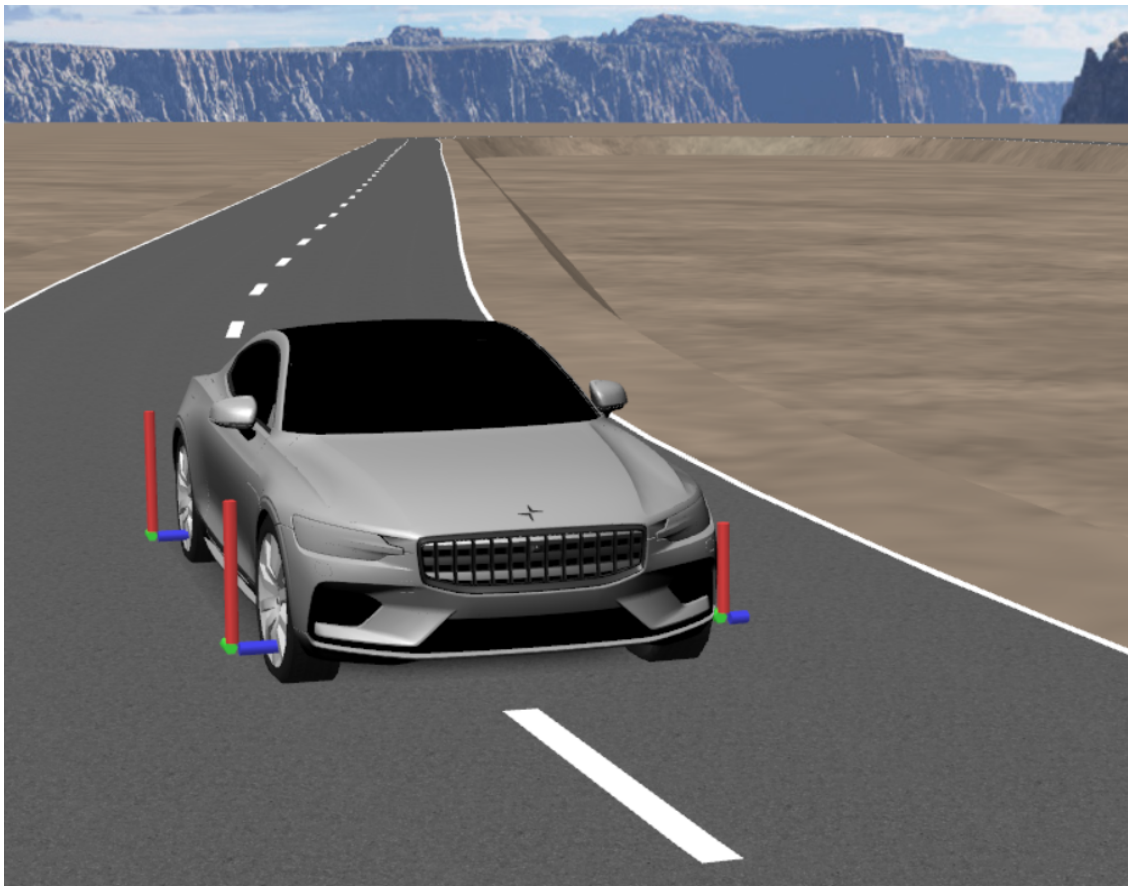
CHALMERS
UNIVERSITY OF TECHNOLOGY



Energy Efficient Lateral Motion Control for Future Electric Vehicles

Master's thesis in Systems, Control and Mechatronics

Alexandre Vieira da Rocha
Daniel Alejandro Poveda Pi

Department of Electrical Engineering
CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden 2023
www.chalmers.se

Master's thesis 2023

Energy Efficient Lateral Motion Control for Future Electric Vehicles

Alexandre Rocha
Daniel Poveda

CHALMERS
UNIVERSITY OF TECHNOLOGY

Department of Electrical Engineering
Chalmers University of Technology
Gothenburg, Sweden 2023

Energy Efficient Lateral Motion Control for Future Electric Vehicles
Alexandre Rocha & Daniel Poveda

Supervisors: Derong Yang, Anand Ganesan, Juliette Torinsson
Examiner: Nikolce Murgovski, Mats Jonasson

Master's Thesis 2023
Department of Electrical Engineering
Chalmers University of Technology
Gothenburg, Sweden 2023
SE-412 96 Gothenburg
Telephone +46 31 772 1000

Typeset in LaTeX
Printed by Chalmers Reproservice
Gothenburg, Sweden 2023

Energy Efficient Lateral Motion Control for Future Electric Vehicles
Alexandre Rocha & Daniel Poveda
Department of Electrical Engineering
Chalmers University of Technology
Gothenburg, Sweden 2023
SE-412 96 Gothenburg
Telephone +46 31 772 1000

Abstract

The automotive industry is facing one of its most exciting technological turning points. On one hand, manufacturers are electrifying their fleets driven by environmental concerns, regulation, and economic incentives. On the other hand, future electric vehicles (EV) are expected to exploit various sensors concerning environment perception and vehicle motion sensing, as well as different actuator configurations, where even more control degree of freedom is added due to the advent of autonomous driving. Nevertheless, EVs still suffer from a major limitation when compared with internal combustion vehicles: the energy density, and consequently the driving range. The integration of autonomous capabilities in EVs, coupled with the need for energy-efficient vehicle motion control, suggest energy-optimal trajectory planning as a promising technology to address this challenge. While energy-efficient longitudinal control technologies such as ECO-cruise control are already mature and widely implemented, the combination of lateral and longitudinal control is still an active area of research. In this thesis, nonlinear and economic model predictive control is applied to minimize the energy consumption of an EV given a reference path and longitudinal velocity profile. The focus of the research is on achieving efficient, robust, and real-time computational performance. To this end, the vehicle model is simplified, and the testing conditions are constrained to low acceleration limits. High-fidelity vehicle simulations using the IPG CarMaker software are conducted on a test track. The relation between lateral and longitudinal control and savings in energy consumption is explored, in order to conclude on what is the energy consumption reduction potential from the control perspective. In specific, the displacement from path's center line and the reference longitudinal velocity are examined individually and in combination to determine their influence on energy savings. Results show that, for limited accelerations, the greatest potential lie on optimizing longitudinal control, where significant energy savings can be attained at the expense of limited decrease in average longitudinal velocity. However, the intricate trade-off between the economic cost, the energy consumption, and the reference tracking cost in the MPC formulation led to limited energy savings when moving from path tracking to trajectory planning.

Keywords: model predictive control, economic model predictive control, energy minimization, electric vehicle, autonomous driving

Acknowledgements

We would like to express our gratitude to Chalmers University of Technology and Volvo Cars Cooperation for the opportunity to research the interesting topic of energy-efficient autonomous driving of electric vehicles and contribute to the development of sustainable and intelligent vehicle technology.

During this period, we had the chance to meet and discuss with highly knowledgeable people who always provided us with insightful feedback, comments, and ideas. We learned immensely from all of you who contributed to our project and dedicated some of your time to discuss with us, and we greatly appreciate it. Namely, at Volvo Cars, we would like to start by thanking Dr. Derong Yang for her knowledgeable contributions in every discussion, her never-ending ambition to continuously improve, and her out-of-the-box mindset, which was critical in guiding us along the way and ensuring we always followed a product-oriented research direction. Our co-supervisors, Juliette Torinsson and Anand Ganesan, played an equally important role in this project, acting as structural pillars that supported us during the difficult moments of the thesis. Your insights, questions, and advice were essential in guiding us and solving the challenges we faced. Finally, we also thank our manager, Anna Söderlund, for believing in us and allowing us to be part of Volvo Cars. At Chalmers, our academic supervisors, Nikolce Murgovski and Mats Jonasson, provided continuous support and were always available to discuss any doubts or ideas with us. We thank both of you for your guidance, tutorship, patience, time, and lastly, your motivation on this project.

We finished this thesis as better individuals than when we started. It was a pleasure to work at Volvo Cars, especially with all of you.

Alexandre Rocha and Daniel Poveda, Gothenburg, June 2023

The end of this thesis represents the conclusion of six years of education. To arrive here, I could not have made it without the ceaseless support from my parents, who taught me the values of work, perseverance, and dedication. They always believed in me and made me believe I could succeed in the challenges I set for myself. I would also like to thank all my friends in Portugal for enriching my life with special moments and helping me achieve what I have achieved. It is beautiful to look back and compare it to where we are now and see how we have contributed to each other's paths. To my friends in Sweden and all around the world, I thank you immensely for these past two years. This experience has shaped my life, and you are now a part of me. You have made me a better person, and I hope I have had an impact on you as well. To Anna, I thank you for your constant support, care, and for making my life richer and happier. During these six months, you have helped me surpass difficult moments in my life, giving me the emotional structure I needed to keep my head up through the challenges I faced. Thank you for making me as well a better person, and I hope you continue to do so every day. Finally, I would like to thank João Sousa, Miguel Aguiar, Torsten Wik, and Nikolce Murgovski, for the opportunities you gave me throughout these years and for your remarkable tutorship. You all played important and special roles in my education and I can only be grateful for that. To all of you,

Obrigado.

Alexandre Rocha, Gothenburg, June 2023

To my mom and my dad, thank you for everything you've done for me since day one. Thank you for always supporting me in ways that cannot be measured or quantified. I hope I can turn back all you've done for me.

To Gilberto Espinoza, this is just a very tiny proof of how big the impact you've had (and currently have) in people through your kindness, generosity and fun way of teaching. You literally offered me help when nobody else did, you truly cared. I'll be forever grateful to you.

To my beloved Kirsten, thank you for being there and take care of me not only for the last 5 years, but since the day we met 15 years ago. It's been fun, te amo chiquita.

To my brother Jonathan, thank you for being there, I love you.

To all of you, I'll never be able to thank you enough for all you've done to help me fulfill a goal that has been a dream since I was a little kid.

Daniel Poveda, Gothenburg, June 2023

List of Acronyms

Below is the list of acronyms that have been used throughout this thesis listed in alphabetical order:

EMPC	Economic Model Predictive Control
EV	Electric Vehicle
IPM	Interior Point Method
LQR	Linear Quadratic Regulator
MPC	Model Predictive Control
MAE	Mean Absolute Error
NLP	Nonlinear Programming Problem
NMPC	Nonlinear Model Predictive Control
OCP	Optimal Control Problem
ODE	Ordinary Differential Equation
PPC	Pure Pursuit Control
RTI	Real Time Iteration
SQP	Sequential Quadratic Programming

Nomenclature

Below is the nomenclature of indices, sets, parameters, and variables that have been used throughout this thesis.

x^I, y^I, ψ	Vehicle coordinates and orientation in inertial frame
x_s, y_s, ψ_s	Path coordinates and orientation in inertial frame
v_x	Vehicle longitudinal velocity in vehicle frame
v_y	Vehicle lateral velocity in vehicle frame
r	Yaw rate
s	Distance travelled along reference path
d	Vehicle lateral displacement from reference path
$\Delta\psi$	Vehicle heading deviation from reference path
κ	Curvature of the reference path
F_x	Resultant longitudinal force in vehicle frame
F_y	Resultant lateral force in vehicle frame
M_z	Resultant yaw moment at vehicle center of mass
$i = \{f, r\}$	Axle index where f and r stand for front and rear respectively
$F_{x,i}$	Longitudinal tire force at axle i
$F_{y,i}$	Longitudinal tire force at axle i
$F_{z,i}$	Normal tire force at axle i
$F_{rr,i}$	Rolling resistance tire force at axle i
F_{aero}	Aerodynamic drag force at vehicle center of mass
$\sigma_{x,i}$	Longitudinal tire slip ratio at axle i
$\sigma_{y,i}$	Lateral tire slip ratio at axle i
T	Total vehicle propulsion torque
δ	Steering angle
m	Vehicle mass
g	Gravity force
l_i	Distance from axle i to vehicle center of mass
I_{zz}	Rotational inertia around z axle of vehicle frame
I_w	Rotational wheel inertia
w_i	Rotational speed of wheel at tire i
$C_{\sigma_{x,i}}$	Longitudinal tire stiffness at axle i
$C_{\sigma_{y,i}}$	Lateral tire stiffness at axle i
r_0	Undeformed wheel radius
ρ	Air density

c_D	Aerodynamic drag coefficient
A_f	Vehicle frontal area
μ	Road tire-friction coefficient
P	Total power consumed by the vehicle electric machines
P_{mech}	Outputted mechanical power from the vehicle electric machines
$P_{\text{el.loss}}$	Electric power losses of the vehicle electric machines
P_{inertia}	Inertial power
$P_{\text{tire loss}}$	Tire power losses
P_{aero}	Aerodynamic power
$\mathbf{x}$	State vector in state-space formulation
$\mathbf{u}$	Control input vector in state-space formulation
s, t	Independent variables: space and time
$\dot{\mathbf{x}}$	Time derivative of $\mathbf{x}$
$\mathbf{x}'$	Space derivative of $\mathbf{x}$
f	System continuous dynamics in time domain
f_d	System discrete dynamics in time domain
f_s	System discrete dynamics in space domain
Δs	Discretization interval in distance
S_f	Look-ahead distance
N	MPC prediction horizon length
Δv	Deviation from constant (raw) reference speed
$\bar{v}_x(s)$	Saturated reference longitudinal velocity
$\bar{V}_x$	Constant (raw) reference longitudinal velocity

Contents

List of Figures

List of Tables

1
Introduction

1.1 Motivation

The automotive industry is undergoing a phase of disruptive transformation imposed by economic, regulatory, and sustainability drivers. From the total global emissions of carbon dioxide (CO_2), the transportation sector is responsible for around one-fifth of those, while road passenger vehicles account for around half of that share, in front of road freight transportation, shipping and aviation [30]. Reduction of CO_2 is well proven to have a positive impact on the urgent climate crisis the world is living today, which makes the transportation sector a clear target of emission-reduction policies and incentives. Such policies and incentives are normally in the form of regulation and economic measures to enhance the development of promising cleaner and more efficient technologies, namely vehicle electrification. In response to that, automotive manufacturers are actively shifting their research and development focus to electric vehicles (EVs). Volvo Cars Corporation, where this thesis is written at, has the particularly ambitious goal to become fully electric until 2030 [40].

