vehicle and the front axle of the leading vehicle in a 2D environment. The choice of the two front axles of the vehicles as reference is only to be compliant to the Cross-Track error quantities. An offset reflecting the length of the leading vehicle can be easily added if needed; however, in a simulation environment, this detail is not particularly critical.

iv) W_{Δ_d} : weight of Safety Distance error.

Note 5.1

In the ACC the safety distance from the vehicle in front of the controlled vehicle should be fixed by the user at a constant value, as said in [8]. In the developing we want to try a different range of speeds, so the safety distance is based on the longitudinal speed of the vehicle. This is a small difference from the classic ACC available.

• **Comfort part**

$$J_{Com} = \sum_{k_p=0}^{N_p-1} W_{\Delta_{\delta_f}} \cdot \Delta_{\delta_f}^2 + W_{\Delta_a} \cdot \Delta_a^2 \quad (5.4)$$

where:

- i) $\Delta_{\delta_f} = (\delta_f(k_p) - \delta_f(k-1))$: **Steering Angle error**. It is the difference between the value of the Steering Angle at the current prediction horizon time step and the Steering Angle found at the previous time step. Practically, we want to have at each step a Steering Angle that is not too far from the previous control action. In this manner, the controller should mitigate sudden variations in the car's orientation, contributing to enhance driving comfort.
- ii) $W_{\Delta_{\delta_f}}$: weight of Steering Angle error.
- iii) $\Delta_a = a_x(k_p) - a_x(k-1)$: **Acceleration error**. It is the difference between the value of the longitudinal acceleration at the current prediction horizon time step and the longitudinal acceleration found at the previous time step. The reasoning behind this component of the cost function is the same of the Steering Angle error: $a_x(k)$ and $a_x(k-1)$ must not diverge significantly, otherwise the passengers will experience a high and sudden acceleration or deceleration. Moreover, the longitudinal acceleration is related, by the torque provided by EM, to the battery power P_b , as seen in (3.31) and (3.32). Therefore, a too high acceleration or deceleration in a short time (time step) corresponds to a high power demand, that can be not available.

Note 5.2

The changing of the acceleration in a sampling time is actually the physical quantities named *jerk* in discrete time: $j = \frac{a_x(k) - a_x(k-1)}{T_s}$.

iv) W_{Δ_a} : weight of Acceleration error.

- For both the ACC part and the LKA part we need another tracking term inside the summation: we need a penalty on the difference between the desired value and the actual value for the first four states of the model:

$$x_{1 \rightarrow 4}^T \cdot Q \cdot x_{1 \rightarrow 4}$$

The vector of the first four states is indicated as:

$$x_{1 \rightarrow 4} = \begin{bmatrix} X(k_p) - X_s(k) \\ Y(k_p) - Y_s(k) \\ \psi(k_p) - \psi_r(k) \\ v_x(k_p) - v_{ref}(k) \end{bmatrix} \quad (5.5)$$

Q is the positive diagonal matrix of the weights for the elements of $x_{1 \rightarrow 4}$. This term helps the tuning between a fast or a slow tracking. To support the maintenance of a safety distance, inside the vector $x_{1 \rightarrow 4}$ the reference position to be tracked is (X_s, Y_s) , where s stands for *safe*. Specifically, the safe position is determined as follows:

- If the minimum safety distance at the previous time instant is respected ($\Delta_d > 0$) then we can track the real position of the leading vehicle: $X_s(k) = X_r(k)$ and $Y_s(k) = Y_r(k)$.
- If the minimum safety distance at the previous time instant is not respected ($\Delta_d < 0$) a distancing from the leading vehicle must be en-

forced:

$$X_s(k) = X_r(k) - d_{s,min}(k) \cdot \cos(\psi(k))$$

$$Y_s(k) = Y_r(k) - d_{s,min}(k) \cdot \sin(\psi(k))$$

- Moreover, for stability, a **terminal cost** must be added in the final cost function, and, once again, it will encompass the first four states. The vector of the first four states for the terminal cost is indicated as:

$$x_{1 \rightarrow 4, \text{term}} = \begin{bmatrix} X(N_p) - X_r(k) \\ Y(N_p) - Y_r(k) \\ \psi(N_p) - \psi_r(k) \\ v_x(N_p) - v_{ref}(k) \end{bmatrix} \quad (5.6)$$

P is the positive diagonal matrix of the weights for the elements of $x_{1 \rightarrow 4, \text{term}}$.

The final cost function will be:

$$J_1 = \left(\sum_{k_p=0}^{N_p-1} e^T \cdot W \cdot e + x_{1 \rightarrow 4}^T \cdot Q \cdot x_{1 \rightarrow 4} \right) + x_{1 \rightarrow 4, \text{term}}^T \cdot P \cdot x_{1 \rightarrow 4, \text{term}} \quad (5.7)$$

where:

$$W = \begin{bmatrix} W_{ct} & 0 & 0 & 0 & 0 & 0 \\ 0 & W_h & 0 & 0 & 0 & 0 \\ 0 & 0 & W_v & 0 & 0 & 0 \\ 0 & 0 & 0 & W_{\Delta_d} & 0 & 0 \\ 0 & 0 & 0 & 0 & W_{\Delta_{\delta_f}} & 0 \\ 0 & 0 & 0 & 0 & 0 & W_{\Delta_a} \end{bmatrix}$$

$$e = \begin{bmatrix} e_{ct} \\ e_h \\ \Delta_v \\ \Delta_d \\ \Delta_{\delta_f} \\ \Delta_a \end{bmatrix}$$

Note 5.3

Due to solver characteristics, as we will see in Section 5.4, the expression of the safety distance cannot be inserted as an *hard constraint*: the square root in the computation of the Euclidean distance between two points in a 2D environment is not differentiable in all the situations. So we need to insert it as *soft constraint*, i.e. as a part of the cost function J_1 .

5.2 Constraints Formulation

In general a constrained optimization problem can have two types of constraints:

- **Hard constraint**

They are rigid constraints, which must be respected in the solution of the problem. In general the violation of these types of constraints lead to a non admissible solution of the problem.

- **Soft constraint**

They are desirable situations in which we want to find optimal solution, but their violation is admissible. In general, their violation applies a penalty in the cost function of the problem and push the solver to a direction or another, based on the penalty. Anyway, their violation does not lead to non admissible solution, for these reasons they are integrated in the cost function: they are flexible constraints.

5.2.1 Hard Constraint for the NMPC Controller

In this section we will detail all the **hard constraints**, both from a physical and a practical point of view, in order to build the total constrained optimization problem for the NMPC controller.

Equality hard constraints for NMPC

For the integration of the model presented in Section 3.7 the Euler Discretization method is employed. It is accurate enough to provide satisfying results, but it is also simple to implement.

The use of Euler Discretization as an integration method imposes dynamical constraints on the system states. The expression of the Euler Discretization technique is given in (5.8)

$$x(k_p + 1) = x(k_p) + \dot{x}(k_p) \cdot h \implies x(k_p + 1) - [x(k_p) + \dot{x}(k_p) \cdot h] = 0 \quad (5.8)$$

where h corresponds to the sampling time chosen.

Basically during the prediction horizon we need to respect the expression of the integration of each state from $k_p = 0$ to $k_p = N_p - 1$. In the end, (5.8) are equality constraints of the optimization problem.

Note 5.4: Euler Discretization

Euler Discretization is one of the simplest and most intuitive numerical methods for solving Ordinary Differential Equations (ODEs). Its major characteristics are the following.

- In the explicit Euler method, the solution is updated using the slope (or derivative) evaluated at the current point. The update formula is:

$$y_{n+1} = y_n + h \cdot f(t_n, y_n) \quad (5.9)$$

where h is the time step amplitude and $f(t, y)$ represents the derivative term.

- The explicit Euler method can become unstable if the step size h is too large, especially for stiff problems. In such cases, the discretization may produce numerical solutions that diverge from the exact solution.

Geometrically, the explicit Euler method is based on the idea of approximating the solution curve of a differential equation by drawing, at each step, the tangent to the curve at the current point. This formula means that, starting from the point (t_n, y_n) , the slope $f(t_n, y_n)$ (i.e., the derivative of the solution) is computed, and then the employed method "moves forward" by a time interval h in the direction defined by this slope. In practice, the solution is approximated by successive line segments, each of which represents the tangent to the curve at a given point over a time interval h .

In summary, the explicit Euler method transforms the problem of determining the solution of a differential equation into a sequence of simpler problems, namely, tracing tangent lines that locally represent the system's dynamics.

As seen in Chapter 4 the terminal constraint is important for the stability of the NMPC controller, we choose to insert it only on some of the states, not on all the seven states. To be compliant to terminal cost seen in Section 5.1.1, also the

terminal constraint is applied only on the first four states. The equality constraints are expressed as:

$$X(N_p) - X_r(k) = 0 \quad (5.10a)$$

$$Y(N_p) - Y_r(k) = 0 \quad (5.10b)$$

$$\psi(N_p) - \psi_r(k) = 0 \quad (5.10c)$$

$$v_x(N_p) - v_{ref}(k) = 0 \quad (5.10d)$$

The terminal constraint is important not only for technical reasons related to stability, but it also carries a logical meaning: without it, the optimal solution of the optimization problem may be: to remain at the initial position, failing to drive the system forward. Meaningless result, in fact even if the terminal cost push the solution of the optimization problem to avoid this possible evolution of the system, it is admissible in the absence of the terminal constraints and can be chosen to minimize the cost function. The presence of a terminal constraint eliminates this possibility. Other strange situations that can happen in the absence of the terminal constraints are: to do not track the longitudinal speed in a good way or to find a trajectory that minimizes the difference from the minimum safety distance, but which is very different from the leading vehicle's trajectory.

Another essential point for the controller is the continuity between two successive time instant, for this reason the following equality constraints are added:

$$X(k_p) - X(k-1) = 0 \quad (5.11a)$$

$$Y(k_p) - Y(k-1) = 0 \quad (5.11b)$$

$$\psi(k_p) - \psi(k-1) = 0 \quad (5.11c)$$

$$v_x(k_p) - v_x(k-1) = 0 \quad (5.11d)$$

$$v_y(k_p) - v_y(k-1) = 0 \quad (5.11e)$$

$$\omega_\psi(k_p) - \omega_\psi(k-1) = 0 \quad (5.11f)$$

$$\zeta(k_p) - \zeta(k-1) = 0 \quad (5.11g)$$

The constraints from (5.11a) to (5.11g) are applied only for $k_p = 0$. The meaning of the previous expressions concerns the prediction horizon's first step: it should be as close as possible to the previous time instant status. The solution that the controller will use for the optimization problem at the time instant k must start from the information of the previous time instant $k - 1$. These constraints are essential for the good working of the controller, if they are not present the solver will looking for the solution of the optimization problem in a direction that does not take into account what is the actual condition of controlled vehicle. The absence of these constraints will lead to strange solution, such as sudden changing of position or speed in a T_s interval.

Inequality hard constraints for NMPC

Also inequality constraints are inserted, they are related to the physical limits that cannot be exceeded, like maximum value of Torque provided by the motor, or the maximum value of battery power. They are listed in the following.

- Minimum and maximum values of the control action are inserted to be sure to do not overcome the physical limits of the EM and to do not have an impossible rotation of the steering wheel.

$$T_{EM}^{min} \leq T_{EM}(k) \leq T_{EM}^{max} \quad \forall k. \quad (5.12)$$

$$\delta_f^{min} \leq \delta_f(k) \leq \delta_f^{max} \quad \forall k. \quad (5.13)$$

- Again for comfort reasons, but also for being consistent with the physical limitations of the vehicle and the electric powertrain, bounds on the longitudinal acceleration of the vehicle are applied:

$$a_x^{min} \leq a_x(k) \leq a_x^{max} \quad \forall k. \quad (5.14)$$

- To avoid deep discharge and overcharging, the State of Charge of the battery, which is the seventh state of the model, has to stay below a maximum

value and above a minimum value:

$$SOC_{min} \leq \zeta(k) \leq SOC_{max} \quad \forall k. \quad (5.15)$$

- The lateral acceleration of the vehicle must be maintained within suitable boundaries: a too high value of a_y can lead to lack of adhesion, that is a dangerous situation, and to discomfort for the passengers.

$$a_y^{min} \leq a_y(k) \leq a_y^{max} \quad \forall k. \quad (5.16)$$

- As said in Section 3.6 the battery power has an upper and a lower bound, they must be considered in the resolution of the optimization problem, so the following constraint is added:

$$P_b^{low} \leq P_b(k) \leq P_b^{up} \quad \forall k \quad (5.17)$$

where the battery power is given by: $P_b = \frac{P_{EM}}{[\eta_{EM}(\omega_{EM}, T_{EM}) \cdot \eta_{inv}]^{\text{sign}(P_{EM})}}$.

All the values of the limits are listed in the Table 5.1.

Note 5.5

The meaning of $\forall k$ is: for each time instant of the simulation, and so for each control action, the constraints must be respected. This implies that at each time instant of the prediction horizon (from $k_p = 0$ to $k_p = N_p - 1$) the constraints valid for $\forall k$ must be respected.

Note 5.6

In general a human passenger is more sensitive to a high deceleration than a high acceleration, so the limit of a_x^{min} is slightly less, in module, with respect to a_x^{max} .

Parameter	Symbol	Value	Unit
Minimum longitudinal acceleration	a_x^{min}	-2.5	m/s^2
Maximum longitudinal acceleration	a_x^{max}	3	m/s^2
Minimum lateral acceleration	a_y^{min}	$-\frac{g}{2}$	m/s^2
Maximum lateral acceleration	a_y^{max}	$\frac{g}{2}$	m/s^2
Minimum steering angle	δ_f^{min}	$-0.3 (-17.19)$	$rad(deg)$
Maximum steering angle	δ_f^{max}	$0.3 (17.19)$	$rad(deg)$
Minimum EM Torque	T_{EM}^{min}	-280	Nm
Maximum EM Torque	T_{EM}^{max}	280	Nm
Minimum State of Charge value	SOC_{min}	0.20	—
Maximum State of Charge value	SOC_{max}	0.90	—
Minimum battery power value	P_b^{low}	-132.71	kW
Maximum battery power value	P_b^{up}	132.71	kW

Table 5.1: Bounds of the inequality constraints for all controllers.