In this exciting industry, not only vehicles are being electrified, which is by itself a considerable change on what are the standards on vehicle design and manufacture, but also the concept of what is a road passenger vehicle is rapidly evolving. Based on the drastic increase of computational power of embedded systems, future EVs are expected to exploit various sensors concerning environment perception and vehicle motion sensing, as well as different actuator configurations concerning electric motors and active chassis. Such increase in degrees of freedom of vehicle control is responsible for the advent of autonomous driving; once vehicles have enough online processing power, they are expected to replace the human driver in the perception and decision-making tasks. Nevertheless, while the human driver has limited capacity of deciding on control actions that optimize the performance of the vehicle, control actuation sequences can be computed to do so according to different performance objectives, such as energy-consumption, vehicle stability, and time travelled; commonly, multiple objectives should be considered for a safe and efficient ride. Among different objectives, while vehicle stability has to be guaranteed by all means, efficient energy consumption is within the set of main challenges of current vehicle development. The aforementioned EVs proliferation not only comes with advantages but also raises many challenges, namely driving range. Although very important and noticeable developments are being achieved in the battery field [15], the energy-density of fossil fuels is still difficult to overcome. This in turn limits the driving range of EVs even if their overall energy-production cycle efficiency is quite higher than internal combustion engines.

Combining the two trends described in each of the previous paragraphs, this thesis focuses on the problem of energy-optimal trajectory planning of an EV. Related work include different strategies to compute the control actuation online based on the perception from vehicle sensing.

For longitudinal vehicle control, methods such as cruise-control have been developed and are now widely implemented in commercial vehicles [23][38]. In this specific example, the vehicle's longitudinal speed is controlled according to different objectives such as energy-saving or simply by maintaining a certain distance to the vehicle ahead, commonly referred to as eco-cruise and adaptive cruise control, respectively. Similarly, for vehicle lateral control, methods such as torque vectoring split the total torque among electric motors optimize vehicle dynamics enhancing stability [33] and energy efficiency [27]. Other common example of vehicle lateral control is the pure pursuit algorithm to follow a reference trajectory [11], nowadays already implemented for lane-following assist maneuvers. The joint problem of longitudinal and lateral control is however a more complex problem, given the large dimension of the set of feasible control actuation sequences and its redundancy. Moreover, such controllers need to operate at a fast enough rate and should be robust to uncertainties and unexpected scenarios.

1.2 Problem definition

This thesis addresses the problem of energy-optimal trajectory planning of an electric vehicle, given a road profile, i.e. curvature, heading, and its lateral bounds, and a reference velocity profile. The control actuators considered are the front-axle steering angle and the total torque request. The performance attribute to optimize is the vehicle motors' total energy consumption, i.e. motor electrical losses and mechanical output power.

The problem is to be approached in a predictive manner, by formulating an optimal control problem (OCP) to be solved iteratively over a certain finite horizon. Specifically, model predictive control (MPC) is the algorithm to be used [29]. This is motivated by its demonstrated potential on trajectory planning applications due to its ability on predicting future trajectories based on a dynamic model and including constraints on the vehicle state and input sets. Emphasis will be placed on time-efficient computations, aiming to derive algorithms that can run on real-time processors. Therefore, an efficient and robust software and control architecture must be developed for this purpose. Finally, results are taken from a high-fidelity vehicle dynamics software, IPG CarMaker, allowing for near-reality validation of the developed algorithms.

1.3 Research questions

The research questions this thesis proposes to answer can be summarized as follows:

- How to formulate the MPC problem for path tracking and trajectory planning of an EV, given path boundaries and its center line curvature profile?
- How to formulate the MPC problem for trajectory planning of an EV, for energy minimization? What is the decrease in energy consumption of taking the energy-optimal path?
  - How is the tuning of economic MPC resolved when energy and tracking objective terms are considered simultaneously?
  - What is the trade-off between energy efficiency and tracking of longitudinal velocity and lane positioning?
- To what extent is it possible to enhance persistent feasibility in the proposed MPC? How should the problem be formulated to increase feasibility?

- How to efficiently implement MPC in a vehicle, guaranteeing realistic computational times? How should the MPC be tuned to attain efficient computation times, e.g. in terms of the sampling frequency, prediction/control horizon, vehicle model complexity, etc?
- What physical phenomena are important to model in the internal controller to mimic the behavior of a High Fidelity model (e.g., CarMaker) for the energy optimal problem?

1.4 Thesis outline

The thesis is organised as follows. Section 2 presents a detailed review on work related to this thesis and highlights its novelty and scope. Section 3 describes theory on numerical optimal control, transcription methods and numerical solvers for nonlinear programs, and section 4 presents the vehicle and energy consumption modelling. Previous to presenting results, Section 5 presents a general formulation of arbitrary road profiles and test scenarios. Then, Section 6 presents simulations results of the simplified problem of path-tracking, where the vehicle objective is to track the center of the track, followed then by the counterpart on energy-optimal trajectory planning, where the incentive on tracking the center of the path is relaxed. Finally, Section 7 concludes the thesis.

2

Literature review

Literature on vehicle motion control is extensive. Until recent years, the great majority of related work have been in the scope of vehicle stability control, such as cruise-control and anti-lock braking, for vehicle longitudinal control, and electronic stability control, for the lateral dynamics counterpart. Lately, as the computational power of in-vehicle computers increases and EVs gain traction in the market, there has been an increase in the number of technologies in development for vehicle active control, both for vehicle stability control and energy consumption minimization. Moreover, in those, autonomous driving scenarios are increasingly more common.

For human driving scenarios, such algorithms for active vehicle control can be predictive, i.e. can estimate the future trajectory of the vehicle at a certain state based on dynamic vehicle models and state and input constraints, and thus can optimise the control action in real time according to a predefined performance index [8]. For this reason, strategies such as MPC have been increasingly used in related literature. The authors of [32] presented a MPC approach for the problem of vehicle stabilization at near the limit of handling. Based on the vehicle state and radius of curvature, understeer characteristics references are tracked using MPC, that uses longitudinal rear slip ratios as inputs. These slips, in turn, serve as a reference for a sliding mode controller responsible for torque allocation. This architecture is shown to improve stability in two safety-critical scenarios when compared to LQR and equal-distribution torque allocation strategies. The same authors extended the work for the case of nonlinear MPC in [33] and, testing in the same scenarios as in the previous paper, proved that a more accurate nonlinear vehicle model improves vehicle dynamic behaviour. Analogously, in [35], engineers at the vehicle manufacturer Rimac proposed a linear time variant MPC for yaw angle and rate tracking and drift minimising. Once again, the method proved its potential in enhancing vehicle stability. In overall, all these works show that predictive approaches outperform the non-predictive counterpart.

With the advent of autonomous driving, the steering angle and propulsion forces can be as well optimised, thus originating the problems of path planning and/or tracking. A common scope for autonomous driving applications is obstacle avoidance, which can be triggered in non-autonomous driving when suddenly sensing an obstacle in the vehicle predicted trajectory. The work in [20] addresses this problem using NMPC for both trajectory planning and torque vectoring, optimizing steering angle and engine propulsion torque. Obstacles are defined as constraints in the MPC formulation and the controller minimises the distance from the center of the track while respecting the feasible state set. In [18], the authors developed and implemented active steering angle control for vehicle stabilization applied to the double-lane change maneuver test. Results demonstrate that with such predictive approach the vehicle is capable of negotiating the maneuver with higher entry speeds than without the predictive characteristic. The method was tested in a real system and the computational aspects of the implementation are thoroughly described, where it becomes clear that its burden is very high. More recently,

[24] proposed nonlinear MPC (NMPC) to optimise steering angle and rear braking torque vectoring to minimise the time needed for the lateral displacement of a vehicle in the presence of a obstacle. Results show that a faster displacement can be attained when combining braking torque vectoring and steering angle together. In [17], instead, iterative linearization is used, i.e. LTV-MPC, for control of steering angle and slip ratios of each wheel. At a lower level, slip controllers then regulate the individual torque of each wheel. The developed method was compared with an only-negative braking slip counterpart, i.e. braking torque-vectoring, having performed better both in terms of position and velocity tracking. For the same objective, [22] presented a two-layer NMPC architecture although without torque vectoring. At a higher level, spatial NMPC is responsible for trajectory planning, where the translation from time to space domain proved to ease problem formulation. The lower level control is also NMPC, although with a higher sampling rate and a more-detailed vehicle model. Time-sampling is instead used for simpler vehicle implementation. Results proved the success of this approach, which was one of the earliest in the use of space-sampling in MPC. The joint problem of torque vectoring and path planning and tracking has also been studied for racing autonomous vehicles, as in [34]. In this work, a three layer architecture is proposed, organized by the following order: path planning, tracking, and torque vectoring. The proposed architecture enables efficient computation of torque allocation, by using low-level control for this task. The two upper layers are formulated as NMPC strategies. Trajectory planning is in the space domain, offline, and it is aimed to optimise lap-time, including the torque-vectoring-induced yaw moment in the input vector, therefore coupling both problems. Path tracking is conversely time-based and implemented online. Results showed that this architecture allowed for faster laps than with a human driver. This work is inspiring due to its software architecture that simplifies each of the NMPC formulation while still coupling the problems of path planning and tracking and torque vectoring.

Energy-efficiency is another main objective in the development of EVs and autonomous driving is undoubtedly a platform for potential savings in this regard. When including this objective in the problem formulation of the problems referred in the previous paragraphs, its complexity naturally grows and computational and accuracy problems arise. This is, therefore, a still very active research topic. In a non-predictive fashion, and solely addressing the problem of torque vectoring, the works of [12] and [10] both present a rule-based torque vectoring algorithm. This algorithm optimises the overall understeer characteristics based on power losses model of both the powertrain and tire slips. The results are promising showing that the optimisation of understeer characteristics together with a simple set of rule-based strategies for torque vectoring can reduce energy consumption (3%) in normal driving and improve vehicle dynamic behaviour in cornering manoeuvres. Different control allocation strategies are presented in [37] for momentaneous energy minimisation of an EV over a realistic driving cycle. In this work it was found that both offline and online optimisation approaches, the former based on look-up tables and the latter on the analytical solution of a quadratic program include vehicle dynamics, achieved significant savings (4%) when controlling the torque allocation of each wheel individually. All these works demonstrate the potential of considering torque vectoring for energy optimisation purposes. Nevertheless, none of this include predictive approaches neither include the problem of path planning and/or tracking. The authors of [27] proposed a similar architecture for energy-efficient torque vectoring however introducing a predictive step. Firstly, offline, the understeer characteristics are optimised based on the driver steering angle and pedals input, together with the longitudinal torque split. Powertrain and tire slip power losses are minimised.