5.3 EMPC Design

For the EMPC design we can replace the tracking part of the cost function in Section 5.1.1 with an economic focus, in particular we will use a **battery focus**. Whereas the comfort part and the terminal cost will remain as before.

5.3.1 EMPC Cost Function

Again the cost function to be minimize consists of three main parts, and the terminal cost.

- **Battery Focus part**

This component of the cost function accounts for the energy required to execute the movement.

$$J_{Batt.Focus} = \sum_{k_p=0}^{N_p-1} W_{\Delta\zeta} \cdot \Delta\zeta^2 + W_{Pow} \cdot \left\| \frac{P_{EM}}{P_{ref}} \right\|^2 \quad (5.18)$$

where:

- i) $\Delta\zeta = \zeta(k_p) - \zeta(k-1)$: **SOC rate**. It is the difference between the SOC at the current prediction horizon time step and the SOC value at

the previous time step. This term aims to have a small variation of the SOC during the prediction horizon, and it can help to have a more regular behaviour of the battery, reducing the stress on the battery due to deep and fast cycle of charging or discharging.

- ii) W_{Δ_ζ} : is the weight of the SOC rate.
- iii) $\left\| \frac{P_{EM}}{P_{ref}} \right\|$: **Power Ratio**. It is the ratio between the power provided by the electric motor $P_{EM} = \omega_{EM} \cdot T_{EM}$ and the reference power $P_{ref} = F_{trac} \cdot v_{x,ref}$ that, approximately, the leading vehicle uses to move forward itself. Basically we can think that P_{ref} is the minimum power needed to follow a certain trajectory.
 - If $\left\| \frac{P_{EM}}{P_{ref}} \right\| > 1 \implies P_{EM} > P_{ref}$: we are using more power than (approximately) we need, so we want to decrease P_{EM} .
 - If $\left\| \frac{P_{EM}}{P_{ref}} \right\| < 1 \implies P_{EM} < P_{ref}$: we are using less power than (approximately) we need, so if necessary we can increase P_{EM} to follow the trajectory.
- iv) W_{Pow} is the weight of the Power Ratio.

Note 5.7

In discrete time the SOC rate should be $\Delta_\zeta = \frac{\zeta(k) - \zeta(k-1)}{T_s}$, but the constant value of the sampling time is absorbed by the weight W_{Δ_ζ} .

• **Comfort part**

The comfort part is exactly the same as in (5.4):

$$J_{Com} = \sum_{k_p=0}^{N_p-1} W_{\Delta_{\delta_f}} \cdot \Delta_{\delta_f}^2 + W_{\Delta_a} \cdot \Delta_a^2$$

• **Safety Distance Consideration**

As said in Note 5.3 we need to insert a part of the cost function for the soft constraint on the safety distance, which is essentially equivalent to the

formulation presented in (5.1):

$$J_{SafeDistance} = \sum_{k_p=0}^{N_p-1} W_{\Delta_d} \cdot \Delta_d^2 \quad (5.19)$$

Once again, for stability reasons, we need to add the terminal cost, as to the one previously employed in (5.7).

The total cost function will be:

$$J_2 = \left(\sum_{k_p=0}^{N_p-1} e_{Eco}^T \cdot W_{Eco} \cdot e_{Eco} \right) + x_{1 \rightarrow 4, \text{term}}^T \cdot P \cdot x_{1 \rightarrow 4, \text{term}} \quad (5.20)$$

where:

$$W_{Eco} = \begin{bmatrix} W_{\Delta_\zeta} & 0 & 0 & 0 & 0 \\ 0 & W_{Pow} & 0 & 0 & 0 \\ 0 & 0 & W_{\Delta_{\delta_f}} & 0 & 0 \\ 0 & 0 & 0 & W_{\Delta_a} & 0 \\ 0 & 0 & 0 & 0 & W_{\Delta_d} \end{bmatrix}$$

$$e_{Eco} = \begin{bmatrix} \Delta_\zeta \\ \left| \frac{P_{EM}}{P_{ref}} \right| \\ \Delta_{\delta_f} \\ \Delta_a \\ \Delta_d \end{bmatrix}$$

P and $x_{1 \rightarrow 4, \text{term}}$ has the same meaning as seen in (5.7)

5.3.2 Hard Constraints for the EMPC Controller

The EMPC controller uses the same hard constraints, both equality and inequality, as the NMPC, as detailed in Section 5.2.1.

This situation highlights that the problem to be solved is the same, from the physical and logic viewpoints; the constraints indicate what are the limits and admissible trajectory that the controller can use to solve the control problem, but the approach is slightly difference: the cost function is different.

5.4 Practical implementation on MATLAB

In the previous sections we have detailed the optimization problem, in a logic e theoretical manner. In the real implementation some adjustments and approximations must be done to fit the problem for the practical method used for finding the solution. In particular, the solver used and its setting has to meet the form of the problem. In the following sections, all practical observations and problems encountered during the development of the controller are examined, and the practical solutions adopted in the final work are presented.

5.4.1 Selecting the Sampling Time

The utilization of the Euler Discretization method implies some limitations on the sampling time T_s . We can imagine that the informations, such as pose and longitudinal speed, of the leading vehicle are transmitted periodically to the controlled vehicle in order to allow a successful "pursuit". We can think that the leading vehicle should transmit continuously its status informations, but it is impossible to manage: the controlled vehicle needs some time to perform the optimization and to perform the control action. Thus, it's necessary a critical task: selecting an appropriate sampling time. Unfortunately, another complication arises: the sampling time becomes the time interval h in which the integration is numerically performed. So if we want to have good performance we need a T_s small enough to be compatible with the limitations of the Euler's method, but great enough to do not be high demanding from a computational and simulation point of view.

After some attempts we conclude:

- $T_s = 100\ ms$ is too high to have a good accuracy in the numerical integration.
- $T_s = 10\ ms$ generates a great computational demand also for very short simulation, then will be unmanageable in a real application of a controller of this kind.
- **A good trade-off is identified in $T_s = 50\ ms$, this value is used for all the simulations in the following sections.**

5.4.2 Setting of the Solver

The implementation of the controller is developed in MATLAB, exploiting the features of CasADi for solving the constrained optimization problem. CasADi is a symbolic framework to solve nonlinear optimization problem, among the different solver available the IPOPT is chosen. Some options can be configured in the solver in order to "escape" from particularly challenging part of the trajectory:

- `opts.ipopt.max_iter = 300`: the maximum number of iteration that the solver will perform for evaluating the solution is 300, it is a good trade-off between computational effort and satisfying performances.
- `opts.ipopt.tol = 1e-3`: it is the convergence tolerance on the solution of the problem.
- `opts.ipopt.constr_viol_tol = 1e-3`: it is the maximum admissible tolerance for the violation of the constraints to consider valid a solution.

Note 5.8

Even if the violation of the hard constraints will lead to a non admissible solution of the problem, in some cases the constraints are inevitably violated. For example, in the middle of a chicane the sudden changing of the orientation of the curve to be performed is a challenging area of the path for the controller. In these particular cases, the physical nature of the problem does not permit a solution within the admissible range of the solution. Therefore, a reasonable tolerance is introduced to obtain a satisfying solution, even if the constraints are not fully respected, without compromising the overall problem.

Initial Guess

The solver needs an initial guess of the solution for starting the optimization problem. The initial guess is fundamental: if it is too far from the optimal solution the solver cannot converge. Choosing this information is a critical task. The same logic explained in Section 5.2.1 for the constraints from (5.11a) to (5.11g) is applied here: the initial guess of the solution of the optimization problem, both for

all the states and the control actions, at time k is chosen equal to $\mathbf{x}$ and $\mathbf{u}$ at the previous time instant: $k - 1$.

The reasoning behind this choice is simple: we want to follow a leading vehicle, driven by a human; then we can imagine that the leading vehicle will perform a smooth trajectory, without sudden changing (in speed, orientation, position etc.) in normal condition. So, at each time instant the actual and the previous status of the leading vehicle will be close, then also for the controlled vehicle should be the same. In conclusion, we can be quite sure that the solution of the problem at time k and time $k - 1$ will be close. Clearly, we are assuming that exceptional events, like an animal on the road or a burst of wind, will not occur.

5.4.3 Numerical Approximation to facilitate the Solver

The IPOPT solver addresses the solution of the problem using the gradient of the objective function and the Jacobians of the constraints, basically it needs that everything is differentiable for solving the problem. Unfortunately, we have seen in Chapter 3 that some of the relationships used are not always differentiable, some of them are not even continuos: for example the function $\text{sign}(\cdot)$ is not continuos around 0. To overcome these problems of differentiability or to have a more stable function from the numerical point of view some approximation are made. The approximations used in the constrained problem are listed below:

- **Sign function**

The function $\text{sign}(y)$ is not continuos for $y \rightarrow 0$, unfortunately we need to use this function to determine if we are in charging or discharging mode, but we can also be in a particular condition: $P_{EM} = 0$. This latter situation occurs during costing: we do not need to accelerate, but we do not need a braking, basically the vehicle goes on thanks to its inertia. The function $\text{sign}(y)$ can be approximated with the hyperbolic tangent:

$$\text{sign}(y) \approx \tanh(\gamma y) \quad \text{for } \gamma \gg 1 \quad (5.21)$$

when $\gamma \gg 1$ the S-shaped $\tanh(\gamma y)$ can approximate quite well the $\text{sign}(y)$ function, but the hyperbolic function is continuos $\forall y \in \mathbb{R}$. In the implemen-

tation is used $\gamma = 1 \cdot 10^3$. For all controllers this approximation is used.

- **Exponential Operation**

The exponential operation can generate numerical problems during the computation of the solution of the optimization problem, thus we replace this operation with an equivalent mathematical form, but numerically stable:

$$a^b = e^{b \cdot \ln(a)} \quad (5.22)$$

The expression $e^{b \cdot \ln(a)}$ is not only numerically more stable, but the differentiation is easier. For all controllers this equivalence is used.

- **Evaluation of $P \rightarrow 0$**

The approaching of a very low value of the power (for example in coasting or close to a costing situation) in the computation can generate problem or indefinite expression like 0^0 , to avoid these situations for the power values a little tolerance of 1 mW is added based on the sign of the power:

$$P_b \approx P_b + 1 \cdot 10^3 \cdot \text{sign}(P_b) \approx P + 1 \cdot 10^3 \cdot \tanh(\gamma P_b) \quad (5.23)$$

This tolerance is added at all the expressions of P_{EM} and P_b , both in symbolic form and quantitative form to be consistent. For all controllers this precaution is used.

5.5 Tuning and Simulation Results

The tuning of the weights of a MPC controller is a critical task: a high value of a weight push towards a specific direction the solver in an aggressive way. So the balance between different values of weights is not easy to find. For complex algorithm months can be requested to reach a suitable tuning. We have not that aim, the tuning is import to verify the good working of the algorithm, but long tuning phase are unmanageable in this work. So the aim of the following simulations is to show the influence of the changing of the weights on the final results in a quantitative way.

The tuning is divided in three main parts:

1. In the first part the NMPC and the EMPC approach are tested on a very short path, named Reference Trajectory 000. This path is used to verify the good working of the controller and find an acceptable length of the prediction horizon.
2. In the second part the simulations are carried out on a longer path, more similar to a real environment in which a car could be. This path, named Reference Trajectory 001, is used to find a satisfying balance between the weights.
3. In the third and final part of simulations, a significantly longer path with respect to the second phase of simulations, named Reference Trajectory 002, is used to test the best combinations of weights found in the second phase in order to understand if the using of EMPC approach can lead to a significant energy saving without losing the tracking performances.

In the end the third controller is tested on the Reference Trajectory 001 and compared to the NMPC.

5.6 Simulation on Reference Trajectory 000

The Reference Trajectory 000 is a short path, generated from a sinusoid of amplitude $10m$. The coordinates X_r, Y_r of the reference trajectory are related by the following expression:

$$Y_r = 10 \sin(0.04X_r) \quad (5.24)$$

While the heading of the car is computed using the `atan2` native function of MATLAB, which return a value of the heading in the interval $[-\pi, \pi]$. Regarding the longitudinal speed, for this first track a constant value is chosen. This is a quite problematic point: the curves that the car will perform are quite challenging, and it is not easy to find a feasible solution in which, at the apex of the curve, we can maintain the constant speed. However, it can be useful to assess the limitations of the algorithm when handling a challenging curve.

Note 5.9

The algorithm is though for quite high speed: the DST is not valid at low speed, so we cannot choose a longitudinal speed lower than 30km/h (8.33m/s) in all the simulations.

The main characteristics of the reference trajectory are listed in Table 5.2.

Characteristic	Value	Unit
Total length	622.95	m
Total Time of travel	80	s
Longitudinal constant speed	40 (11.11)	$\text{km/h}(\text{m/s})$

Table 5.2: Characteristics of Reference Trajectory 000.

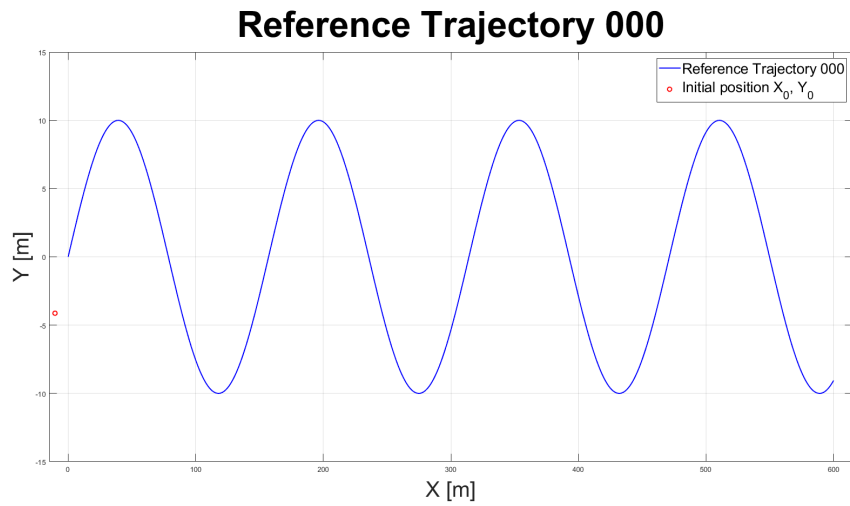
The initial condition of the simulation is given from the values in Table 5.3.