Once again, it is found that this optimisation is a main driver of energy optimisation in vehicle motion control. Then, NMPC is run online for optimal allocation of individual wheel torques. Extensive testing of this architecture is done in different scenarios: driving cycles, obstacle avoidance with low-friction, and racing track. Overall energy reduction due to longitudinal torque vectoring is around 2% increasing up to 10% if in racing track scenarios involving very dynamic maneuvers. In those, also the steering angle usage was decreased significantly. Based on such testing and further tuning analysis, the authors conclude the power losses are mainly in the powertrain for low-dynamics scenarios whereas for more dynamical scenarios, like race tracks, tire slip losses are the most relevant. Finally, in the very relevant study conducted in [25], the authors aimed to improve energy efficiency in an EV applied to the autonomous driving case. As for vehicle stability control, NMPC was applied for path tracking and power losses minimisation, where it was proved that predictive approaches can achieve higher energy savings than instantaneous ones. It was however found that such optimal strategies couldn't outperform equal-torque-distribution due to tuning trade-offs involved in the MPC formulation. However, this work showed a possible path for further development of this topic.

The presented survey on related literature shows that the joint problem of trajectory planning and torque vectoring has not yet been solved. This is motivated by the potential on optimal torque distribution for tracking of optimally selected understeer characteristics. In the scenario of autonomous driving, the complete vehicle attitude can be optimally selected, and thus full potential on energy saving can be exploited. In previous works at Volvo Cars, different master theses [34][44] have described the potential of predictive control for both stability control and energy consumption minimisation, considering steering, total torque request, and torque vectoring optimization. In the latest works of [25] and [44], torque vectoring proved promising when applied in a predictive fashion. The results of this thesis then complement these works, in the sense that it focuses on the simpler problem of energy minimisation optimizing steering angle and total torque request, while torque vectoring is not considered. This way, the complexity of the problem is reduced and more emphasis can be put on fast computations for potential vehicle implementation. The contribution is then not only to describe potential savings on energy from trajectory planning but also to document important computational aspects if real-vehicle implementation is the goal.

3
Numerical Optimal Control

3.1 Optimization Problem

An optimal control problem (OCP) consists of determining the best set of control inputs over a given time period to optimize a certain performance criterion, subject to system dynamics and constraints. The type of dynamical systems addressed in this thesis can be defined by a set of states $\mathbf{x} \in \mathbb{R}^{n_x}$ and control inputs $\mathbf{u} \in \mathbb{R}^{n_u}$ describing the relevant properties of the system and a differential equation $\dot{\mathbf{x}} = f(\mathbf{x}, \mathbf{u}, \tau)$ characterizing the evolution of the states with respect to the independent variable τ, commonly set as time, $\tau = t$. Such dynamical systems can evolve in time within the manifold described by f and can describe a multitude of trajectories depending on the initial state and the control input trajectory. If one is interested in optimizing the state trajectory over a certain time horizon, then an OCP can be formulated and solved to obtain the optimal control input sequence with respect to the optimization objective defined in the problem. Moreover, the OCP can consider both physical constraints arising from physical modelling of the dynamical system and actuators as well as user-defined constraints arising from specifications of the control design. A discrete OCP can be defined as follows:

$$\min_{\substack{\mathbf{x}_0,\mathbf{u}_0,\mathbf{x}_1,\mathbf{u}_1,\ldots,\\ \mathbf{x}_{N-1},\mathbf{u}_{N-1},\mathbf{x}_N}} = \sum_{k=0}^{N-1} l(\mathbf{x}_k, \mathbf{u}_k, \tau_k) + m(\mathbf{x}_N, \tau_N) \tag{3.1}$$

$$\text{s.t.} \quad \mathbf{x}_0 - \hat{\mathbf{x}}_0 = 0 \tag{3.2}$$

$$\mathbf{x}_{k+1} - f_d(\mathbf{x}_k, \mathbf{u}_k, \tau_k) = 0, \quad \forall k \in [0, N] \tag{3.3}$$

$$\mathbf{x}_k \in \mathcal{X}, \quad \forall k \in [0, N-1] \tag{3.4}$$

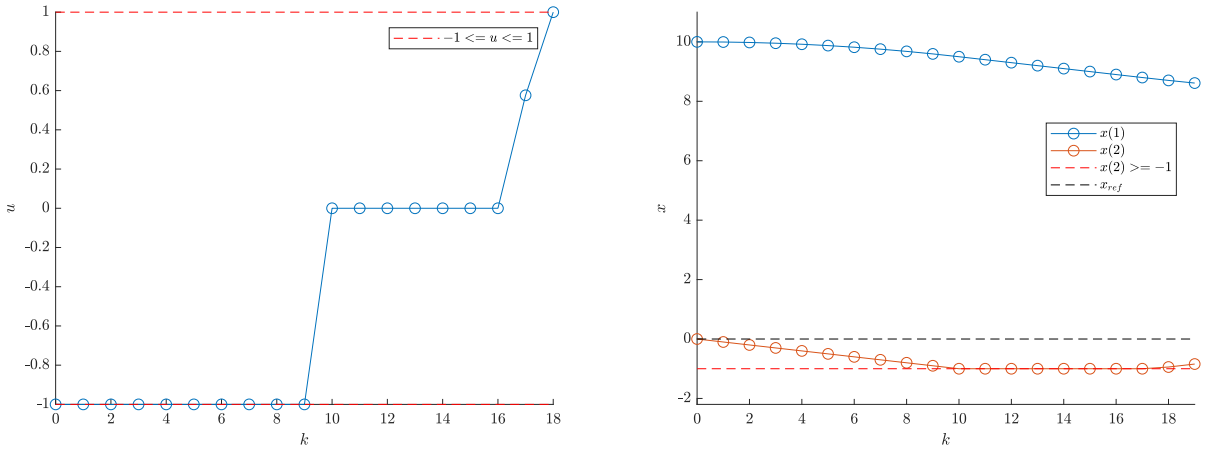
$$\mathbf{u}_k \in \mathcal{U}, \quad \forall k \in [0, N-1] \tag{3.5}$$

$$g(\mathbf{x}_k, \mathbf{u}_k, \tau_k) \leq 0, \quad \forall k \in [0, N-1] \tag{3.6}$$

$$\mathbf{x}_N \in \mathcal{X}_f \tag{3.7}$$

where the subscript k represents the k^{th} discrete node of the trajectory and N is the prediction horizon. In (3.1), l and m are the running and target costs, respectively, the latter being also commonly referred to as the Meyer term. Regarding the constraints, $\hat{\mathbf{x}}_0$ is the initial state, $f_d(\mathbf{x}_k, \mathbf{u}_k, \tau_k)$ represents the discretized dynamics of the systems, and $\mathcal{X}$, $\mathcal{U}$, and $\mathcal{X}_f$ are the feasible state, feasible control input, and target sets, respectively. Additionally, nonlinear constraints $g(\cdot)$ can also be defined to relate different states and control inputs together. The illustration of an OCP solution can be seen in Fig. 3.1.

Although solving the OCP stated above yields an optimal control input trajectory, that in reality is only optimal if there are no model-plant-mismatches arising from imperfect modelling of the real system and unmodelled disturbances at different levels of the architecture. Therefore, to apply optimal control online, a feedback structure is needed. The strategy of solving optimal

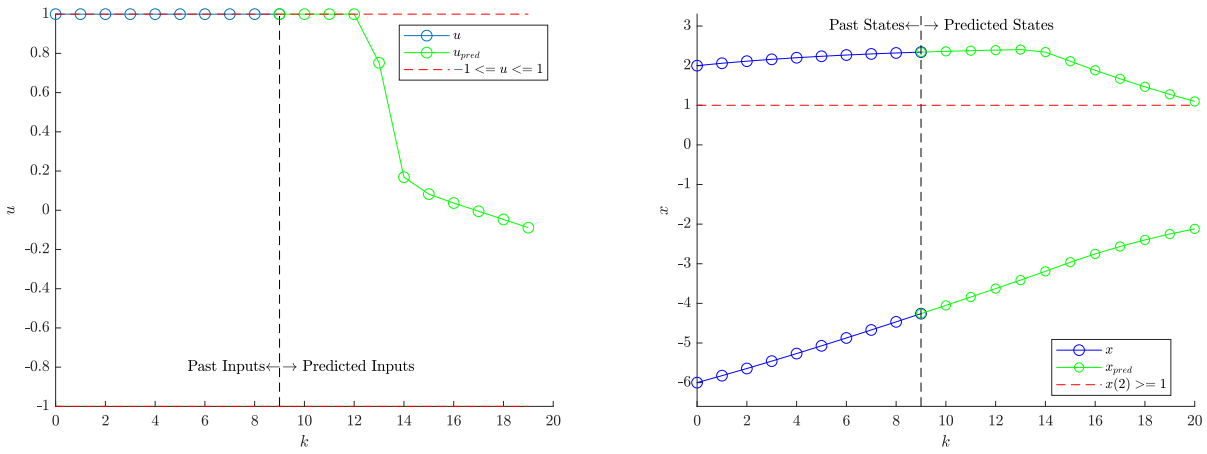


(a) Control input trajectory **(b)** State trajectory

Figure 3.1: OCP solution example

control problems online in a feedback fashion is in essence Model Predictive Control (MPC) [29]. MPC is an optimal feedback control law that solves a finite-horizon OCP at every iteration of the control update. It competes with traditional methods such as Proportional-Integral-Derivative (PID) and Linear Quadratic Regulator (LQR) controllers due to its capacity of computing optimal control inputs while operating within state and input constraints. The MPC algorithm, illustrated in Fig. 3.2, is as follows:

- Measure or estimate current state $\hat{\mathbf{x}}_0$;
- Solve OCP to find optimal control input trajectory $[\mathbf{u}_0^*, \mathbf{u}_1^*, \cdots, \mathbf{u}_{N-1}^*]$ based on predicted state trajectory $[\mathbf{x}_0^*, \mathbf{x}_1^*, \cdots, \mathbf{x}_N^*]$ given the system dynamic model;
- Apply the first optimal control input $\mathbf{u}_0^*$ at the real system;
- Move the optimisation horizon one step forward and repeat loop.



(a) Control input trajectory **(b)** State trajectory

Figure 3.2: MPC illustration

3.2 Solving Model Predictive Control

The OCP described in equations (3.1) - (3.7) is stated in discrete time. Nevertheless, real dynamic systems are naturally in continuous time and they need to be discretized to be solved by digital embedded controllers. Assuming that in reality every optimal control problem is by its nature defined in continuous time, following the outline presented in [14], three main approaches to address and solve such problems can be enumerated. These approaches are called transcription methods and are the following:

- Dynamic programming [5] relies on the principle of optimality and computes a control law for the entire state and time domains. It is however intricate to solve numerically for high dimensional problems and therefore is often not suited to real-time implementations when the state-space dimension is large.
- Indirect methods, such as the Pontryagin Maximum Principle [28], define a boundary value problem from the necessary optimality conditions in the form of ordinary differential equations (ODE). These methods are often referred to as "first optimize then discretize" methods since the optimality conditions are written in continuous time and only after the solution is numerically solved by discretizing the problem. The main disadvantage of such methods is the difficulty associated with solving numerically the ODEs, due to strong nonlinearity and instability.
- Direct methods, instead, are "first discretize then optimize" methods. They consist of transforming the continuous optimal control problem into a nonlinear programming problem (NLP), which can then be efficiently solved by state-of-the art numerical solvers. These methods are the most widely used methods for constrained OCPs, since they present multiple advantages in the treatment of inequality and equality constraints.

The transcription method selected in the present thesis is the last one presented, the direct method, due to its ease of constraint formulation and how these are treated mathematically by the numerical solver. The other two alternatives require cumbersome implementation and are not well suited for real-time applications, the reason for which they are discarded.