Initial condition	Symbol	Value	Unit
Initial coordinates on the x -axis	X_0	-10.31	m
Initial coordinates on the y -axis	Y_0	-4.12	m
Initial heading of the vehicle	ψ_0	0.38 (21.81)	$\text{rad}(\text{deg})$
Initial longitudinal speed of the vehicle	$v_{x,0}$	11.11	m/s
Initial lateral speed of the vehicle	$v_{y,0}$	0	m/s
Initial yaw rate of the vehicle	$\omega_{\psi,0}$	0	rad/s
Initial value of SOC	ζ_0	0.80	-

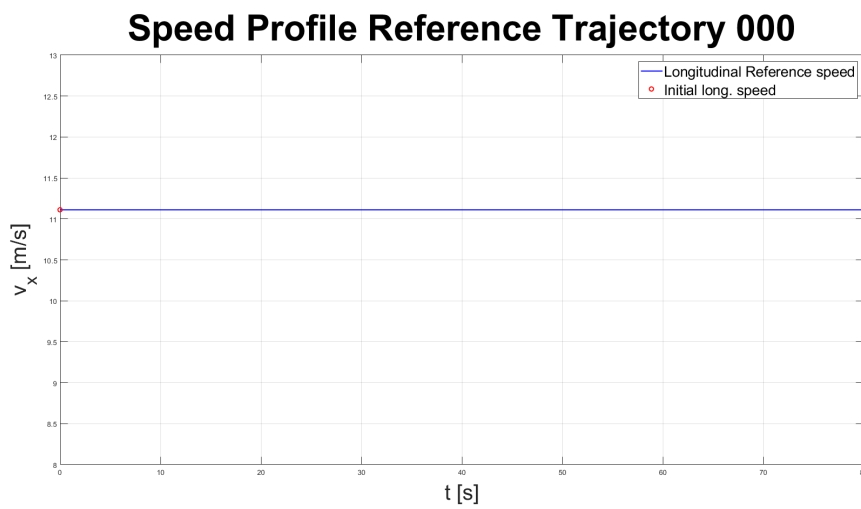
Table 5.3: Initial conditions Reference Trajectory 000.

Values in Table 5.3 are chosen for having the controlled vehicle starting perfectly aligned with the leading vehicle, proceeding at the same speed and in a position X_0, Y_0 that respect the minimum safety distance.

The trajectory and the speed profile of Reference Trajectory 000 are shown in Figure 5.1a and Figure 5.1b.



(a) Reference Trajectory 000.



(b) Speed profile Reference Trajectory 000.

Figure 5.1: The reference trajectory and corresponding speed profile employed in the first set of simulations, on Reference Trajectory 000.

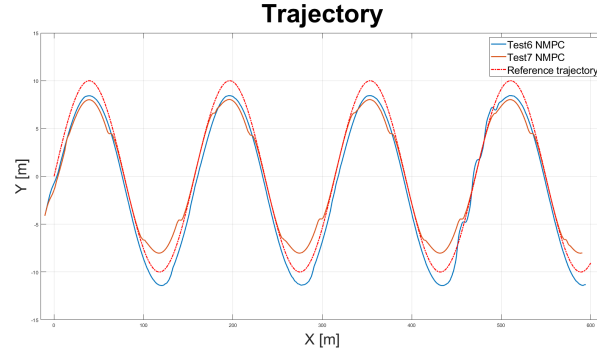
The simulations on this track have the aim to find the suitable prediction horizon length. A rough tuning of the weights of the cost function is used to perform the tests, in Table 5.4 are listed the values for the NMPC controller.

Reference Weight	Value
W_{ct}	100
W_h	10
W_v	2
W_{Δ_d}	1
W_{δ_f}	$1.5 \cdot 10^4$
W_{Δ_a}	2
Q	I_4
P	I_4

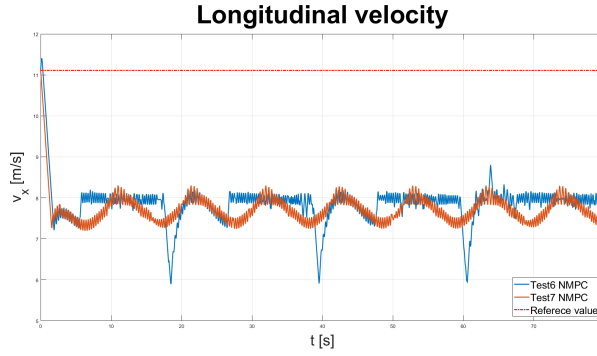
Table 5.4: Weights for selecting the prediction horizon length.

The results of the simulations performed on Reference Trajectory 000 for the NMPC controller are listed in Table 5.5. As expected, when N_p increases the computational effort of the optimization problem increases: the optimization problem becomes more difficult. This is true until Test 5, then the average time to solve the optimization problem starts to decrease, this effect is probably caused by the tuning of the weights of the cost function: the combination of values of the weights helps high values of N_p . However, the high value of N_p will be a problem in a more structured tuning: intuitively is more difficult to predict a more distant future, so it is only a lucky case that this tuning of the weights lowers down the computational effort of Test 8 and Test 9. Moreover, the value of the tracking performances of Test 8, Test 9 are worst with respect to test 7: the improvement is only on the computational effort.

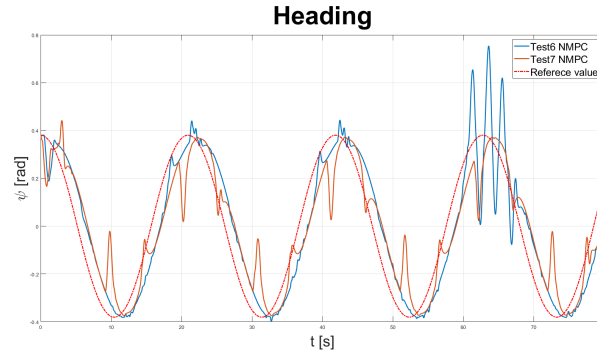
Acceptable performances are given from Test 6 and Test 7. In Figure 5.2a, Figure 5.2b, Figure 5.2c are shown their performances.



(a) Comparison Test 6 and Test 7 of Table 5.5 in tracking the Reference Trajectory 000.



(b) Comparison Test 6 and Test 7 of Table 5.5, longitudinal speed tracking.



(c) Comparison Test 6 and Test 7 of Table 5.5, Heading tracking.

Figure 5.2: NMPC tests on Reference Trajectory 000 for choosing N_p .

Test	N_p	Avg. time to solve the optimization problem [s]	$\max e_{ct} [m]$	$\max e_h [rad]$
1	5	0,22	251,84	3,12
2	10	0,37	48,99	0,43
3	12	0,46	38,98	0,54
4	15	0,57	27,52	0,74
5	18	0,76	13,97	0,29
6	20	1,22	2,09	0,38
7	22	0,97	2,68	0,36
8	25	0,74	3,08	0,44
9	28	0,90	5,68	0,61

Table 5.5: Tests for choosing the prediction horizon of NMPC on Reference Trajectory 000.

From the rough performances showed in Figures 5.2 we can start a more structured tuning for finding the right balance of the weights. In order to have a meaningful comparison with the EMPC controller the same value of prediction horizon is used in all the following tests for the NMPC and EMPC ($N_p = 22$).

5.7 Simulation on Reference Trajectory 001

The second set of simulations is carried out on Reference Trajectory 001, which is generated by means of Automated Driving Toolbox of MATLAB (as Reference Trajectory 002). The toolbox allows choosing some waypoints through which the vehicle must travel, and then a trajectory connecting those points is generated. Moreover, the toolbox can assign a specific speed at the vehicle when it passes through waypoints, but the changing of speed, for example due to a deceleration between two successive waypoints, is linear. A linear change of speed is not a real situation: it is unnatural for a driver to follow a pattern of changing speed that follows a precise linear law. Thus, to have a more real speed profile to associate at the trajectory, the MATLAB function `csaps` is used to smoothly interpolate the speed values assigned at the waypoints: the resulting speed profile is shown in Figure 5.3b. The main characteristics of this second path are listed in Table 5.6, in this case the vehicle's speed is not constant throughout the entire

trajectory. Notably, the varying longitudinal speed reflects a more realistic driving scenario.

Characteristic	Value	Unit
Total length	4200	<i>m</i>
Total Time of travel	190.55	<i>s</i>
Max Longitudinal speed	103.24 (28.68)	<i>km/h(m/s)</i>
Min Longitudinal speed	69.45 (19.29)	<i>km/h(m/s)</i>

Table 5.6: Characteristics of Reference Trajectory 001.

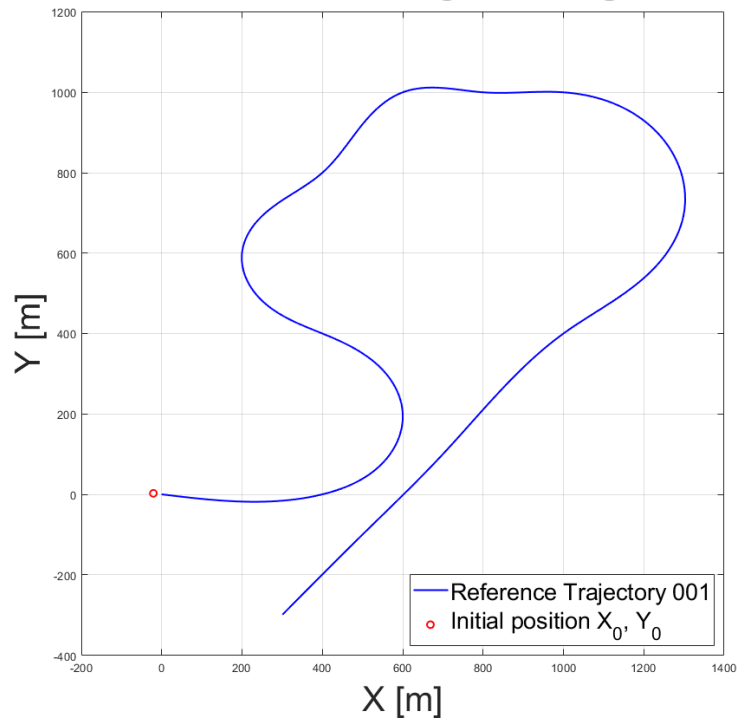
Note 5.10

The Economic approach is expected to promote smooth reference tracking, such as speed, to minimize energy consumption. In contrast, classical tracking may rely on high energy usage to follow even sharp reference variations. Since we are in a scenario in which both the reference trajectory and the speed profile are already smooth (see in Figure 5.3a and Figure 5.3b), we will expect only a slight difference between the energy consumption of the two controllers. However, sudden change of some quantities, or sharp behaviour can occur in real-world, where the EMPC can fully demonstrate its energy-saving potential.

The initial condition of the simulation is given from the values in Table 5.7. Once again, we can see that the controlled vehicle starts perfectly aligned with the leading vehicle, moving at the same speed and positioned at coordinates X_0, Y_0 , which complies with the minimum safety distance.

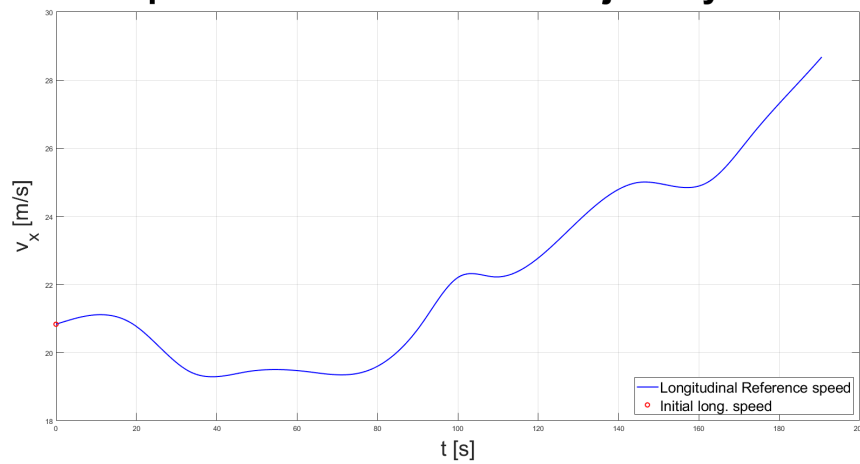
For this second phase of testing, the prediction horizon is set to $N_p = 22$ for all trials, corresponding to a future prediction window of $T_s \cdot N_p = 1.1 s$. Now a more structured tuning is carried out on both NMPC controller and EMPC controller. Since the EMPC is the most important part of the work, more tests are carried out employing that controller.

Reference Trajectory 001



(a) Reference Trajectory 001.

Speed Profile Reference Trajectory 001



(b) Speed profile Reference Trajectory 001.

Figure 5.3: The reference trajectory and corresponding speed profile employed in the second set of simulations, on Reference Trajectory 001.

Initial condition	Symbol	Value	Unit
Initial coordinates on the x -axis	X_0	-20.6837	m
Initial coordinates on the y -axis	Y_0	2.4927	m
Initial heading of the vehicle	ψ_0	-0.12 (-6.9)	$rad(deg)$
Initial longitudinal speed of the vehicle	$v_{x,0}$	20.83	m/s
Initial lateral speed of the vehicle	$v_{y,0}$	0	m/s
Initial yaw rate of the vehicle	$\omega_{\psi,0}$	0	rad/s
Initial value of SOC	ζ_0	0.80	-

Table 5.7: Initial conditions for all the simulations on Reference Trajectory 001.

Test	W_{ct}	W_h	W_v	$W_{\Delta\delta_f}$	$W_{\Delta a}$	$W_{\Delta d}$	Q	P	$\max e_{ct} [m]$	$\max e_h [rad]$
1	1	1	1	$1 \cdot 10^4$	2	1	I_4	I_4	2,16	0,144
2	10	1	1	$1 \cdot 10^4$	2	1	I_4	I_4	2,14	0,144
3	10	10	1	$1 \cdot 10^4$	2	1	I_4	I_4	2,14	0,144
4	10	25	1	$1 \cdot 10^4$	2	1	I_4	I_4	2,14	0,144
5	100	1	1	$1 \cdot 10^4$	2	1	I_4	I_4	1,98	0,143
6	100	10	1	$1 \cdot 10^4$	2	1	I_4	I_4	1,98	0,143
7	100	25	1	$1 \cdot 10^4$	2	1	I_4	I_4	1,98	0,143
8	100	25	1.5	$1 \cdot 10^4$	2	1	I_4	I_4	1,98	0,143
9	100	25	2	$1 \cdot 10^4$	2	1	I_4	I_4	1,98	0,143

Table 5.8: Tuning results of the NMPC controller, on Reference Trajectory 001.

In Table 5.8 are reported the tests on Reference Trajectory 001, with the NMPC controller. The aim of these tests is to identify the combination of weights that yields the best tracking performance. We can see that from Test 5 to Test 9 the tracking performances are practically the same, thus to choose the reference NMPC controller to be compared with the EMPC controller we can evaluate the computational effort of these tests. In Table 5.9 we can see that the Test 9 is the fastest to build and solve the optimization problem, so it is selected as benchmark.

Test	Avg. time to solve the optimization problem [s]
1	0,56
2	0,58
3	0,58
4	0,57
5	0,61
6	0,59
7	0,59
8	0,66
9	0,56

Table 5.9: Average computational time of NMPC controller for different trials on Reference Trajectory 001.

Then the EMPC controller is tested on the Reference Trajectory 001, but now we are evaluating the energy savings achieved by the EMPC in comparison to the benchmark provided by Test 9 of the NMPC controller, which, at the end of the simulation, reaches a final value of the SOC equal to 77,6834%. In Table 5.10 we can see the major results of the tuning, particular important is the last column of the table, where we can see if the EMPC can save energy with respect to the NMPC, in case of a positive value, or it has worst energy performances, in case of a negative value, or it has the same energy performances, in case of 0 as value reported.

Test 7, 21, 24 do not converge during the evaluation of the optimization problem, so the solver of CasADi returns an error (NaN detected) at some time instant k during the simulation because it cannot find a proper solution for the problem.

Test	W_{Δ_ζ}	W_{pow}	$W_{\Delta_{\delta_f}}$	W_{Δ_a}	W_{Δ_d}	P	Final SOC[%]	SOC [%] saved
1	1	1	$1,5 \cdot 10^4$	2	1	I_4	77,6831	-0,0003
2	1	1	$1,5 \cdot 10^4$	2	1	I_4	77,6831	-0,0003
3	$1 \cdot 10^6$	1	$1,5 \cdot 10^4$	2	1	I_4	77,6833	-0,0001
4	$1 \cdot 10^6$	10	$1,5 \cdot 10^4$	2	1	I_4	77,6835	0,0002
5	$1 \cdot 10^6$	25	$1,5 \cdot 10^4$	2	1	I_4	77,6832	-0,0002
6	$1 \cdot 10^6$	50	$1,5 \cdot 10^4$	2	1	I_4	77,6834	0,0000
7	$1 \cdot 10^6$	100	$1,5 \cdot 10^4$	2	1	I_4	Error	Error
8	$1 \cdot 10^7$	1	$1,5 \cdot 10^4$	2	1	I_4	77,6834	0,0000
9	$1 \cdot 10^7$	10	$1,5 \cdot 10^4$	2	1	I_4	77,6833	0,0000
10	$1 \cdot 10^7$	25	$1,5 \cdot 10^4$	2	1	I_4	77,6835	0,0001
11	$1 \cdot 10^7$	50	$1,5 \cdot 10^4$	2	1	I_4	77,6835	0,0001
12	$1 \cdot 10^7$	100	$1,5 \cdot 10^4$	2	1	I_4	76,0927	-1,5907
13	$2 \cdot 10^7$	1	$1,5 \cdot 10^4$	2	1	I_4	77,6836	0,0002
14	$2 \cdot 10^7$	10	$1,5 \cdot 10^4$	2	1	I_4	77,6835	0,0002
15	$2 \cdot 10^7$	25	$1,5 \cdot 10^4$	2	1	I_4	77,6837	0,0004
16	$2 \cdot 10^7$	50	$1,5 \cdot 10^4$	2	1	I_4	77,6837	0,0003
17	$2 \cdot 10^7$	100	$1,5 \cdot 10^4$	2	1	I_4	77,6840	0,0006
18	$3 \cdot 10^7$	1	$1,5 \cdot 10^4$	2	1	I_4	77,6836	0,0002
19	$3 \cdot 10^7$	10	$1,5 \cdot 10^4$	2	1	I_4	77,6835	0,0002
20	$3 \cdot 10^7$	25	$1,5 \cdot 10^4$	2	1	I_4	77,6837	0,0003
21	$3 \cdot 10^7$	50	$1,5 \cdot 10^4$	2	1	I_4	Error	Error
22	$3 \cdot 10^7$	100	$1,5 \cdot 10^4$	2	1	I_4	73,9387	-3,7447
23	$1 \cdot 10^8$	1	$1,5 \cdot 10^4$	2	1	I_4	77,4578	-0,2255
24	$1 \cdot 10^8$	10	$1,5 \cdot 10^4$	2	1	I_4	Error	Error
25	$1 \cdot 10^8$	25	$1,5 \cdot 10^4$	2	1	I_4	77,3124	-0,3710
26	$1 \cdot 10^8$	50	$1,5 \cdot 10^4$	2	1	I_4	75,8659	-1,8175
27	$1 \cdot 10^8$	100	$1,5 \cdot 10^4$	2	1	I_4	73,9512	-3,7322

Table 5.10: Tuning results of the EMPC controller on Reference Trajectory 001.

Test 12, 23, 25, 26, 27 provides bad results for both tracking performances and final SOC value, then are ignored.

The best results in terms of energy savings were obtained in Tests 4, 13, 15, 16, 17, and 20. Subsequently, the computational effort and tracking performance of these tests were evaluated. As shown in Table 5.11, the tracking performance is nearly identical across all selected tests, while the computational effort is slightly lower for Tests 4, 13, and 17. Therefore, they are selected as the best trade-offs between computational effort and performances and compared with the benchmark (Test 9 for NMPC controller).

Test	Avg. time to solve the optimization problem [s]	$\max e_{ct} [m]$	$\max e_h [rad]$
4	0,42	2,19	0,145
13	0,42	2,19	0,144
15	0,44	2,19	0,144
16	0,44	2,19	0,144
17	0,42	2,19	0,144
20	0,43	2,19	0,144

Table 5.11: Evaluation of the EMPC Controller's best performances on Reference Trajectory 001.

5.7.1 NMPC vs EMPC on Reference Trajectory 001

In Figure 5.4 the best trials of the EMPC controller are graphically compared with Test 9 NMPC, the Table 5.12 is inserted for a clearer reading of the results.

Test	Avg. time to solve the optimization problem [s]	$\max e_{ct} [m]$	$\max e_h [rad]$	Final SOC[%]
4 EMPC	0,42	2,19	0,145	77,6835
13 EMPC	0,42	2,19	0,144	77,6836
17 EMPC	0,42	2,19	0,144	77,6840
9 NMPC	0,56	1,98	0,143	77,6834

Table 5.12: Resume of bests EMPC tests vs benchmark NMPC on Reference Trajectory 001.

The differences between the main performances for the two controllers are poor from the tracking viewpoint, but interesting for the final SOC and the average computational time. For the Cross-Track error the difference is about $0.16m$, whereas is about 0.01 rad (i.e. 0.57°) for the Heading error, negligible differences in large scale.

From the energy consumption point of view (SOC evolution and Power Ratio) Figure 5.4a illustrates how the EMPC enhances energy recovery during regenerative braking, thereby improving energy efficiency over the entire trip, in fact in Table 5.12 we can read that the Test 4 saves the 0.0001% of SOC, whereas Test 13 the 0.0002% and the Test 17 the 0.0006% .

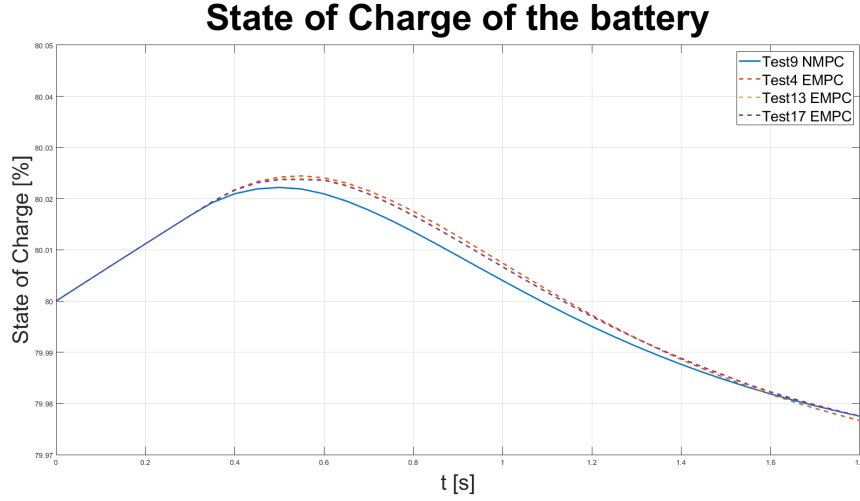
Regarding the Power Ratio, in Figure 5.4b we can see that:

- When $\frac{P_{EM}}{P_{ref}} > 1$, the NMPC controller leads to a greater Power Ratio (the blue solid line is above the others). The meaning is that in general the NMPC controller requires more power than it really needs to follow the trajectory.
- When $\frac{P_{EM}}{P_{ref}} < 1$ the EMPC controller leads to a lower power ratio (the dashed lines are below the solid line). Thus, in general, the EMPC requires less energy to follow the trajectory.

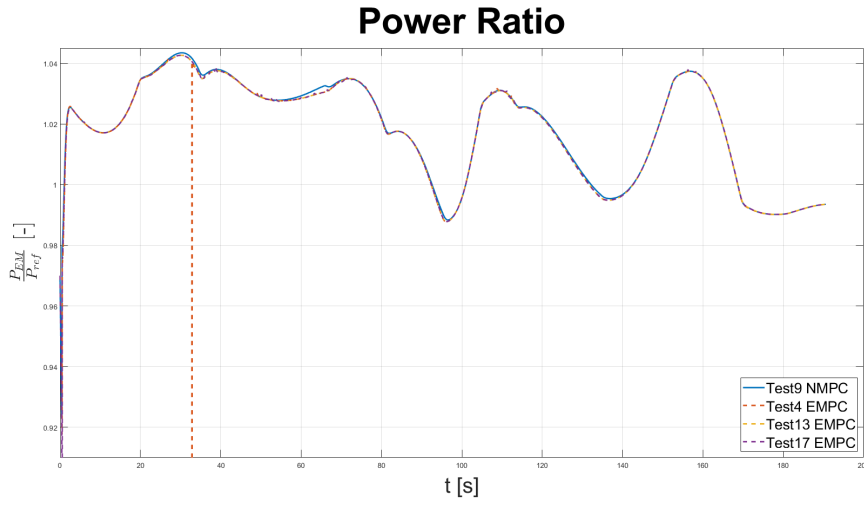
In average, the NMPC controller uses 1.83% of power more than the reference power needed, while the EMPC only the 1.78% . On a short track like Reference Trajectory 001 the energy saved is a very small amount, but normally a vehicle performs hours of travel, not just three minutes. In prospective the energy saved can be accumulated and exploited during a longer trip. Moreover, in the Reference Trajectory 001 there is only one short portion of regenerative braking, that can help to recharge the battery, while during a long trip this condition occurs multiple times, enhancing the overall energy recovery.

The computational time for setting and solving the optimization problem is also reduced for the EMPC controller, the 25% less, thanks to the simpler cost function with respect to the NMPC controller.

In the following section, a longer track is used as a reference to illustrate the potential improvement in energy-saving performance.

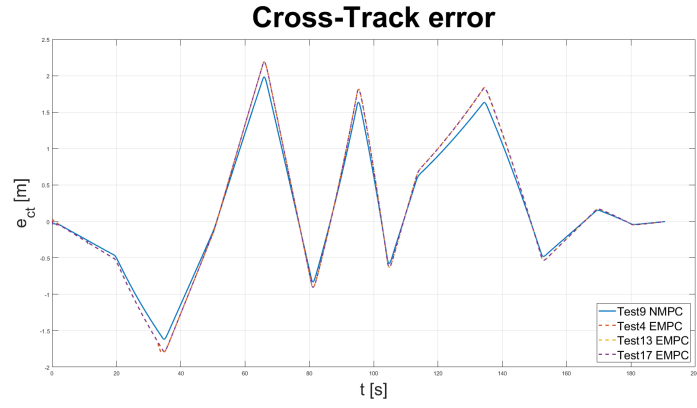


(a) SOC behaviour of tests in Table 5.12 on Reference Trajectory 001 during a regenerative braking.

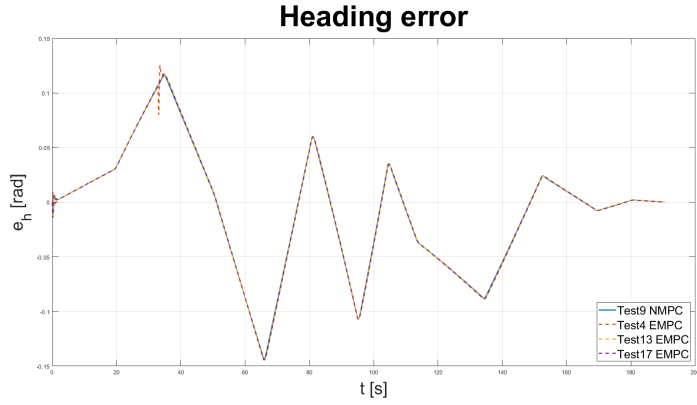


(b) Power ratio profile of tests in Table 5.12 on Reference Trajectory 001.