3.2.1 Nonlinear programming

In essence, direct methods reformulate an OCP

$$\min_{\mathbf{x}(\tau),\mathbf{u}(\tau)} = J(\mathbf{x}(\tau),\mathbf{u}(\tau)) \tag{3.8}$$

$$\text{s.t.} \quad \dot{\mathbf{x}} = f(\mathbf{x}(\tau),\mathbf{u}(\tau),\tau), \quad \tau \in [0,\tau_f] \tag{3.9}$$

$$\mathbf{x}(\tau) \in \mathcal{X}, \quad \tau \in [0,\tau_f] \tag{3.10}$$

$$\mathbf{u}(\tau) \in \mathcal{U}, \quad \tau \in [0,\tau_f] \tag{3.11}$$

$$g(\mathbf{x}(\tau),\mathbf{u}(\tau),\tau) \leq 0, \quad \tau \in [0,\tau_f] \tag{3.12}$$

$$\mathbf{x}(\tau_f) \in \mathcal{X}_f \tag{3.13}$$

into a NLP of the form

$$\min_{\mathbf{w}} = J(\mathbf{w}) \tag{3.14}$$

$$\text{s.t. } g(\mathbf{w}) \leq 0 \tag{3.15}$$

$$h(\mathbf{w}) = 0 \tag{3.16}$$

where the decision variables now assume the form of a vector of real numbers $\mathbf{w}$ and the entire problem is formulated as a function of $\mathbf{w}$. Among the discrete method family of transcription methods, all apply some finite dimensional parameterization of the control input trajectory while differing in the way the state trajectory is handled. Two different approaches can be identified: sequential and simultaneous.

A sequential approach consists of parameterizing only the control input trajectory, commonly assuming piece-wise constant control input value within each sampling interval. In this case, the state trajectory is set as dependent of the control input trajectory and the initial state and is simulated based on the ODE describing the dynamics of the system, as in the direct single shooting method [31]. It is thus called sequential since, for each "shooting node" of the optimization problem, it first optimizes the control input and then simulates the state evolution. Such approach thus results in a dense problem with low dimension, with the trade-off that it can easily make the problem highly nonlinear. Given a nonlinear ODE describing the dynamics of the system, for large enough sampling intervals or long horizons, even slight nonlinearities in the dynamics can induce high nonlinearities in the constraint and cost functions throughout the integration of the equation. Such problems can be critical e.g. in the implementation of MPC where problems can have long horizons and nonlinear functions, which disregards sequential approaches as a candidate for the scope of this thesis.

Conversely, in a simultaneous approach, both the control input and the state trajectories are parameterized, where optimization and simulation are performed simultaneously. Essentially, compared with its counterpart, this approach trades nonlinearity by dimensionality where the number of optimization variables grow, in order to include all the state trajectory, as well as the number of equality constraints, added to ensure continuity of the state trajectory. This approach results in high dimension problems but with a sparse structure that can be efficiently exploited [14]. Furthermore, it usually results in faster local convergence rates and less numerically instability, mainly because an initial guess of the state trajectory can be provided and integration of the system is now performed piecewise between points that are in an arbitrarily small vicinity of each other. Direct multiple shooting [7] and direct collocation [6] are two direct methods following a simultaneous approach, and are used in this thesis. A more thorough explanation of each of those follows.

3.2.1.1 Direct multiple shooting

Commonly to all direct methods, the direct multiple shooting method starts by parameterizing the control input trajectory:

$$\mathbf{u}(\tau) = \mathbf{u}_k,\ \tau \in [\tau_k, \tau_{k+1}]\ .$$

Then, it simulates the ODE $\dot{\mathbf{x}}(\tau) = f(\mathbf{x}(\tau), \mathbf{u}(\tau), \tau)$ for each sampling interval $[\tau_k, \tau_{k+1}]$ independently. Such independent simulations at each sampling interval result in mismatch between the target and starting points of subsequent sampling intervals. For that reason, the optimization problem should include the set of equality constraints,

$$\begin{aligned} \mathbf{x}_{k+1} &= f_d(\mathbf{x}_k, \mathbf{u}_k, \tau_k) \\ \mathbf{x}_0 &= \hat{\mathbf{x}}_0\ , \end{aligned}$$

where f_d is an approximation of f resulting from numerical integration. Note that by including such constraints the problem formulation remains in anyway identical to the single shooting counterpart but with a sparser structure.

Typically a Runge-Kutta (RK) integration scheme is used to obtain f_d. The RK method can be of different order of approximation, according to the number of slope approximations within one sampling interval. RK4 employs a fourth order approximation of such slope and results in the following approximation of f for each sampling interval:

$$p_1 = f(\mathbf{x}_k, \mathbf{u}_k, \tau_k) \tag{3.17}$$
$$p_2 = f(\mathbf{x}_k + p_1/2, \mathbf{u}_k, \tau_k + h/2) \tag{3.18}$$
$$p_3 = f(\mathbf{x}_k + p_2/2, \mathbf{u}_k, \tau_k + h/2) \tag{3.19}$$
$$p_4 = f(\mathbf{x}_k + p_3, \mathbf{u}_k, \tau_k + h) \tag{3.20}$$
$$\mathbf{x}_{k+1} = f_d(\mathbf{x}_k, \mathbf{u}_k, \tau_k) = \mathbf{x}_k + h\frac{1}{6}\left(p_1 + 2p_2 + 2p_3 + p_4\right) \ , \tag{3.21}$$

where p_i are the intermediate points where the slope is evaluated at and h is the sampling interval length.

3.2.1.2 Direct collocation

The direct collocation method is analogous to the direct multiple shooting one, with the difference that additional intermediate points, the "collocation" points, enter the NLP as decision variables, thus increasing the resolution of the discretization grid and consequently the problem dimension.

The control input is treated equally as in direct multiple shooting, being parameterized per sampling interval regardless of the finer state trajectory grid:

$$\mathbf{u}(\tau) = \mathbf{u}_k, \ \tau \in [\tau_k, \tau_{k+1}] \ .$$

Conversely, the state trajectory is discretized both at the extremes of the sampling interval and also at the collocation points. The placement of such points depends on the parameterization chosen. One common choice is the Gauss-Legendre (third order) collocation scheme, which consider additional points at:

$$\tau_{k,i} = \tau_k + h\mu_i, \quad \forall k \in [0, N]$$
$$\mu = \begin{bmatrix} 0 & 0.112702 & 0.500000 & 0.887298 \end{bmatrix} \ ,$$

where N is the prediction horizon of each NLP. For what follows, let $\mathbf{x}_{k,i}$ denote the states at $\tau_{k,i}$. Once again, similarly to the direct multiple shooting method, equality constraints are necessary to ensure the continuity of the state trajectory and need to be derived for all collocation points [1]. At each sampling interval, a Lagrangian basis polynomial $l(\mu)$ is defined:

$$l_i(\mu) = \prod_{j=0, j\neq i}^{3} \frac{\mu - \mu_j}{\mu_i - \mu_j}, \tag{3.22}$$

which satisfies

$$l_i(\mu_r) = \begin{cases} 1 & i = r \\ 0 & \text{otherwise} \end{cases} \ . \tag{3.23}$$

That basis can then serve to approximate the state through a linear combination,

$$\tilde{\mathbf{x}}_k(\tau) = \sum_{j=0}^{3} l_j\left(\frac{\tau - \tau_k}{h}\right)\mathbf{x}_{k,j}, \tag{3.24}$$

and to approximate the state derivatives at $\tau_{k,i}$, $i = \{1, 2, 3\}$,

$$\dot{\tilde{\mathbf{x}}}_k(\tau_{k,i}) = \frac{1}{h}\sum_{j=0}^{3} \dot{l}_j(\mu_i)\mathbf{x}_{k,j} \ . \tag{3.25}$$

The state at the ending extreme of the sampling interval is:

$$\tilde{\mathbf{x}}_{k+1,0} = \sum_{j=0}^{3} l_j(1)\mathbf{x}_{k,j} \ . \tag{3.26}$$

At this point it is possible to formulate the aforementioned equality constraints:

$$hf(\mathbf{x}_{k,i}, \mathbf{u}_k, \tau_{k,i}) - \sum_{j=0}^{3} \dot{l}_j(\mu_i)\mathbf{x}_{k,j} = 0, \quad \forall k \in [0, N], i \in 1, 2, 3. \tag{3.27}$$

$$\tilde{\mathbf{x}}_{k+1,0} - \sum_{j=0}^{3} l_j(1)\mathbf{x}_{k,j} = 0, \quad \forall k \in [0, N] \ , \tag{3.28}$$

(3.27) ensuring continuity of the state trajectory among collocation points and (3.28) doing the same among sampling intervals.

The main advantage of this method compared to the multiple shooting counterpart is the increase in sparsity of the problem and in robustness to instability and nonlinearity resulting from considering more and closer points per sampling interval.

3.2.2 Numerical solvers

Once the OCP is transformed into a NLP, the local optimal solution needs to be found. Under some assumptions on constraint qualifications, there is a local solution $\mathbf{w}^*$ of (3.14) - (3.16) that satisfies the Karush-Kuhn-Tucker (KKT) conditions that states that there exist multiplier vectors λ^* and μ^* such that the following hold:

$$\nabla_{\mathbf{w}}\mathcal{L}(\mathbf{w}^*, \lambda^*, \mu^*) = 0 \tag{3.29}$$

$$g(\mathbf{w}^*) \leq 0 \tag{3.30}$$

$$h(\mathbf{w}^*) = 0 \tag{3.31}$$

$$\mu^* \geq 0 \tag{3.32}$$

$$g(\mathbf{w}^*)\mu^* = 0, \tag{3.33}$$

where the Lagrangian is defined as:

$$\mathcal{L} = J(\mathbf{w}) + g(\mathbf{w})^T\lambda + h(\mathbf{w})^T\mu \ . \tag{3.34}$$

Nearly all algorithms to solve this system of equations derive from the Newton's method. Newton type of optimization methods attempts to solve this system of equations in a line-search approach based on successive linearizations of the problem. Sequential Quadratic Program (SQP) and Interior Point Method (IPM) are two main methods of this family and are the most established approaches for solving problems as the ones in this thesis [29].

3.2.2.1 Sequential Quadratic Programming

SQP methods linearize the entire problem and solve an quadratic program (QP) iteratively until convergence. The underlying QP at each iteration m is of the form

$$\min_{\mathbf{w}} \quad \nabla J(\mathbf{w}^m)^T \mathbf{w} + \frac{1}{2} (\mathbf{w} - \mathbf{w}^m)^T B_m (\mathbf{w} - \mathbf{w}^m) \tag{3.35}$$

$$\text{s.t.} \quad g(\mathbf{w^m}) + \nabla g(\mathbf{w}^m)^T g(\mathbf{w}) (\mathbf{w} - \mathbf{w}^m) \leq 0 \tag{3.36}$$

$$h(\mathbf{w^m}) + \nabla h(\mathbf{w}^m)^T h(\mathbf{w}) (\mathbf{w} - \mathbf{w}^m) = 0 \ , \tag{3.37}$$

where $\mathbf{w}^m$ is the linearization point at iteration m and B_m is either an approximation of the Hessian or its exact value $\nabla^2_{\mathbf{ww}} \mathcal{L}(\mathbf{w}^*, \lambda^*, \mu^*)$, depending on the nonlinearities of the functions. Commonly, one often chooses positive semi-definite approximations of the Hessian to ensure convexity of the QP. By doing so the problem becomes itself convex, being easier to solve.

For real-time applications, the solution of the SQP problem needs oftentimes to be outputted within strict time requirements. Moreover, the surrounding environment to the system is normally changing dynamically and unpredictably. As a result of such circumstances, normally one is interested in trading optimality by computational speed. One possibility is to relax the requirement on convergence of the SQP method, meaning that less QP need to be solved. Such alternative is denoted as real time iteration (RTI) scheme, which performs one or a reduced number of iterations of the SQP. Furthermore, even higher reduction in computational cost can be attained per sampling interval by performing simpler computations than in Newton's method, basically relying on the reuse of matrices and system factorizations [13]. It is nevertheless crucial to highlight that RTI depends heavily on the initial guess, meaning that it should be always warm started, possibly with the shifted previous solution. In practice, strategies such as running SQP for the initial part of the trajectory until convergence and then switching to RTI can be useful to balance accuracy and computational speed.

3.2.2.2 Interior Point Method

IPM, instead, treats the inequality constraints and the complementary slackness differently, modifying it to become:

$$g(\mathbf{w}^*)\mu^* = \nu, \ \nu > 0 \tag{3.38}$$

where ν is called the barrier term. Then, the constraints g are relaxed and taken into account in the cost function, which becomes:

$$J(\mathbf{w}) - \nu \log(-g(\mathbf{w})) \ . \tag{3.39}$$

The method works as follows. It starts by setting ν high enough and solves the first problem with the Newton's method. The change in complementary slackness yields a solution inside the feasible set, instead of at its boundary as it would be the case for the original KKT system (3.29) - (3.33). This point is called the Interior Point. Then, ν is reduced and the algorithm is once again run, a process that is repeated until convergence to the solution of the original KKT system, apart from a numerical tolerance.