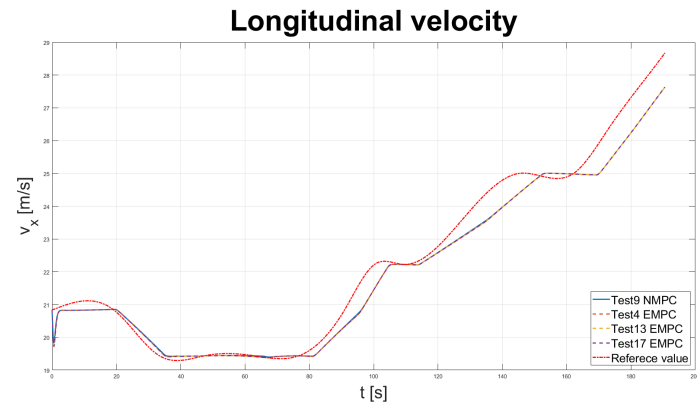
Figure 5.4: SOC behaviour and Power Ratio profile for tests in Table 5.12, on Reference Trajectory 001.



(a) Cross-Track error comparison for tests in Table 5.12 on Reference Trajectory 001.



(b) Heading error comparison for tests in Table 5.12 on Reference Trajectory 001.



(c) Speed Profile tracking for tests in Table 5.12 on Reference Trajectory 001.

Figure 5.5: Tracking performances comparison for tests in Table 5.12 on Reference Trajectory 001.

5.8 Simulation on Reference Trajectory 002

The last phase of simulations employing the EMPC controller concern a longer track, in order to highlight the increasing of energy performances with respect to the NMPC approach.

This part of the work is carried out on Reference Trajectory 002, which main characteristics are listed in Table 5.13.

Characteristic	Value	Unit
Total length	10043	<i>m</i>
Total Time of travel	445.900	<i>s</i>
Max Longitudinal speed	100.00 (27.78)	<i>km/h(m/s)</i>
Min Longitudinal speed	69.45 (19.29)	<i>km/h(m/s)</i>

Table 5.13: Characteristics of Reference Trajectory 002.

In Figure 5.6a and Figure 5.6b are shown the Reference Trajectory 002 and the speed profile associated to the trajectory.

The combination of weight of Test 4 and Test 15 in table 5.10 are chosen for the trials on Reference Trajectory 002. Even if they do not provide the best results, they demonstrate to be the most balanced: they are not too aggressive in the control action, which lead to a more adaptability to different paths, with satisfying performances. They are compared always with the combinations of weights of Test 9 in Table 5.8 for the NMPC and resumed in Table 5.14.

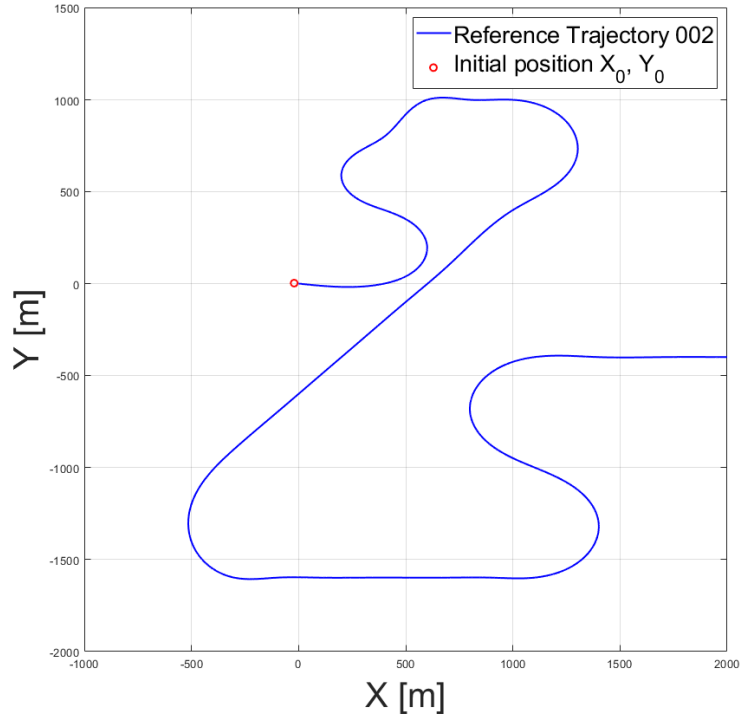
Test	W_{ct}	W_h	W_v	$W_{\Delta\delta_f}$	$W_{\Delta a}$	$W_{\Delta d}$	Q	P	Final <i>SOC</i> [%]
1	100	25	2	$1 \cdot 10^4$	2	1	I_4	I_4	75,1861

Table 5.14: Results of NMPC controller on Reference Trajectory 002.

Test	$W_{\Delta\zeta}$	W_{pow}	$W_{\Delta\delta_f}$	$W_{\Delta a}$	$W_{\Delta d}$	P	Final <i>SOC</i> [%]	<i>SOC</i> [%] saved
1	$1 \cdot 10^6$	10	$1,5 \cdot 10^4$	2	1	I_4	75,1867	0,0006
2	$2 \cdot 10^7$	25	$1,5 \cdot 10^4$	2	1	I_4	75,1875	0,0014

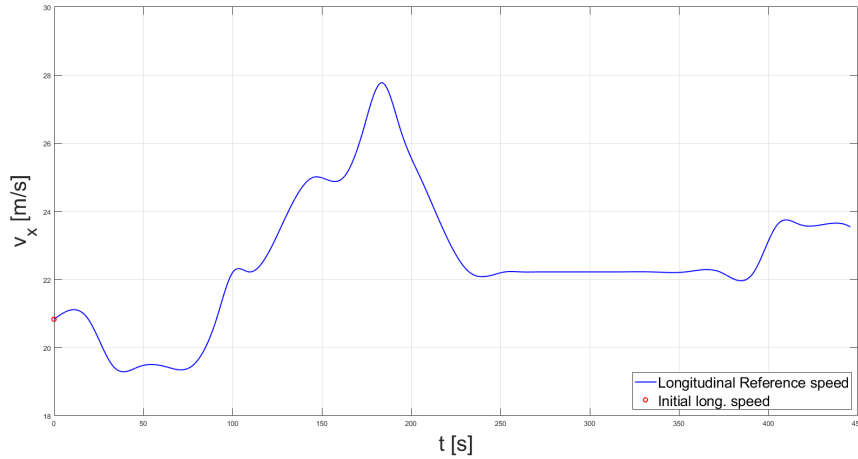
Table 5.15: Performance results of EMPC controller on Reference Trajectory 002.

Reference Trajectory 002



(a) Reference Trajectory 002.

Speed Profile Reference Trajectory 002



(b) Speed profile Reference Trajectory 002.

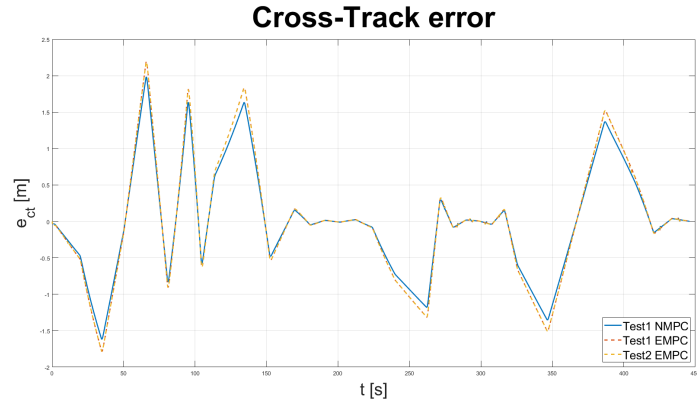
Figure 5.6: The reference trajectory and corresponding speed profile employed in the third set of simulations, on Reference Trajectory 002.

We can see in Table 5.15 that the recovery of the SOC is on the third digit after the comma for Test 2, while previously it was limited on the fourth digit after the comma.

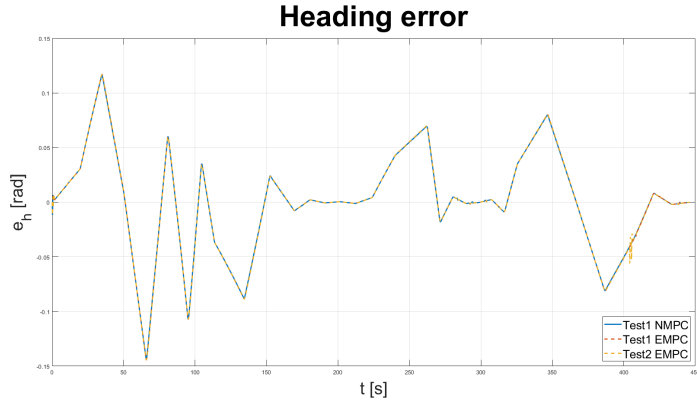
The increasing of the performances from the energy consumptions point of view is followed with a negligible difference on the tracking performances: the EMPC controller can track position, heading and longitudinal speed of the leading vehicle almost in the same way of the NMPC controller also in this scenario, as shown in Figure 5.7a, Figure 5.7b and Figure 5.7c.

Regarding the comfort of the passengers, the NMPC controller can handle a smooth behaviour of the Steering Angle, and small values of the lateral acceleration along all the path are detected. While the EMPC has more difficulties to reach a comfort behaviour, however the spikes of lateral acceleration and sudden changes of δ_f are limited inside the boundaries as showed in Figure 5.8a, Figure 5.8b.

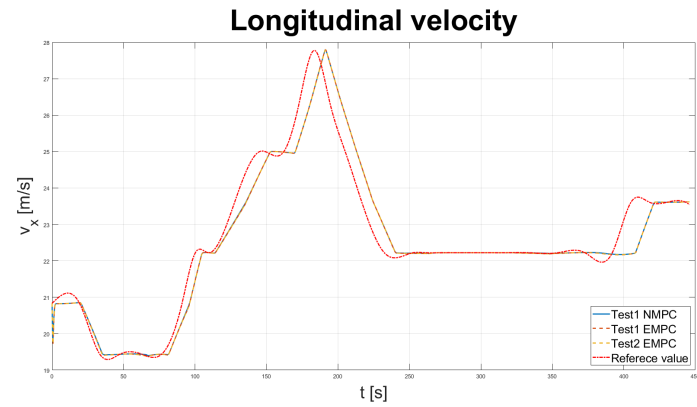
Moreover, the main objective of the ACC, which is the ability to maintain a minimum safety distance from the vehicle ahead meanwhile the vehicle tracks a reference speed, is reached with a margin of more than 1 m above the minimum for almost all the travel, as we can see in Figure 5.9.



(a) Cross Track error for tests in Table 5.14 and Table 5.15 on Reference Trajectory 002.

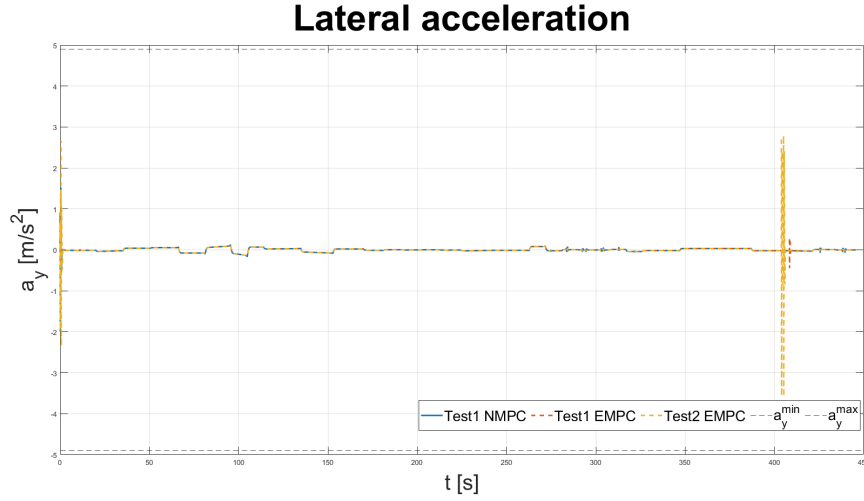


(b) Heading error for tests in Table 5.14 and Table 5.15 on Reference Trajectory 002.

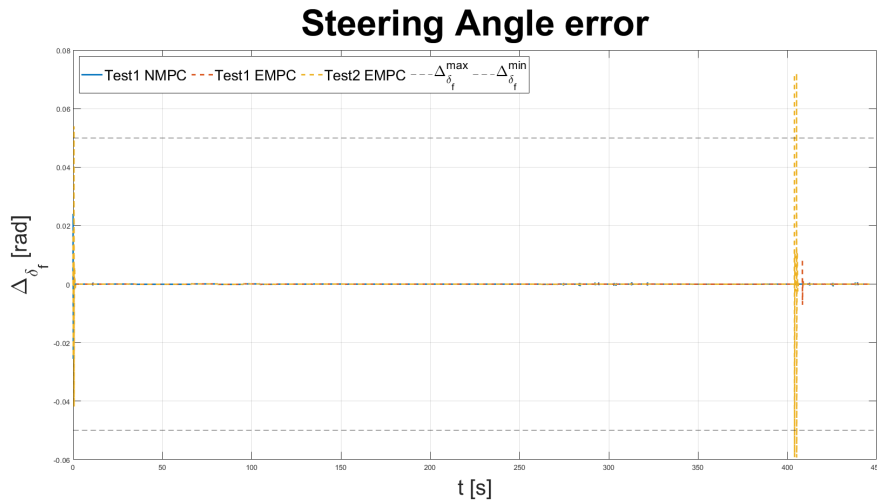


(c) Speed tracking performances for tests in Table 5.14 and Table 5.15 on Reference Trajectory 002.

Figure 5.7: Tracking performances comparison for tests in Table 5.14 and Table 5.15 on Reference Trajectory 002.



(a) Lateral acceleration behaviour for tests in Table 5.14 and Table 5.15 on Reference Trajectory 002.



(b) Steering Angle error behaviour for tests in Table 5.14 and Table 5.15 on Reference Trajectory 002.

Figure 5.8: Comfort performances for tests in Table 5.14 and Table 5.15 on Reference Trajectory 002.

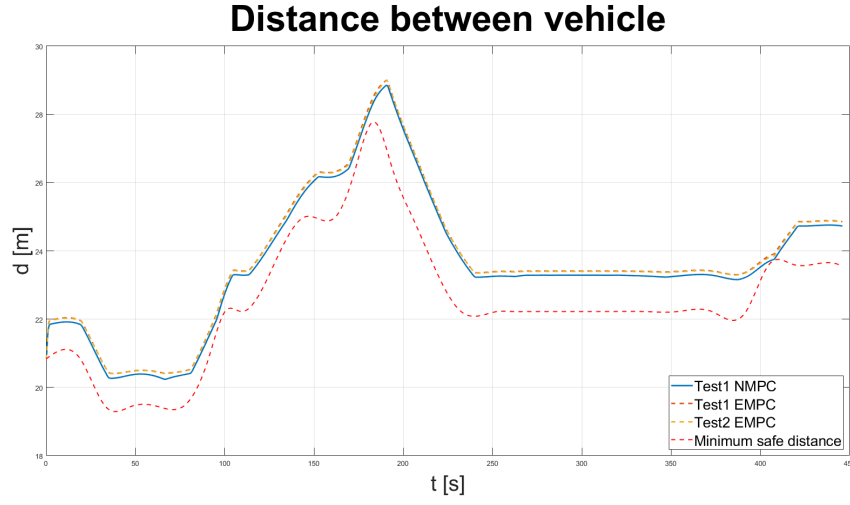


Figure 5.9: Distance between leading and controlled vehicle for tests in Table 5.14 and Table 5.15 on Reference Trajectory 002.

5.9 Control inputs Analysis: NMPC vs EMPC

During the setting of the optimization problem we have explained the importance of the bounds on the control action: in any real system the actuators are not able to provide an infinite effort. Additionally, a gradual and smooth behaviour of the control action can reduce degradation of the system and discomfort of the passengers. For these reasons the two control input, EM Torque and Steering Angle, should have a smooth behaviour. In Figure 5.10 are reported the T_{EM} behaviour on the Reference Trajectory 002. In the graph we can notice that the NMPC controller has a smooth behaviour of T_{EM} everywhere in the track (solid blue line): the NMPC does not choose a sudden change of the EM Torque as control action; the tracking focus has this advantage. While the EMPC controller produces significant spikes in different part of the track. In general, rapid shifts of the torque values are quite normal situations in a travel: for example, releasing the accelerator quickly, or pushing the break are normal actions during the drive and they influence the torque. However, all spikes remain below $30 Nm$ in magnitude, and are therefore manageable. Nonetheless, they may cause passenger discomfort and induce vibrations in the vehicle's mechanical components.

We can notice that the spikes of the torque are in the region in which the speed

error is very low, close to zero, or quite high. Practically, when the speed tracking is almost perfect we are close to Power Ratio equal to 1, then the controller chooses to decrease the power demand to save energy, but after a short time the speed error increases again reaching quite high value. Then the Power Ratio deviates from 1 and the controller compute an action which aims is to contain the Power Ratio change. Torque spikes are the price to pay for saving energy along segments of the path where the trajectory can be followed with less effort than required by the leading vehicle.

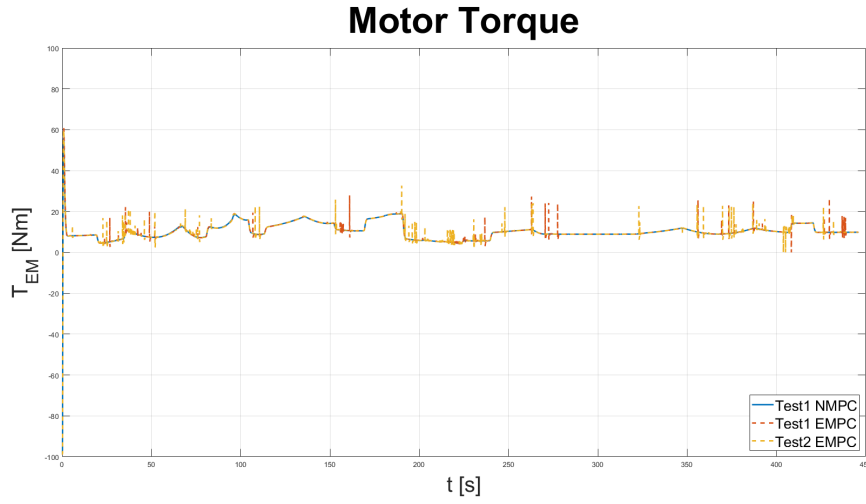


Figure 5.10: T_{EM} behaviour for tests in Table 5.14 and Table 5.15 on Reference Trajectory 002.

Regarding the behaviour of the Steering Angle, in Figure 5.11 we can notice that both the controllers have a smooth behaviour along all the trip, except for the initial regions. It is a normal behaviour: in the initial phase the controller needs to find a "good trajectory" for the following steps, so it has some oscillations of the Steering Angle. Anyway, after the initial phase with some spikes the evolution of the Steering Angle becomes smooth, the oscillating behaviour is only around 405 seconds and only for the Test 2 of the EMPC controller, that means the combinations of weights push the controller to have an aggressive control action in that particular area of the path.

The smoothness of the Steering Angle is a direct consequence of the presence of the Steering Angle error in both the controller: it would not be reasonable

to remove this term from the EMPC approach, as it is associated to passenger comfort rather than the controller's tracking objectives.

However, the combination of weights can influence the behaviour of the controller in an extreme way, then as said this is the longest part of the development process for this type of controller.

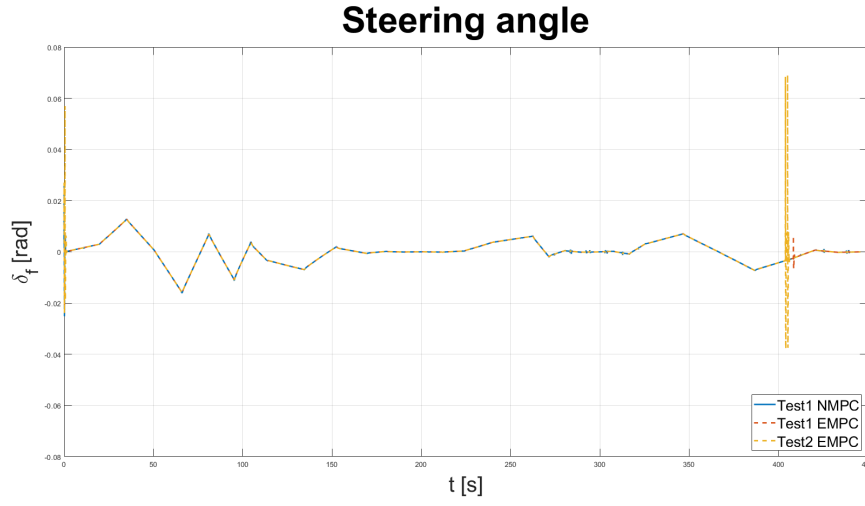


Figure 5.11: δ_f behaviour for tests in Table 5.14 and Table 5.15 on Reference Trajectory 002.

5.10 Controller Design with Averaging Conflicting Objectives Approach

The Conflicting Objectives NMPC will be designed in the following sections, and it will be tested only on Reference Trajectory 001. The comparison between the classic NMPC, detailed in Section 5.1, and this new approach will show a possibility of reducing the prediction horizon length maintaining satisfying performance.

In this approach we have three optimization problems, so we have three cost functions to handle, in the following section the details of the three cost functions, the constraints associated to the optimization problem and the practical implementation are described.

5.10.1 Cost Function for selecting the trade-off equilibrium point

The first optimization problem is to find the trade-off equilibrium point (x_s, u_s) .

The cost function that drives this problem has three main parts:

- **Tracking part**

$$\ell_r(x, u) = \sum_{k_p=0}^{N_p-1} W_{ct} \cdot e_{ct}^2 + W_h \cdot e_h^2 \quad (5.25)$$

- **Economic part**

$$\ell_e(x, u) = \sum_{k_p=0}^{N_p-1} W_{\Delta_\zeta} \cdot \Delta_\zeta^2 + W_{pow} \cdot \left\| \frac{P_{EM}}{P_{ref}} \right\|^2 \quad (5.26)$$

- **Comfort and safe part**

$$\ell_{Com}(x, u) = \sum_{k_p=0}^{N_p-1} W_{\Delta_a} \cdot \Delta_a^2 + W_{\Delta_d} \cdot \Delta_d^2 + W_{\Delta_{\delta_f}} \cdot \Delta_{\delta_f}^2 \quad (5.27)$$

Then the total stage cost for the first optimization problem will be:

$$\ell(x, u) = \sum_{k_p=0}^{N_p-1} e_{CO}^T \cdot W_{CO} \cdot e_{CO} \quad (5.28)$$

where:

$$W_{CO} = \begin{bmatrix} W_{ct} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & W_h & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & W_{\Delta_\zeta} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & W_{pow} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & W_{\Delta_a} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & W_{\Delta_d} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & W_{\Delta_{\delta_f}} \end{bmatrix}$$

$$e_{CO} = \begin{bmatrix} e_{ct} \\ e_h \\ \Delta\zeta \\ \left\| \frac{P_{EM}}{P_{ref}} \right\| \\ \Delta_a \\ \Delta_d \\ \Delta\delta_f \end{bmatrix}$$

5.10.2 Cost Function for selecting the weight of the additional term

When (x_s, u_s) is found, the additional term can be designed by means of the second optimization problem of the procedure. Unlike the previous constrained optimization problems we have addressed, the second one is formulated as a maximization problem. As detailed in [3], the function $\chi(z - z_s)$ is chosen as: $\chi(z - z_s) = |z - z_s|_1$, where z refers to all the states and all the control inputs of our problem. Unfortunately, $|x|$ is not differentiable for $x \rightarrow 0$, but CasADi needs differentiability of the functions that it will handle. To manage the problem an approximation can be employed:

$$|x| \approx \sqrt{x^2 + \varepsilon} \quad (5.29)$$

Then, the positive definite function $\chi(z - z_s)$, in our implementation, becomes:

$$\chi(z - z_s) \approx \sum_{i=1}^{n+m} \sqrt{(z_i - z_s)^2 + \varepsilon} \quad (5.30)$$

where $\varepsilon = 1 \cdot 10^{-3}$, n is the number of the states and m is the number of the control inputs.

The second constrained optimization problem adopts the objective function presented in (4.31), with a minor practical adjustment. In symbolic computation, the denominator of a function must not contain expressions that can evaluate to zero, as this causes difficulties for CasADi during processing. To address this,

a small tolerance is introduced in the practical implementation, where the goal is to find a that can stabilize the system in its trajectory. Although the stability properties of the term associated to the weight a have not yet been demonstrated, the advanced design procedure currently in progress may yield promising results. Then the optimal value of a is chosen as:

$$a^* = \max \left(\frac{\ell(x_s, u_s) - \ell(x_0, u_0)}{\chi(z_0 - z_s) + \varepsilon} \right) + \varepsilon \quad (5.31)$$

where $\ell(x_0, u_0)$ is the stage cost of the first optimization problem, but evaluated only in the first time step of the prediction horizon, the same concept is applied to $\chi(z_0 - z_s)$. The using of (x_0, u_0) as optimization variables in this second step is related to the complexity of the problem: in [3] the procedure for computing a^* includes as optimization variables all the states and all the control inputs in $\mathcal{Z} \setminus (x_s, u_s)$, but for a complex system it would be unaffordable. In conclusion, we need a lighter approach from the computational point of view, which is presented in this work.

Note 5.11

It can happen that $a^* < 0$, but it does make sense a negative weight in the cost function. This event implies that we cannot find a point (x, u) for which $\ell(x_s, u_s) > \ell(x_0, u_0)$, in other words (x_s, u_s) is already the point in the space $\mathcal{Z}$ that minimize the stage cost at the first prediction horizon step. In this case we can saturate a^* to 0, and do not perform the third optimization problem: we have already found the optimal control action by means of the first optimization problem.

5.10.3 Cost Function for finding the optimal control action

For the final optimization problem, we adopt the stage cost used in the first optimization problem, namely (5.28), but augmented by the additional term previ-

ously constructed.

$$\bar{\ell}(x, u) = \ell(x, u) + \alpha(x_0, u_0) = \sum_{k_p=0}^{N_p-1} e_{CO}^T \cdot W_{CO} \cdot e_{CO} + a^* \cdot \chi(z_0 - z_s) \quad (5.32)$$

The final optimal control action is selected by means of (5.32).

5.10.4 Constraints of the Conflicting Objectives NMPC Controller

The Conflicting Objectives approach needs to address *three* constrained optimization problem, as seen in Section 5.10. Then, the three problems must be simplified, otherwise they will be too demanding for a practical implementation.

5.10.5 Hard constraints for Conflicting Objectives NMPC Controller

The three problems will be simpler in terms of constraints with respect to the other two controllers, NMPC tracking-based and EMPC.

Equality constraints

In order to simplify the problem, the equality constraints used in the Conflicting Objectives NMPC controller are the following:

- The hard constraints for the first and the third constrained optimization problem will be the constraints on the dynamics, i.e. those ones about the Euler Discretization method for the numerical integration of the prediction model, as in (5.8).
- Then, the constraint about the first step of the prediction horizon is adopted, exactly in the same way seen in (5.11), both in the first and in the third optimization problem.
- Additionally, the terminal constraints are employed: it is part of the constrained optimization problem, as detailed in [3]. As in the other two con-

trollers, the terminal constraints encompass only the first four states, exactly as (5.10). Once again, it present in the first and in the third optimization problem.

For the second optimization problem is a different story, it is the most important part of the approach. The maximization problem is not performed on a prediction horizon, but only on a single step, so the equality constraints regarding the dynamical model, i.e. the Euler Discretization method, are applied only on the first step $k_p = 0$.