State-of-the-art solvers that implement IPM, such as IPOPT [41] and HPIPM [21], do not actually implement the simpler IPM solver, but rather the primal-dual interior point method (PDIPM) counterpart. Differently from IPM, PDIPM do not require explicitly the logarithmic barrier function as in (3.39) and solutions are not necessarily feasible. Moreover, as the name

suggests, PDIPM finds search directions for both the primal and dual variables, and update them iteratively driving the barrier term to zero. Performance-wise, it often results in improved efficiency and convergence properties compared with IPM, with enhanced numerical stability. For a detailed analysis on such methods please refer to [43].

3.2.2.3 SQP vs IPM

Firstly, it is important to note that SQP needs to solve multiple QP iteratively whereas IPM solve instead multiple linear system of equations which is cheaper. In the case the QP is solved with an IPM, the computational cost of SQP becomes even higher. Nevertheless, the computational expenditure does not depend solely on number of problems or iterations.

As described in [13], in all Newton type of optimization algorithms two main steps are responsible for increase of computational cost: derivative computation and solution of the quadratic sub-problems.

Regarding the derivative computation, the computation of the Hessian in both methods can be very expensive in memory and in linear algebra computations for large problems, i.e. with a high number of optimization variables. To address this problem, linear algebra operations can be employed to exploit sparsity in matrices. Alternatively, reduction of the state space can be employed to reduce the number of derivative computations.

On another point, there are multiple considerations to be done regarding the solution of each QP. Different QP solvers, such as qpOASES [19] or HPIPM [21], can exploit the structure of the problem and its associated sparsity, mainly in the case of simultaneous approaches, by using structure-exploiting linear algebra libraries. Furthermore, the QP can be condensed [13] so that the problem size is reduced and the problem becomes denser. Condensing a problem essentially means to reduce its size by eliminating decision variables, namely the state trajectory in the simultaneous approach, making it a function of the initial state and the control trajectory. This is done while still guaranteeing the same state integration properties of the original approach, therefore not inducing the numerical problems present in sequential approaches.

In general, taking multiple factors in consideration, if the NLP involves cheap evaluation of functions and their derivatives, as in direct collocation, and if the initial guess is far from the region of convergence, IPM would be preferable. Conversely, assuming the problem has a small to medium size, SQP would be better in case of expensive function evaluations, as in direct multiple shooting, and when good initial guesses can be provided.

3.2.3 Software

Two different software are used to solve the MPC problems in this thesis. The open source tool for nonlinear programming and algorithm differentiation CasADi [2] is used to implement an IPM solver, IPOPT [41], which is designed for large-scale nonlinear optimization. Given the characteristics of IPM, casADi implements the direct collocation transcription method. Then, acados [39], also an open source tool that implements fast solvers for NLP arising from OCP, is used to implement an SQP solver with direct multiple shooting. The underlying solver for each QP is the HPIPM [21], a high-performance interior-point method solver for optimal control convex and quadratically-constrained QP. In the implementation of this thesis, the solver considers exact Hessian approximation and partial condensing routines, outputting solutions within a remarkably low computational time. Additionally, acados provides support

for auto generation of C-code, which makes it specially useful for implementation on embedded systems.

3.2.4 Practical aspects of MPC implementation

When implementing MPC, it becomes critical to guarantee both feasibility and low computational time at each iteration. To enhance these features, in both implementation the solvers are warm started with the previous solution. This showed, as expected, great improvement in computational efficiency.

Furthermore, since it is expected for the system to operate near the constraints, in order to keep the computational complexity simple, those are relaxed and are soft constrained. Mathematically, relaxation of constraints consists of adding a slack variable as an optimization variable, $\epsilon \geq 0 \in \mathbb{R}^N$, and set the relaxed constraints as:

$$g(\mathbf{w}) - \epsilon \leq 0$$

as well as adding a penalization of violation of the constraint in the cost function. This strategy has been found to be useful to decrease the standard variation of computational times, mainly for IPM [33], meaning that the solver faces less problems when getting closer to the constraints.

On another point, numerical stability is known to be enhanced if the domain of the operation variables lies within the range $[0, 1]$. This helps avoiding matrices to be ill-conditioned, as e.g. search directions are normalized in magnitude. To do so, all the optimization variables were scaled as

$$z^s = \frac{z}{z_{\max}} \ ,$$

where also constraints were defined in the scaled domain. Then, when computing the cost function, the variables were scaled inversely to penalize quantities in their original magnitude.

Lastly, when computing the cost function, high values can be attained, which can be heavy in memory and linear algebra computations. For that reason, at each instant, the cost functions can be scaled by a high enough value to reduce their size in magnitude, as e.g.

$$J^s = \frac{J}{J_{\max}},$$

where the scaling factors do not need to be accurately chosen since they impact the entire value of the cost function thus .not changing therefore the relative weight of each term.

4
Vehicle modelling

This chapter outlines the chassis and tire models that are used to describe the vehicle dynamics. Secondly, energy dissipation in the powertrain is presented alongside models of the electric motor and tire power losses. Finally, the MPC problem is reformulated in the spatial domain and its formulation presented.

4.1 Global coordinate system

The vehicle pose (x^I,y^I,ψ) with respect to the inertial frame can be obtained as:

$$\begin{bmatrix} \dot{x}^I \\ \dot{y}^I \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} \cos(\psi) & -\sin(\psi) & 0 \\ \sin(\psi) & \cos(\psi) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} v_x \\ v_y \\ r \end{bmatrix} \tag{4.1}$$

where v_x, v_y, and r are the longitudinal and lateral velocities and the yaw rate of the vehicle.

4.2 Curvilinear coordinate system

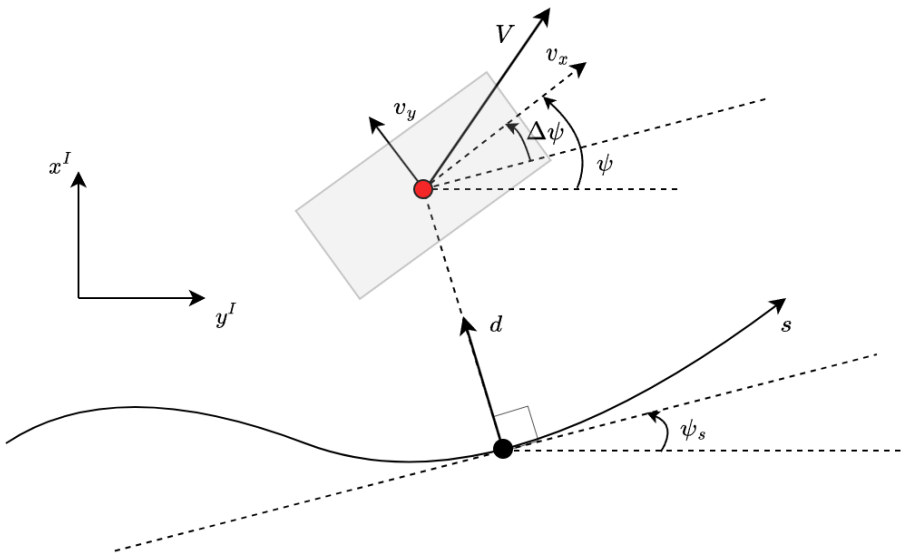


Figure 4.1: Reference path and curvilinear coordinate system variables

If a certain path is given, as in Fig. 4.1, the vehicle position (highlighted by the red dot in the figure) can instead be defined relative to it, for ease of problem formulation. Let s, d, and

$\Delta\psi = \psi - \psi_s$ be the distance travelled along the path, the lateral displacement from s, and the local heading angle, respectively. Once defined the states relative to path, analogously to [22], [20], and [9], the kinematics of these variables are described by the equations

$$\dot{s} = \frac{v_x \cos(\Delta\psi) - v_y \sin(\Delta\psi)}{1 - \kappa(s)d} \tag{4.2}$$

$$\dot{d} = v_x \sin(\Delta\psi) + v_y \cos(\Delta\psi) \tag{4.3}$$

$$\Delta\dot{\psi} = r - \kappa(s)\dot{s} \ , \tag{4.4}$$

where $\kappa(s)$ is the curvature of the path at s.

4.3 Chassis model

In this work, the chassis model that is selected to describe the vehicle dynamics is the single-track model [4], also known as bicycle model. The main reason for such decision is that, since this project aims for a real-time implementation in a test vehicle, model simplicity plays a role in the feasibility of such implementation considering the available computational capabilities.

The single-track model, illustrated in Fig. 4.2, is a two-wheel representation of the chassis.

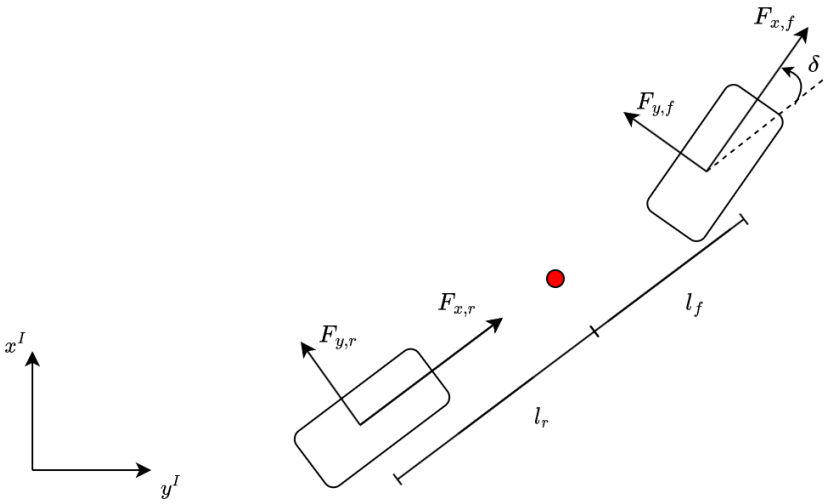


Figure 4.2: Single-track model

Consider a planar road with zero slope and bank angles. From Newton's second law applied to forces and moments, and taking into account the rotating coordinate frame attached to the center of mass of the vehicle, the resulting equations of motion are:

$$\dot{v}_x = \frac{F_x}{m} + v_y r \tag{4.5}$$

$$\dot{v}_y = \frac{F_y}{m} - v_x r \tag{4.6}$$

$$\dot{r} = \frac{M_z}{I_{zz}} \ , \tag{4.7}$$

where I_{zz} is the rotational inertia around the z-axis. The total forces acting on the center of mass are:

$$F_x = \cos(\delta)F_{x,f} - \sin(\delta)F_{y,f} + F_{x,r} - F_{\text{rr}} - F_{\text{aero}} \tag{4.8}$$
$$F_y = \cos(\delta)F_{y,f} + \sin(\delta)F_{x,f} + F_{y,r} \tag{4.9}$$
$$M_z = l_f \cos(\delta)F_{y,f} + l_f \sin(\delta)F_{x,f} - l_r F_{y,r} \ , \tag{4.10}$$

where $F_{x,i}$ and $F_{y,i}$ are the longitudinal and tire forces, δ is the steering angle, l_i is the distance from the center of mass to the front (f) and rear (r) axle, all respective to wheel $i \in \{f, r\}$. Finally, F_{rr} and F_{aero} are the rolling resistance and the aerodynamic drag, respectively.

Given the nature of the bicycle model, only the longitudinal load transfer can be modelled. In order to accurately model it, a dynamic equation should be considered depending on the acceleration and the suspension parameters. To do so, load transfer should be added as a state, thus increasing the model complexity and dimension. Therefore, to keep model simplicity, static load transfer is considered and modelled as:

$$F_{z,f} = \frac{mgl_r}{l_f + l_r} \tag{4.11}$$
$$F_{z,r} = \frac{mgl_f}{l_f + l_r} \ , \tag{4.12}$$

where $F_{z,i}$ is the normal tire force of wheel $i = f, r$.

4.3.1 Rolling Resistance

The rolling resistance as a resistive force emerges as a consequence of the tire carcass deflection and hysteresis. During the rotation of the tire, the rubber deforms when contacting the ground. The energy involved in this deformation is not completely released when leaving the ground due to internal damping in the tire. This leads to a front-biased pressure distribution that generates a moment that opposes the wheel rotation [26] and can be described as a force by:

$$F_{rr_i} = -F_{zi}\{q_{sy1} + q_{sy2}\frac{F_{xi}}{F_{z0}} + q_{sy3}||\frac{v_{xi}}{v_{ref}}|| + q_{sy4}(\frac{v_{xi}}{v_{ref}})^4\} \ , \tag{4.13}$$

where F_{zi} is the normal load of the tire, r_0 is the unloaded tire radius, q_{sy1}, q_{sy2}, q_{sy3} and q_{sy4} are fitting parameters, F_{z0} and v_{ref} are the reference normal load and longitudinal speed of the tire for which measurements were conducted. This model is however highly nonlinear, which is not according to the design requirements of this work. Taking into account the nonlinearity of the last two terms, the fact that the third term is not differentiable, and the reduced relative importance of terms q_{sy3} and q_{sy4}, the model is thus simplified to:

$$F_{rr_i} = \left\{-F_{zi}\{q_{sy1} + q_{sy2}\frac{F_{xi}}{F_{z0}}\right\} \ . \tag{4.14}$$

4.3.2 Aerodynamic Drag

The motion of vehicles is developed within air as fluid. As a result, a resistance that opposes the motion of the vehicle between the air and the vehicle is generated. Such resistance can be decomposed in pressure drag and friction drag.

Friction drag arises from the shear stress between the air and the surface of an object as it moves through the air. This force is caused by the resistance of the air molecules directly in contact with the surface of the vehicle and is proportional to the surface area of the object.

Pressure drag, on the other hand, is caused by the difference in pressure between the front and rear of the vehicle as it moves through the air. This pressure difference creates a force perpendicular to the direction of the flow, which is known as the pressure drag.

Generally, given the aspect ratio of vehicles, they are considered as bluff bodies which leads to pressure drag being the main factor that drives the overall aerodynamic resistance [3]. Friction drag, on the other hand, is neglected given its low magnitude. Pressure drag force is modeled as:

$$F_{\text{aero}} = \frac{1}{2}\rho c_D A_f v_x^2 \tag{4.15}$$

where ρ, c_D, and A_f are the air density, drag coefficient, and vehicle frontal area, respectively, and are constant parameters. Note, however, that such model is an approximation since it disregards wind direction and speed relative to the vehicle.

4.4 Tire model

Throughout time, different deterministic and empirical tire models have been developed to describe the physics of the tires and its interaction with the road. One example of a deterministic tire model is the brush model where the forces provided by the rubber considering a stiff carcass is defined by sliding and adhesion regions in the contact patch. In this work, nevertheless, due to its accuracy and practicality, the empirical Pacejka tire model is considered [36]. Here, such model, defined in [26], is simplified to the following expression:

$$F_{x,i} = f_x(\sigma_{x,i}, \mu, F_{z,i}) \tag{4.16}$$
$$F_{y,i} = f_y(\sigma_{y,i}, \mu, F_{z,i}) \ . \tag{4.17}$$

However, in order to maintain the complexity low by avoiding model non-linearities, the tire model that is used is the linear tire, defined as:

$$F_{x,i} = C_{\sigma_{x,i}}\sigma_{x,i} \tag{4.18}$$
$$F_{y,i} = C_{\sigma_{y,i}}\sigma_{y,i} \tag{4.19}$$

The linear model's longitudinal and cornering stiffness coefficients C can be calibrated using the Pacejka's model by obtaining the slope of the latter at slip 0, as illustrated in Fig. 4.3 by the dotted line.

4.4.1 Tire slip and wheel dynamics

Tire slip, which is defined as the difference between the translational velocity of the wheel hub and the wheel tire at the contact point with the road, is generated during the application of braking or propulsive torque, as well as wheel steering [26]. The velocity of each wheel in its local coordinate system, $v_{x,i}, v_{y,i}$, is described by:

$$\begin{bmatrix} v_{x,i} \\ v_{y,i} \end{bmatrix} = \begin{bmatrix} \cos(\delta_i) & \sin(\delta_i) \\ -\sin(\delta_i) & \cos(\delta_i) \end{bmatrix} \left(\begin{bmatrix} v_x \\ v_y \end{bmatrix} + r \begin{bmatrix} -l_{y,i} \\ l_{x,i} \end{bmatrix} \right) \ , \tag{4.20}$$

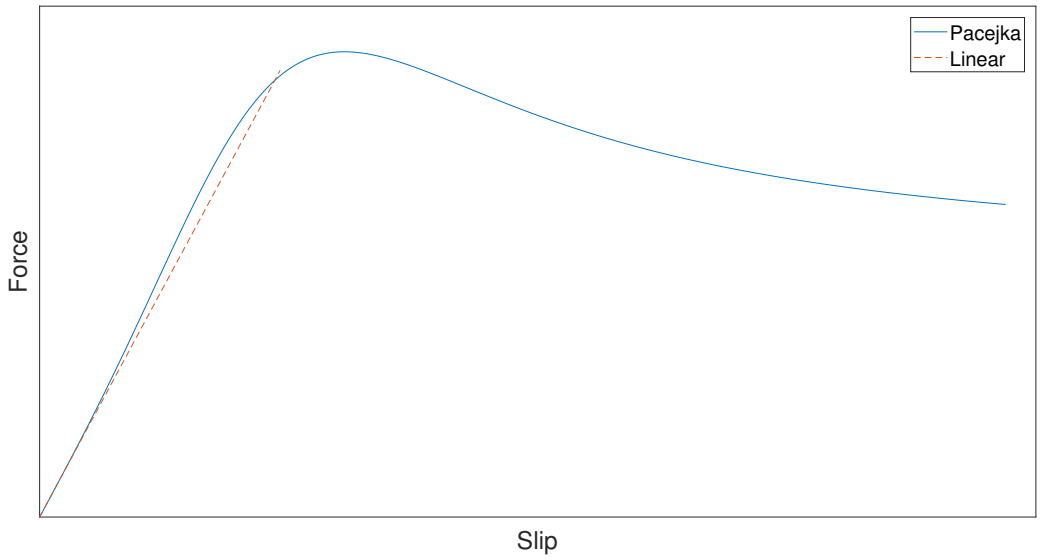


Figure 4.3: Linear and Pacejka's Tire Force vs Slip models and Cornering Stiffness estimation.

where $l_{y,i}$ and $l_{x,i}$ are the longitudinal and lateral position of wheel $i \in \{f, r\}$ with respect to the center of mass. The longitudinal and lateral slip ratios are then defined as [4]:

$$\sigma_{x,i} = \frac{r_0 w_i - v_{x,i}}{v_{x,i}} \ , \tag{4.21}$$

$$\sigma_{y,i} = \arctan(\frac{v_{y,i}}{v_{x,i}}) \ . \tag{4.22}$$

On the other hand, the wheel dynamics are modelled as:

$$I_w \dot{w}_i = T_i - r_0 F_{x,i} \ ,$$

where I_w is the wheel inertia, r_0 is the undeformed wheel radius, whereas T_i and $F_{x,i}$ are the torque allocated and tire force respective to wheel $i \in \mathcal{I}$.

However, analogously to the load transfer in the chassis model, the wheel dynamics have not been included in the tire model. Considering wheel dynamics would entail increasing system complexity and nonlinearity, as well as increasing state dimension where two more states would be needed. As a result, longitudinal slip is considered as zero and the wheel dynamics definition gets simplified to the following expression:

$$T_i = r_0 F_{x,i} \ . \tag{4.23}$$

4.5 Energy consumption

The energy consumed by an electric vehicle is the result of the energy required to overcome road and environmental elements such as road slope, tire slip, and aerodynamics. Additionally, the path that the energy stored on batteries needs to pass to be delivered at the wheels by

the electric motors faces a series of inefficiencies contained in all the elements that are between the batteries and the road, such as the batteries itself, the inverter, the electric motors, the gearboxes and the tires. Electrical components, such as the batteries, the inverter and electric motors, manifest this inefficiency in the form of heat. On the other hand, mechanical components such as gearboxes and tires manifest this inefficiency in the form of heat, deformation and slippage between the tire and the road. In this regard, Fig. 4.4 displays a simplified overview of the energy flow from the batteries to the road.

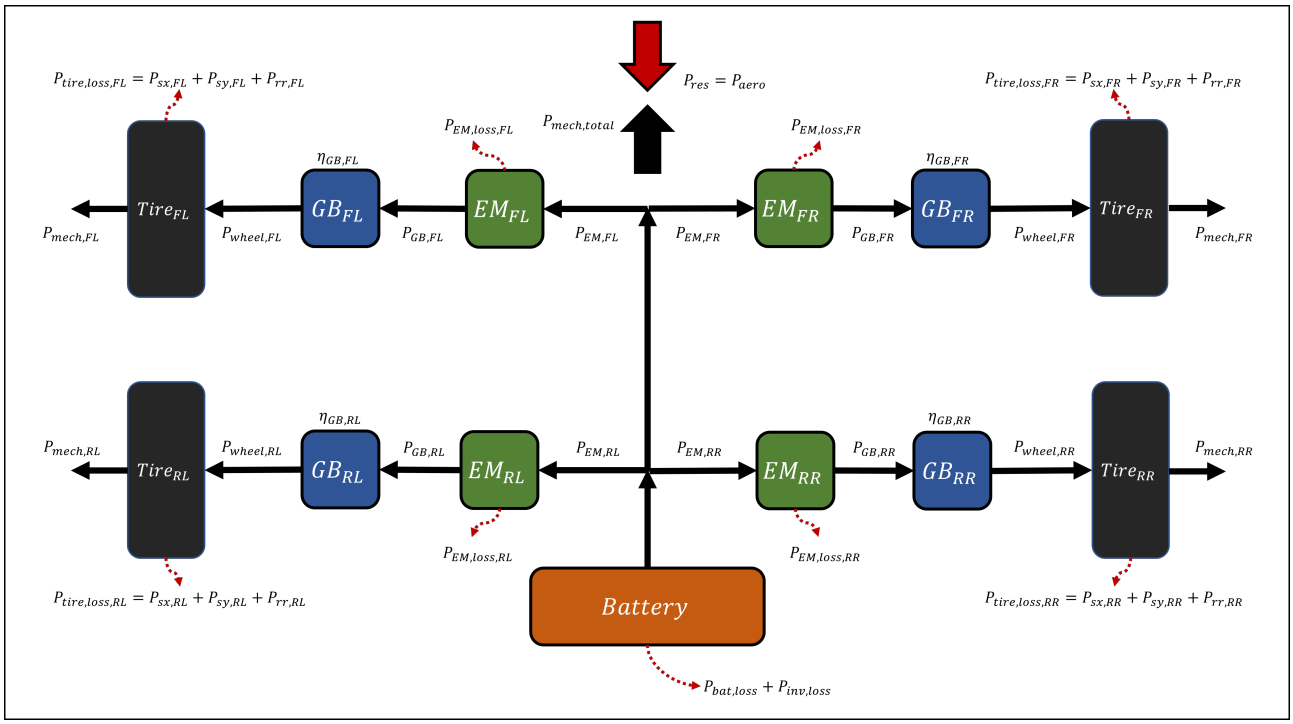


Figure 4.4: Energy flow diagram.

Given the dynamics modelled and presented previously, apart from the electric machine losses, the remaining losses result from the tire and aerodynamic resistive forces. It is therefore possible to state the following power balance equations that apply for this system:

$$P = P_{\text{mech}} + P_{\text{el.loss}} \tag{4.24}$$

$$P_{\text{mech}} = P_{\text{inertia}} + P_{\text{tire loss}} + P_{\text{aero}} \ . \tag{4.25}$$

Here, the total power that the battery needs to deliver to the motor is a result of both the mechanical power P_{mech} needed for vehicle propulsion and the associated losses $P_{\text{el.loss}}$ from the electric machines, i.e. motors and inverter. In turn, P_{mech} is the result of the inertial power P_{inertia}, induced by the longitudinal and lateral accelerations of the vehicle, the tire power losses $P_{\text{tire loss}}$ and aerodynamic power losses P_{aero}. Following, a description of each enumerated power is presented.

4.5.1 Electric power losses

Commonly, electric motors have a high efficiency throughout the entire operating range in comparison to other technologies, such as internal combustion engines. However, in the

scope of this work, a detailed profile of the efficiency across its operating range is required to capture accurately the the most efficient operating points at a given angular speed. Given efficiency operating points measured experimentally, and assuming symmetry around motor torque, provided by Volvo Cars, a curve was fitted to model continuously the electric losses of the electric machines, motor and inverter. The basis function chosen for the fitting is a 5th order polynomial in motor rotational speed ω and 2nd order in motor torque, T:

$$P_{\text{el.loss}}(\omega, T) = p_{00} + p_{10}\omega + p_{01}T + p_{20}\omega^2 + p_{11}\omega T + p_{02}T^2 + p_{30}\omega^3 + p_{21}\omega^2 T \tag{4.26}$$

$$+ p_{12}\omega T^2 + p_{40}\omega^4 + p_{31}\omega^3 T + p_{22}\omega^2 T^2 + p_{50}\omega^5 + p_{41}\omega^4 T + p_{32}\omega^3 T^2 \tag{4.27}$$

whose fitting is illustrated in Fig. 4.5.