Inequality constraints

Regarding the inequality constraints, they will be:

- The boundaries on the control action, $(T_{EM}^{max}, T_{EM}^{min})$ and $(\delta_f^{max}, \delta_f^{min})$ as seen in (5.12) and (5.13).
- The boundaries on the value of the SOC, as seen in (5.15).
- The aim of the second optimization problem is to determine a better trajectory for the final control action. To guide this process, we introduce constraints on the values of a_x, a_y, P_b , thereby encouraging the solution to remain within acceptable bounds while steering the system along a feasible trajectory.

Notably, the constrained optimization problem for searching a^* is, inherently, simpler with respect to the others, therefore we can add some constraints without weigh it down. So we will have:

$$a_x^{min} \leq a_x(k_p) \leq a_x^{min} \quad (5.33)$$

$$a_y^{min} \leq a_y(k_p) \leq a_y^{min} \quad (5.34)$$

$$P_b^{low} \leq P_b(k_p) \leq P_b^{up} \quad (5.35)$$

Note 5.12

Once again, the constraints on the second optimization problem, (5.33), (5.34), (5.35), are applied only on the first time instant of the prediction horizon, $k_p = 0$.

Moreover, as detailed in 4.1, the diversity constraints on states and inputs must be added, i.e.:

$$(x_0, u_0) \in \mathcal{Z} \setminus (x_s, u_s) \quad (5.36)$$

The expression in (5.36) can be thought as a distance between (x_0, u_0) and (x_s, u_s) , which must be always different from zero. The distance can be computed, component-wise, in the following way:

$$|x_{0i} - x_s| \forall i = 1, \dots, n \text{ and } |u_{0j} - u_s| \forall j = 1, \dots, m \quad (5.37)$$

As said, $|x|$ is not differentiable for $x \rightarrow 0$, but we can employ the same approximation seen in (5.29). Then, (5.36) can be managed as an inequality constraint of the kind:

$$\varepsilon \leq \sqrt{x^2 + \varepsilon} \leq \infty \quad (5.38)$$

for both the states and the inputs.

Numerical approximation to Facilitate the Solver

Once again, some numerical approximation are employed to facilitate the solver during the resolution of the constrained optimization problem. They are exactly the same seen in Section 5.4.

Initial Guess and Convergence tolerance

Once again, an initial guess of the solution must be provided to the solver, and it is a critical choice.

For the first optimization problem, the initial guess can be the previous status of the system, i.e. states and inputs at the time instant $k - 1$.

In the second optimization problem, we have found the trade-off equilibrium

point, and we want to push the entire system towards it or towards its neighbourhood. Then, a reasonable initial guess is the equilibrium trade-off point itself (x_s, u_s) .

In the end, for the last optimization problem, since we want to push the entire system towards the stable area around the point (x_s, u_s) we can choose it as the initial guess.

Regarding the setting of the solver, for the first and the second optimization problem, it is equal to the setting using for the NMPC and the EMPC controller (see Section 5.4.2). Whereas for the last optimization problem, which will decide the final control action, in order to facilitate the solver we can reduce a bit the convergence tolerance (from $1 \cdot 10^{-3}$ to $1 \cdot 10^{-2}$).

5.10.6 Comparison Between Conflicting Objectives NMPC and NMPC

As said, the length of the prediction horizon is an important characteristic for reaching satisfying performance and stability of the controller, then a high value of N_p should be used. On the other hand, increasing N_p leads to greater computational complexity. As in most control design problems, a balance between computational complexity and performance must be carefully considered.

The new procedure is thought to reduce the complexity of the optimization problems to be addressed and at the same time steer the controller to a consistent area of $\mathcal{Z}$, ensuring reliable performance even under challenging conditions.

We can compare the NMPC controller with the Conflicting Objectives controller, in Table 5.16 and Table 5.17 are resumed some results.

Note 5.13

The initial conditions for all the simulation of the Conflicting Objectives controller on Reference Trajectory 001 are the same of the NMPC controller (see Table 5.7).

Note 5.14

Table 5.16 reports the same results of Table 5.8, it is placed here only for a clearer reading.

Once again, Test 9 of Table 5.16 is the benchmark, and we can see that the new procedure for averaging the conflicting objectives loses accuracy in tracking the reference trajectory: an increasing of $0.16m$ in the $\max|e_{ct}|$ is detected in the best case, and an increasing of 0.001 rad (0.057°) in the $\max|e_h|$. They are both negligible differences in large scale, but anyway a decreasing of tracking performance is detected.

The same happens for the energy consumption performance: the NMPC shows a better management of the energy, as we can see in Table 5.18.

On the other hand, we can see in Table 5.19 that the decreasing of the prediction horizon from 22 to 20 generates problem at the NMPC controller: in almost all the tests it cannot manage the all travel, resulting in an Error (i.e. a NaN detected from CasADi). Whereas the procedure for Averaging Conflicting Objectives can exploit a shorter prediction horizon, as shows Table 5.20.

Test	W_{ct}	W_h	W_v	$W_{\Delta_{\delta_f}}$	W_{Δ_a}	W_{Δ_d}	Q	P	$\max e_{ct} [m]$	$\max e_h [rad]$
1	1	1	1	$1 \cdot 10^4$	2	1	I_4	I_4	2,16	0,144
2	10	1	1	$1 \cdot 10^4$	2	1	I_4	I_4	2,14	0,144
3	10	10	1	$1 \cdot 10^4$	2	1	I_4	I_4	2,14	0,144
4	10	25	1	$1 \cdot 10^4$	2	1	I_4	I_4	2,14	0,144
5	100	1	1	$1 \cdot 10^4$	2	1	I_4	I_4	1,98	0,143
6	100	10	1	$1 \cdot 10^4$	2	1	I_4	I_4	1,98	0,143
7	100	25	1	$1 \cdot 10^4$	2	1	I_4	I_4	1,98	0,143
8	100	25	1.5	$1 \cdot 10^4$	2	1	I_4	I_4	1,98	0,143
9	100	25	2	$1 \cdot 10^4$	2	1	I_4	I_4	1,98	0,143

Table 5.16: Tuning results of the NMPC controller with $N_p = 22$, on Reference Trajectory 001.

Test	W_{ct}	W_h	W_{pow}	W_{Δ_ζ}	$W_{\Delta_{\delta_f}}$	W_{Δ_a}	W_{Δ_d}	$\max e_{ct} [m]$	$\max e_h [rad]$
1	10	10	1	$1 \cdot 10^4$	$1,5 \cdot 10^4$	2	1	2,17	0,144
2	10	10	10	$1 \cdot 10^4$	$1,5 \cdot 10^4$	2	1	2,17	0,144
3	10	10	25	$1 \cdot 10^4$	$1,5 \cdot 10^4$	2	1	2,17	0,144
4	10	10	1	$1 \cdot 10^5$	$1,5 \cdot 10^4$	2	1	2,17	0,144
5	10	10	10	$1 \cdot 10^5$	$1,5 \cdot 10^4$	2	1	2,17	0,144
6	10	10	25	$1 \cdot 10^5$	$1,5 \cdot 10^4$	2	1	2,17	0,144
7	10	10	1	$1 \cdot 10^6$	$1,5 \cdot 10^4$	2	1	2,17	0,144
8	10	10	10	$1 \cdot 10^6$	$1,5 \cdot 10^4$	2	1	Error	Error
9	10	10	25	$1 \cdot 10^6$	$1,5 \cdot 10^4$	2	1	Error	Error
10	25	25	10	$1 \cdot 10^4$	$1,5 \cdot 10^4$	2	1	2,14	0,144
11	25	25	10	$1 \cdot 10^5$	$1,5 \cdot 10^4$	2	1	2,14	0,144
12	25	25	10	$1 \cdot 10^6$	$1,5 \cdot 10^4$	2	1	Error	0,144

Table 5.17: Tuning results of the controller with the conflicting objectives, $N_p = 22$, on Reference Trajectory 001, tracking focus.

Test	W_{ct}	W_h	W_{pow}	W_{Δ_ζ}	$W_{\Delta_{\delta_f}}$	W_{Δ_a}	W_{Δ_d}	SOC saved [%]
1	10	10	1	$1 \cdot 10^4$	$1,5 \cdot 10^4$	2	1	-0,0047
2	10	10	10	$1 \cdot 10^4$	$1,5 \cdot 10^4$	2	1	-0,0043
3	10	10	25	$1 \cdot 10^4$	$1,5 \cdot 10^4$	2	1	-0,0045
4	10	10	1	$1 \cdot 10^5$	$1,5 \cdot 10^4$	2	1	-0,0047
5	10	10	10	$1 \cdot 10^5$	$1,5 \cdot 10^4$	2	1	-0,0166
6	10	10	25	$1 \cdot 10^5$	$1,5 \cdot 10^4$	2	1	-0,0095
7	10	10	1	$1 \cdot 10^6$	$1,5 \cdot 10^4$	2	1	-0,0047
8	10	10	10	$1 \cdot 10^6$	$1,5 \cdot 10^4$	2	1	Error
9	10	10	25	$1 \cdot 10^6$	$1,5 \cdot 10^4$	2	1	Error
10	25	25	10	$1 \cdot 10^4$	$1,5 \cdot 10^4$	2	1	-0,0100
11	25	25	10	$1 \cdot 10^5$	$1,5 \cdot 10^4$	2	1	-0,0453
12	25	25	10	$1 \cdot 10^6$	$1,5 \cdot 10^4$	2	1	Error

Table 5.18: Tuning results of the controller with the conflicting objectives, $N_p = 22$, on Reference Trajectory 001, energy consumption focus

Test	W_{ct}	W_h	W_v	$W_{\Delta_{\delta_f}}$	W_{Δ_a}	W_{Δ_d}	Q	P	$\max e_{ct} [m]$	$\max e_h [rad]$
1	1	1	1	$1 \cdot 10^4$	2	1	I_4	I_4	Error	Error
2	10	1	1	$1 \cdot 10^4$	2	1	I_4	I_4	Error	Error
3	10	10	1	$1 \cdot 10^4$	2	1	I_4	I_4	3,13	0,292
4	10	25	1	$1 \cdot 10^4$	2	1	I_4	I_4	37,3	0,385
5	100	1	1	$1 \cdot 10^4$	2	1	I_4	I_4	Error	Error
6	100	10	1	$1 \cdot 10^4$	2	1	I_4	I_4	Error	Error
7	100	25	1	$1 \cdot 10^4$	2	1	I_4	I_4	Error	Error
8	100	25	1.5	$1 \cdot 10^4$	2	1	I_4	I_4	4,77	0,358
9	100	25	2	$1 \cdot 10^4$	2	1	I_4	I_4	Error	Error

Table 5.19: Tuning results of the NMPC controller with $N_p = 20$, on Reference Trajectory 001, tracking focus.

Test	W_{ct}	W_h	W_{pow}	W_{Δ_ξ}	$W_{\Delta_{\delta_f}}$	W_{Δ_a}	W_{Δ_d}	$\max e_{ct} [m]$	$\max e_h [rad]$
1	10	10	1	$1 \cdot 10^4$	$1,5 \cdot 10^4$	2	1	1,82	0,128
2	10	10	10	$1 \cdot 10^4$	$1,5 \cdot 10^4$	2	1	1,82	0,128
3	10	10	25	$1 \cdot 10^4$	$1,5 \cdot 10^4$	2	1	1,82	0,128
4	10	10	1	$1 \cdot 10^5$	$1,5 \cdot 10^4$	2	1	1,82	0,128
5	10	10	10	$1 \cdot 10^5$	$1,5 \cdot 10^4$	2	1	1,82	0,128
6	10	10	25	$1 \cdot 10^5$	$1,5 \cdot 10^4$	2	1	Error	Error
7	10	10	1	$1 \cdot 10^6$	$1,5 \cdot 10^4$	2	1	1,82	0,128
8	10	10	10	$1 \cdot 10^6$	$1,5 \cdot 10^4$	2	1	1,82	0,128
9	10	10	25	$1 \cdot 10^6$	$1,5 \cdot 10^4$	2	1	Error	Error
10	25	25	10	$1 \cdot 10^4$	$1,5 \cdot 10^4$	2	1	1,80	0,128
11	25	25	10	$1 \cdot 10^5$	$1,5 \cdot 10^4$	2	1	1,80	0,128
12	25	25	10	$1 \cdot 10^6$	$1,5 \cdot 10^4$	2	1	Error	Error
13	10	10	10	$1 \cdot 10^4$	$1,5 \cdot 10^4$	2	2	1,83	0,128
14	10	10	10	$1 \cdot 10^4$	$1,5 \cdot 10^4$	2	5	Error	Error
15	5	5	5	$1 \cdot 10^4$	$1,5 \cdot 10^4$	2	10	1,93	0,132
16	2	2	2	$1 \cdot 10^4$	$1,5 \cdot 10^4$	2	10	1,93	0,132

Table 5.20: Tuning results of the controller with the conflicting objectives, $N_p = 20$, on Reference Trajectory 001, tracking focus.

Unfortunately, another lack arises when we decrease N_p in the performance of the Conflicting Objectives controller: the minimum safety distance is never respected. This situation is likely related to the method used to compute the minimum safety distance: it is thought as the distance spaced in 1 s by the vehicle, and $N_p = 20$ is exact a prediction of 1 s in the future. Being exactly on the limit of the predicted time does not allow for finding an optimal control action that can respect the soft constraint on the safety distance. For this reason, it does not make any sense to shorten the prediction horizon further. In Figure 5.12 is shown the behaviour of the distance between the vehicles, we cannot increase the distance even if we reduce the weight of the tracking part and increase the weight of the Safety Distance error (Test 16 of Table 5.20).