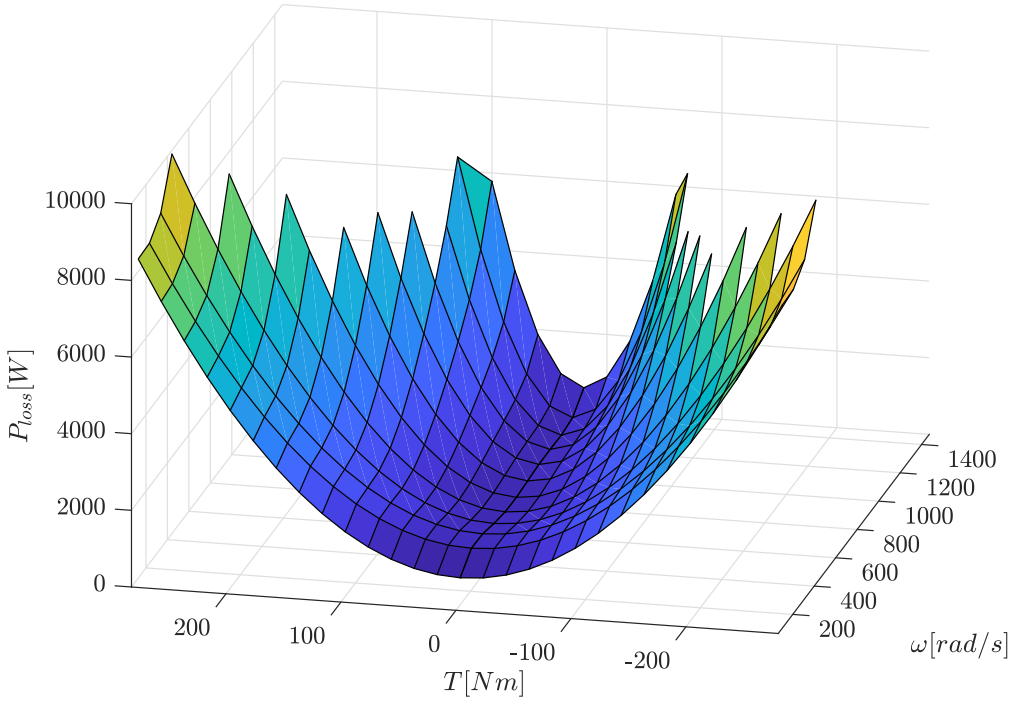


Figure 4.5: Electric machine losses map.

4.5.2 Mechanical power

The mechanical power outputted from the motor can be computed as a function of its rotation speed w_M and torque T_M

$$P_{\text{mech}} = P_{\text{inertia}} + P_{\text{tire loss}} + P_{\text{aero}} = w_M T_M \ , \tag{4.28}$$

where w_M can be computed from the wheel rotational velocity for each axle. This formulation will be useful for the formulation of the cost function later on.

4.5.2.1 Inertial power

The inertial power considered in the controller model is only due to the vehicle mass, the inertia of rotating parts, such as wheels, not being considered. It can therefore be computed as

$$P_{\text{inertia}} = m\left(a_x v_x + a_y v_y\right) \ . \tag{4.29}$$

Nevertheless, in the analysis in Section 6, the inertial power is computed considering as wheel the wheel inertia.

4.5.2.2 Tire power losses

The energy losses coming from the tires that are considered in this work are attributed to two physical phenomenons: tire slip and rolling resistance. Note that all the following losses are necessarily positive, describing the power dissipation in each of the source.

The losses due to tire slip are decomposed in longitudinal and lateral slip losses and are defined as:

$$P_{sx} = \sum_{i=1}^{4} F_{xi}(r_e w_i - v_{xi}) \tag{4.30}$$

$$P_{sy} = -\sum_{i=1}^{4} F_{yi} v_{yi} \ , \tag{4.31}$$

whereas the losses due to rolling resistance are defined as:

$$P_{rr} = \sum_{i=1}^{4} -F_{rr_i} r_0 w_i \ . \tag{4.32}$$

As a result, the total energy loss coming from the tires is defined as:

$$P_{\text{tire loss}} = P_{sx} + P_{sy} + P_{rr} \ , \tag{4.33}$$

In this regard, considering that the longitudinal slip is zero since wheel dynamics are not modelled, the longitudinal slip losses are not minimized and are considered as zero.

4.5.2.3 Aerodynamic power losses

Inline with the aerodynamic drag definition given previously, the losses given by the aerodynamic resistance are given by:

$$P_{\text{aero}} = F_{\text{aero}} v_x \ . \tag{4.34}$$

4.6 Spatial formulation

The curvilinear coordinate system is used to formulate the controller state space. The main advantage of this reference system is that the path constraints, which can be of cumbersome definition in the global coordinate system, become state constraints in the form of bounding boxes, easing problem formulation.

The state space is then a combination of the kinematic states of the curvilinear coordinate system - s, d, $\Delta\psi$ - and the vehicle dynamics states - v_x, v_y, r. Additionally, the input vector is lifted in order to be possible to constrain and penalize the actuators. The state and control vector are thus defined as:

$$\mathbf{x} = \begin{bmatrix} s & d & \Delta\psi & v_x & v_y & r & \delta & T \end{bmatrix}^T \tag{4.35}$$

$$\mathbf{u} = \begin{bmatrix} \dot{\delta} & \dot{T} \end{bmatrix}^T \ . \tag{4.36}$$

The actuators are front-axle steering angle δ, common to both wheels, and total propulsion torque T.

In a following step, a spatial transformation is applied relating the evolution of the state to another independent variable, the distance travelled along the path s. Note that this variable is a state in the curvilinear coordinate system, which allows one to eliminate it when sampling in space, yielding the reduced state vector:

$$\mathbf{x} = \begin{bmatrix} d & \Delta\psi & v_x & v_y & r & \delta & T \end{bmatrix}^T \ . \tag{4.37}$$

The transformation is as follows:

$$\mathbf{x}' = \frac{1}{\dot{s}} f(\mathbf{x}(s), \mathbf{u}(s), s) = \tag{4.38}$$

$$= \frac{1 - \kappa(s)d}{v_x \cos(\Delta\psi) - v_y \sin(\Delta\psi)} \begin{bmatrix} v_x \sin(\Delta\psi) + v_y \cos(\Delta\psi) \\ r - \kappa(s)\dot{s} \\ \frac{F_x}{m} + v_y r \\ \frac{F_y}{m} - v_x r \\ \frac{M_z}{I_{zz}} \\ \dot{\delta} \\ \dot{T} \end{bmatrix} \tag{4.39}$$

$$= f_s(\mathbf{x}(s), \mathbf{u}(s), s) \ , \tag{4.40}$$

where $\mathbf{x}'$ is used to denote $\frac{\mathrm{d}\mathbf{x}}{\mathrm{d}s}$, $\dot{s}$ is defined as in section 4.2, f are the dynamics of the reduced state-vector in (4.37) in time domain, and f_s denote the dynamics sampled in space.

Sampling in space comes with different advantages and disadvantages. Starting from the latter, spatial formulation of the dynamics introduce a numerical singularity when the car is stopped, i.e. $\dot{s} = 0$, making it not possible to run MPC problems from stationarity. Secondly, the dynamics nonlinearity is increased by the pre-multiplication of the factor $\frac{1}{\dot{s}}$. Nonetheless, it also comes with important advantages. Note that the state was reduced, which is specially important for the MPC framework, representing a reduction of overall problem dimension that can lead to faster solutions. Furthermore, recall the path is described by its curvature. If sampling in time, curvature is a function of the state s whereas when sampling in space curvature becomes a function of the independent variable, which avoids problems when integrating the dynamics. This is specially useful for long horizon MPC problems.

5

Track Modelling

As illustrated in Fig. 4.1, the vehicle position is defined relative to a given path. The present section describes the definition of such path and how to obtain its curvature from XY measurements, necessary for the curvilinear coordinate system representation, as well as how does the vehicle localizes itself in relation to the path since the available sensors cannot directly measure the states s, d, and $\Delta\psi$.

5.1 Track description

Assuming the starting position $(x_{s,0}, y_{s,0})$ and heading (ψ_s) of the path in the inertial frame are known, its coordinates in the same frame relate to the curvature $\kappa(s)$ along the path s through the following kinematic expression:

$$x'_s = \cos(\psi_s) \tag{5.1}$$
$$y'_s = \sin(\psi_s) \tag{5.2}$$
$$\psi'_s = \kappa(s)\ . \tag{5.3}$$

Analytical expressions can be implemented for such approach. E.g. say that one parameterizes the path around a certain point and that the parameterization $\gamma(s) = (x_s(s), y_s(s))$ is twice differentiable. Then, the formal definition of curvature could possibly be applied:

$$\kappa = \frac{x'y'' - y'x''}{(x'^2 + y'^2)^{\frac{3}{2}}}\ .$$

It is however cumbersome to find the parameterization γ and could result in nonsmooth variation of the curvature.

Alternatively, the "three-points circle" approach can instead by applied. This approach involves selecting three neighbouring points of the path and defining a circle that includes all those points. In that case, the curvature would simply be $k = \frac{1}{R}$, where R is the radius of the fitted circle. This approach is much simpler than the previous one, resulting however in very noisy variation of the curvature.

Instead, one can make use of the kinematic system of equation (5.1) - (5.3) and estimate the curvature by solving an optimization problem that fits XY coordinates. This way, proper penalty on curvature rate can be set making it smoother along the path. The state space formulation for this optimization problem is as follows:

$$\mathbf{x} = \begin{bmatrix} x_s & y_s & \psi_s \end{bmatrix}^T \tag{5.4}$$
$$u = \kappa\,, \tag{5.5}$$

where

$$\mathbf{x}' = \begin{bmatrix} \cos(\psi_s) \\ \sin(\psi_s) \\ \kappa \end{bmatrix} . \tag{5.6}$$

The optimization problem is therefore written as:

$$\min_{\substack{\mathbf{x}_{s,0},u_0,\mathbf{x}_{s,1},u_1,\dots,\\ \mathbf{x}_{s,L-1},u_{L-1},\mathbf{x}_{s,L}}} = \sum_{k=0}^{L-1} (x^r_{s,k} - x_{s,k})^2 + (y^r_{s,k} - y_{s,k})^2 + q_\kappa(\kappa_{k+1} - \kappa_k)^2 \tag{5.7}$$

$$+ (x^r_{s,L} - x_{s,L})^2 + (y^r_{s,L} - y_{s,L})^2 \tag{5.8}$$

$$\text{s.t.} \quad \mathbf{x}_{s,0} - \mathbf{x}^r_{s,0} = 0 \tag{5.9}$$

$$\mathbf{x}_{k+1} - f_\kappa(\mathbf{x}_k, u_k) = 0, \quad \forall k \in [0, L] \tag{5.10}$$

where L is the number of samples along the entire path, f_κ can be obtained applying the Euler method to discretize (5.6), $x^r_{s,k}$ and $y^r_{s,k}$ are the path reference XY coordinates, and q_k is the penalty on the curvature rate whose tuning allows for trading offset in XY for curvature smoothness. Note that constraint (5.9) is not strictly necessary since it is also possible that the initial measurement is either noisy or inaccurate. Optionally, it could be replaced by

$$\mathbf{x}_{s,f} - \mathbf{x}_{s,0} = 0 \ , \tag{5.11}$$

which instead ensures periodicity in the fitted path. The problem is then solved with the IPM in CasADi for convenience and due to its large dimension.

5.2 Localization and orientation

Measuring the state vector is constrained to what signals are available. Commonly, GPS and IMU measurements are available in the vehicle, providing the controller with full information, although noisy, of the vehicle pose, velocity and acceleration. When using the curvilinear coordinate system, instead, the states d and $\Delta\psi$ need to be obtained from the aforementioned signals. Such problem can be denoted as location and orientation given the reference path.