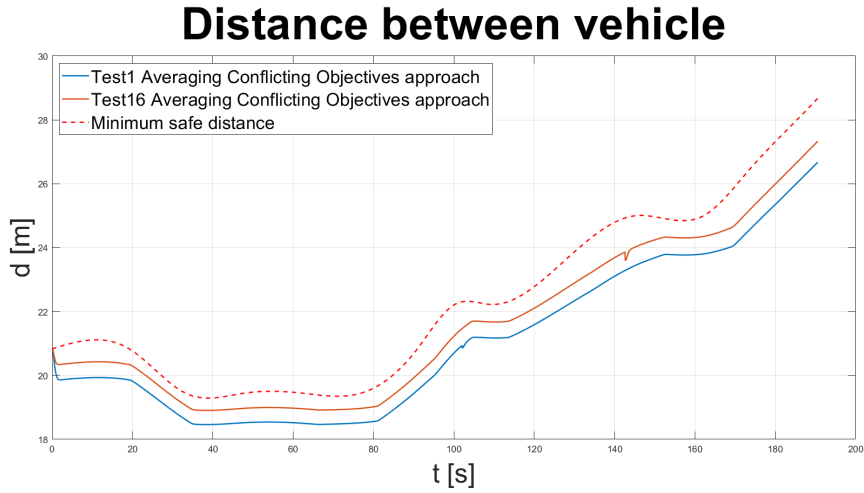


Figure 5.12: Distance between vehicle for the Averaging Conflicting Objectives approach (Tests Table 5.20) with $N_p = 20$, on Reference Trajectory 001.

5.10.7 Control Inputs behaviour in decreasing Prediction Horizon length for Conflicting Objectives NMPC

Reducing the prediction horizon from 22 steps to 20 steps does not affect in a significant way the tracking performance of the Conflicting Objectives NMPC controller, but the controller must be reliable also in the selection of a control action. That means we should have a smooth behaviour of the T_{EM} and δ_f in

order to guarantee a comfortable travel. Test 10 and Test 11 are the best from the tracking point of view in both cases, $N_p = 22$ and $N_p = 20$ for the Conflicting Objectives NMPC. In Figure 5.13 and Figure 5.14 we can see how the shortening of the prediction horizon leads to smother behaviour of the control action, i.e. the controller does not need an aggressive control action even if the prediction is shorter. It is not trivial as result: if we can predict far in the future we can be able to choose a better trajectory to follow, but in this case the using of neighbourhood of the trade-off equilibrium point as area to which we want to approach brings benefit. The controller is pushed towards a good trajectory almost immediately after the system leaves the reference trajectory, in this way it can use less steps to steer the plant to the path to follow. In conclusion, a less aggressive control action is needed with a short prediction horizon, computed in a less complex framework: we do not forget that the reduction of the steps of the prediction horizon simplifies the optimization problem.

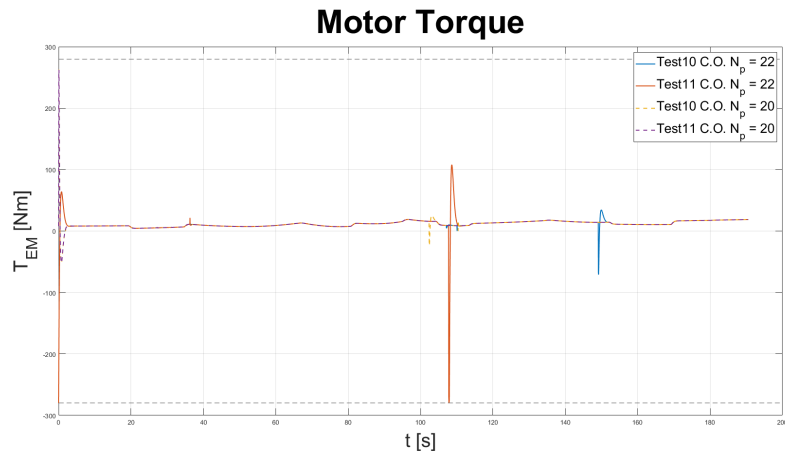


Figure 5.13: T_{EM} behaviour for tests in Table 5.20 and Table 5.17, on Reference Trajectory 001.

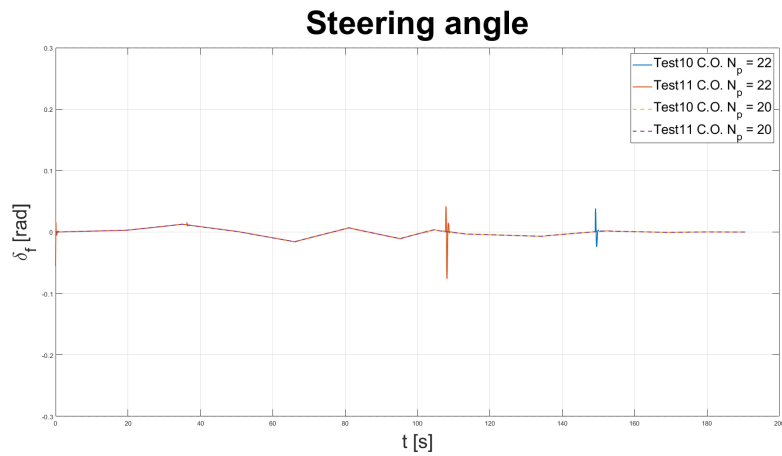


Figure 5.14: δ_f behaviour for tests in Table 5.20 and Table 5.17, on Reference Trajectory 001.

Chapter 6

Conclusion and Further Developments

This work focused on the designing and the implementation of a NMPC controller, in particular the most interesting approach was found in an *economic* form. The study has explored mainly the evolution from a NMPC to an EMPC controller in order to face the problem of the longitudinal-lateral dynamics of a vehicle, keeping as objective a reduction of energy consumption, meanwhile the safety and comfort of the passengers are unchanged. The lateral and longitudinal dynamics of moving vehicle involve two known systems: ACC and LKA. Through methodical testing the interaction between this two systems, with some little differences with respect their classic form, in different scenarios is explored.

The progressive increasing complexity of the driving conditions in which the controller was found are managed by the MPC method in both approaches, tracking and economic, in a satisfying way. Meaningful differences are noticed between the tracking and the economic formulation of the problem: even if the tracking performances are less accurate in the economic approach, and notably the lack of accuracy is negligible in large scale, the energy saving is improved.

The results demonstrate the initial idea: even if the tracking performances are not the main focus of the problem, they can be obtained as consequence of the economic approach. This shift in perspective allows for a focused analysis of energy consumption, which is currently one of the most critical challenges in all

branches of the transportation sector.

Particularly significant are the results on the distance between vehicle during the travel. Any passengers will take as main safety issue the distance from the vehicle ahead, and with both approaches this distance is widely respected, we can see in Figure 5.9 how Test 2 of the economic approach is the best controller on this aspect.

Secondly, also the smoothness of the driving is reached. Intuitively, one can expect that as the deviation from the reference trajectory increases, the controller must act to steer the system back onto the desired path. However, the plots in Chapter 5 clearly show that when either the Cross-Track error or the Heading error increases, their return to the ideal value of zero follows a linear evolution. Notably, there are no abrupt changes or successive spikes in these indices. Moreover, the Steering Angle error is close to zero for the majority of the travel, reflecting the rarest case in which an aggressive change of direction of the heading is chosen by the controller as control action, unfortunately is not the same for the torque provided by the EM. Frequently spikes, even with a low intensity, are uncomfortable and the origin of vibrations, but it is the price to pay for having a energy saving in the regions in which the power demands can be lowered down.

As said, the computational effort is not a side problem of the MPC approach, but the simplification of the cost function resulting in the EMPC approach decreases the average time for building and solving the optimization problem. A decreasing from 0.56 s (NMPC) to 0.42 s (EMPC) is reached, the 25% of time is cut off.

Another interesting development of the work is the employing of a novel approach for the design of a stabilizing term. The procedure was too demanding for the complex lateral-longitudinal dynamics we have faced, in fact the using of the new approach detailed in [3] does not lead to satisfying results. However, a different employing of the design method leads to a slightly increasing of the reliability of the system in different conditions with respect to the NMPC approach: the shortening of the prediction horizon is a crucial difference from the tracking NMPC that does not lead to significant issues. Although accuracy is lost, albeit on a negligible scale, when using the Averaging Conflicting approach, the resulting performances seem to be promising for a future development of this procedure.

6.1 Further Developments

Further study could focus on the response of the model in more complex scenarios, for example the effect of the variations in road inclination can help the EMPC controller to lower down the energy consumptions: in downhill situation the braking torque used to control the longitudinal speed will be an important resource. Additionally, road irregularities, burst of wind both in lateral and longitudinal direction can have a significant impact on the control action and so on the controller performances: they are necessary considerations to be evaluated in a real-world implementation prospective.

We have seen as every conversion of energy (mechanical to electric or vice-versa, DC to AC or vice-versa), unfortunately, is characterized by a loss of energy, then in situation in which there are more conversions an economic approach could be interesting. For instance, the hybrid vehicles use two different sources of energy, and their interaction is different based on the configuration of the HEV. In this complex situation, saving energy can be enormously precious.

Even if the economic approach lowers down the computational effort, and simplify the problem, the complex lateral dynamics of a vehicle force a quite high length of the prediction horizon in order to reach satisfying results. In a real-world application it would be important decreasing the average time needed for computing the control action as much as possible. An average computational time lower than the sampling time should be the goal to address. This represents a crucial research area, as it enables the validation of the economic approach under realistic ground test conditions: the feasibility and robustness of the method must be evaluated outside the simulations environment.

Another interesting area of research could be the integration of other ADAS with an Economic approach. This work did not consider the presence of other vehicles, despite the leading vehicle, but in the real world the presence of multiple nearby vehicle is common. The detection of those vehicles can be performed by the BSM, and its features could be integrated in the dynamics of the car, which should, obviously, take care of nearby vehicle in deciding the control action.

The novel approach of Averaging Conflicting Objectives could be further investigated, particularly for complex system where the original method is unman-

Conclusion and Further Developments

ageable. An interesting theoretical aspect to study is the stability property related to the additional term, the simulations in this work show how it can be a viable and practical solution for employing the procedure.

Bibliography

- [1] David Angeli, Rishi Amrit, and James B Rawlings. “On average performance and stability of economic model predictive control”. In: *IEEE transactions on automatic control* 57.7 (2011), pp. 1615–1626.
- [2] Borhan, Hoseinali and Vahidi, Ardalan and Phillips, Anthony M and Kuang, Ming L and Kolmanovsky, Ilya V and Di Cairano, Stefano. “MPC-based energy management of a power-split hybrid electric vehicle”. In: *IEEE Transactions on Control Systems Technology* 20.3 (2011), pp. 593–603.
- [3] Lorenzo Calogero et al. “Averaging Conflincting Objectives in Economic Nonlinear MPC for Adaptive Cruise Control”. In: *In press* (2025).
- [4] Lorenzo Calogero et al. “Neural Adaptive MPC with Online Metaheuristic Tuning for Power Management in Fuel Cell Hybrid Electric Vehicles”. In: *IEEE Transactions on Automation Science and Engineering* (2025).
- [5] Embitel Technologies. *How Do VCU and MCU in Electric Vehicles Function Hand in Hand?* <https://www.embitel.com/blog/embedded-blog/how-do-vcu-and-mcu-in-electric-vehicles-function-hand-in-hand>. Accessed: 2025-07-02. 2022.
- [6] Eurostat. *Passenger cars in the EU*. https://ec.europa.eu/eurostat/statistics-explained/index.php?title=Passenger_cars_in_the_EU. Accessed: 2025-07-02. 2024.
- [7] EVBoosters. *On-Board Charger*. <https://evboosters.com/ev-charging-academy/ev-charging-vocabulary/on-board-charger/>. Accessed: 2025-07-06. 2024.
- [8] FCA Italy S.p.A. *Nuova Fiat 500 owner’s manual*. https://aftersales.fiat.com/eLumData/IT/00/332_NUOVA500/00_332_NUOVA500_603.85.158_IT_03_08.20_L_LG/00_332_NUOVA500_603.85.158_IT_03_08.20_L_LG.pdf. Accessed: 2025-07-05. 2020.
- [9] Antonio Ferramosca. *Foundamentals of Model Predictive Control: Economic MPC*. Lectures Slides, Università degli Studi di Bergamo. 2022.

- [10] Grüne, Lars and Pannek, Jürgen. *Communication and Control Engineering. Nonlinear model predictive control: Theory and Algorithms*. 2nd ed. Springer, 2017.
- [11] Joel Andersson, Joris Gillis. *CasADi Documentation. Release 3.7.0*. <https://web.casadi.org/docs/>. Accessed: 2025-07-06. 2025.
- [12] Zifei Nie and Hooman Farzaneh. “Adaptive cruise control for eco-driving based on model predictive control algorithm”. In: *Applied Sciences* 10.15 (2020), p. 5271.
- [13] Carlo Novara. “L02 PID - Driver assistance system design A: Proportional Integral Derivative control”. In: *Lecture Slides* (2025).
- [14] S. Onori, L. Serrao, and G. Rizzoni. *Hybrid Electric Vehicles: Energy Management Strategies*. Springer, 2016.
- [15] European Parliament. “Regulation (EU) 2023/851 of the European Parliament and of the Council of 19 April 2023 amending Regulation (EU) 2019/631 as regards strengthening the CO₂ emission performance standards for new passenger cars and new light commercial vehicles in line with the Union’s increased climate ambition”. In: *Official Journal European Union* 110 (2023), pp. 5–20.
- [16] Pereira, Derick Furquim and da Costa Lopes, Francisco and Watanabe, Edson H. “Nonlinear model predictive control for the energy management of fuel cell hybrid electric vehicles in real time”. In: *IEEE Transactions on Industrial Electronics* 68.4 (2020), pp. 3213–3223.
- [17] G.C. Goodwin. M.M. Seron and J.A. de Doná. *Constrained control and estimation: An optimisation approach*. Springer, 2005.
- [18] The MathWorks, Inc. *MATLAB*. Version 24.2.0.2712019 (R2024b). Version 24.2.0.2712019 (R2024b). Mar. 2024.
- [19] Xin, Weiwei and Xu, Enyong and Zheng, Weiguang and Feng, Haibo and Qin, Jirong. “Optimal energy management of fuel cell hybrid electric vehicle based on model predictive control and on-line mass estimation”. In: *Energy Reports* 8 (2022), pp. 4964–4974.
- [20] Haohua Zhang. “Enhancing vehicle safety-the role of PID control in adaptive cruise control systems”. In: *AIP Conference Proceedings*. Vol. 3194. 1. AIP Publishing LLC. 2024, p. 050023.