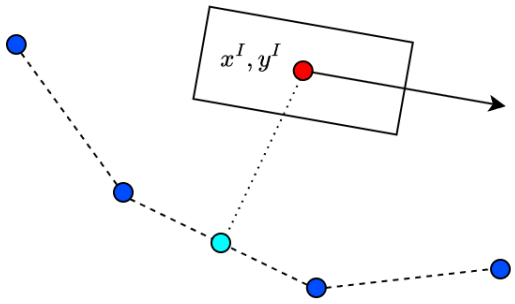


Figure 5.1: Projection onto the line segment between the nearest neighbours of the vehicle XY position.

The designed approach is as follows. At every sampling instant, the controller uses the longitudinal and lateral global position of the vehicle to determine how far it has travelled

along the path. In detail, this is done by a brute search method to find the nearest neighbours based on the Euclidean distance between the pair of current position (x, y) and all the pairs in the neighborhood. Once the nearest neighbours are found, the center of mass of the vehicle is projected onto the line segment that unites them, as illustrated in Fig. 5.1. At this point, having located the vehicle in the path, linear interpolation is applied to obtain d and ψ_s, the latter yielding subsequently the value of $\Delta\psi$. It is, however, important to highlight that the linear interpolation can lead to slightly nonsmooth variation of the estimated values. For that reason, when implementing this module, a low pass filter is used to mitigate this effect.

6
Simulation results

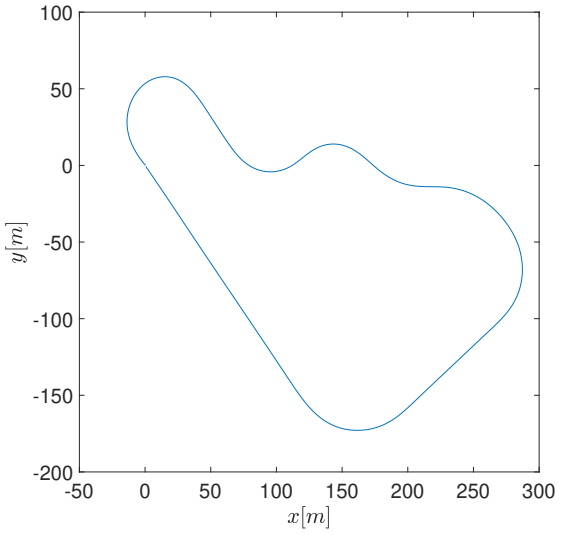
This section presents the results from simulations in a high-fidelity simulation environment, IPG CarMaker. Firstly, in order to speed up the tuning process of the controller, both in terms of selection of the penalisation of the different terms in the cost function and the update rate, the controller was tested in a lower fidelity plant consisting of a double-track chassis model [4] with nonlinear Pacejka's tire model [26] and load transfer [42], with similar parameters to the ones in IPG CarMaker. Once selected the tuning, results were then taken with the high-fidelity model to simulate real vehicle testing. It is also important to state that all results are taken from a laptop computer with the Intel i9-9880H processor with 2.30GHz clock frequency and 32GB of RAM.

Two control strategies were implemented and simulated. First, the controller was tuned for path tracking, i.e. the vehicle was controlled to track the center line of the road and a reference velocity profile. Here, the basic performance of the controller implementation was validated and the performance of two different software and respective numerical solvers for real-time applications was assessed. Furthermore, still for this first application, results are also presented with the so called real time iteration variant of sequential quadratic programming, where the trade-off between optimality and computational speed is discussed. Then, building on the findings from the previous tests, the most promising software was chosen and simulations were repeated this time for energy-optimal trajectory planning, meaning that deviations from the center line and reference velocity are allowed to prioritize energy consumption minimisation. The results of this second study, instead, aim to quantify the potential for energy saving, to analyse the applicability of the developed methods for real-time applications, and to validate and describe the limitations of MPC for this application.

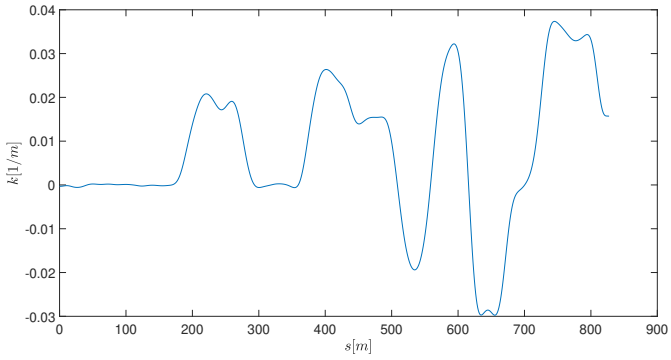
6.1 Test scenario

The test scenario considered to perform simulations consists of a test track and a test vehicle. The test track, illustrated in Fig. 6.1a, is a digital representation of a real test track and consists in a 830m long and 4.6m wide flat road, i.e. with zero-slope. The test track's curvature profile is illustrated in Fig. 6.1b, obtained as explained in Section 5. To illustrate the goodness-of-fit of the results, the deviation from the fitted path coordinates to the reference path is plotted in Fig. 6.1c. In average, the deviations were of less than 5 cm, apart from a higher deviation on the final part of the track, due to accumulated errors and absence of periodicity constraints. It is important to highlight that the curvature of each corner increases sequentially, thus gradually increasing the dynamics that the vehicle has to meet throughout the test track. On the other hand, the test vehicle is illustrated in Fig. 6.2 and has the specifications displayed in Table 6.1. It is important to highlight that the test vehicle has four motors, one per wheel. As stated

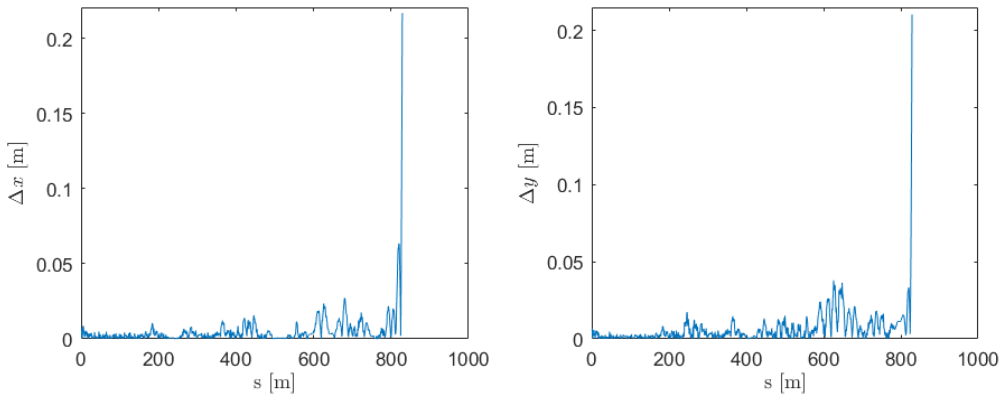
in section 4, the total propulsion torque is minimized. In this implementation, this value is divided in four equal torques which are requested to each of the motors.



(a) XY Path



(b) Curvature



(c) Curvature fitting results

Figure 6.1: Test scenario

Table 6.1: Test Vehicle and Scenario Specifications

Parameter	Value
m	2159 kg
I_{zz}	4858 kgm^2
l_f	1.52 m
l_r	1.22 m
$\delta_{\max}$	0.6872 rad
$\dot{\delta}_{\max}$	0.5454 rad/s
$a_{x,\max}$	3 m/s^2
$a_{y,\max}$	3 m/s^2

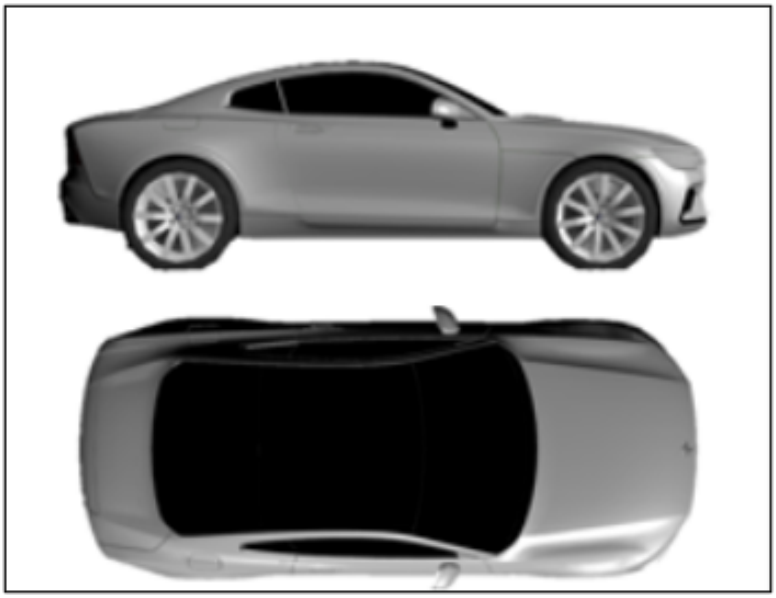


Figure 6.2: Test vehicle.

6.2 Path tracking

For this section in specific the MPC tuning is set as follows, where nomenclature refers to the one presented in Section 3. The cost function is quadratic with terms,

$$l(\mathbf{x}, \mathbf{u}, \epsilon, s) = (\mathbf{x} - \bar{\mathbf{x}}(s))^T Q(\mathbf{x} - \bar{\mathbf{x}}(s)) + \mathbf{u}^T R\mathbf{u} + \epsilon^T Z\epsilon \tag{6.1}$$

$$m(\mathbf{x}, \epsilon, s) = (\mathbf{x} - \bar{\mathbf{x}}(s))^T Q_f(\mathbf{x} - \bar{\mathbf{x}}(s)) + \epsilon^T Z\epsilon \ , \tag{6.2}$$

where Q, R, Z_s, and $Q_f = Q$ are diagonal matrices with penalties in the following terms

$$Q_{1,1} = 10^4,\ Q_{3,3} = 10^1,\ q_{\dot{\delta}} = 10^3,\ q_{\dot{T}} = 10^{-4},\ Z_{1,1} = 10^3,\ Z_{2,2} = Z_{3,3} = 10^2 \tag{6.3}$$

The feasible set is constrained with a set of box and nonlinear constraints, with the respective slack variables,

$$\delta(s) \in [-\delta_{\max}, \delta_{\max}] \tag{6.4}$$
$$\dot{\delta}(s) \in \left[-\dot{\delta}_{\max}, \dot{\delta}_{\max}\right] \tag{6.5}$$
$$d(s)^2 \leq d_{\max} + \epsilon_1(s) \tag{6.6}$$
$$a_x(s)^2 \leq a_{x,\max}^2 + \epsilon_2(s) \tag{6.7}$$
$$a_y(s)^2 \leq a_{y,\max}^2 + \epsilon_3(s)\ , \tag{6.8}$$

where $d_{\max}$ is the maximum track width after subtracting the vehicle width and $a_{x,\max}$ and $a_{y,\max}$ are the maximum longitudinal and lateral acceleration defined in Table 6.1, all the constrains being applied $\forall s \in [s(0), s(N)]$.

Conversely to the constraints on steering actuation and road boundaries, which come from physical characteristics of the system, the constraints on acceleration are included to guarantee that the controller dynamic model is valid within the vehicle's operating region. It should be recalled that the controller considers a single-track chassis and linear tire model, which has limited capacity to represent lateral load transfer and tire forces in the presence of high slip angles and ratios. By constraining the accelerations, the vehicle is constrained to operate in the nearly linear region, as illustrated in Fig. 4.3. Although there are other remaining model mismatches, the model mismatch on tire model assumes a special importance since the simplified model considers unlimited available force from tires as a function of slip, which can lead to divergence issues when near the real force saturation limit.

Another important point to highlight from the formulation above is that, for this application, the penalisation on steering rate, $q_{\dot{\delta}}$, is particularly high. Recall that the only objective at this stage is the tracking of the center of the path and the reference velocity profile. Nevertheless, the cost function itself does not contain a clear incentive on smooth driving and performance in terms of vehicle dynamics such as energy or slip related terms. Without it, the solver can fully exploit input actuation to attain the (local) minimum at each iteration, which can result in noisy actuation even if informative initial guesses are provided. For that reason, R was chosen in order to guarantee smooth actuation across all test scenarios. Still referring to weight selection, Q_f is set as $Q_f = Q$ and the target set is equal to the feasible set. More emphasis could have been put on studying the proper definition of these quantities. However, for simplification reasons, it is let as future work and possible directions as described later in Section 7.

Regarding the selection of the prediction horizon, it is critical that the look-ahead distance, S_f, is chosen as further as possible so that the vehicle can understand the entire corner profile as soon as possible. There are, however, two limitations. Firstly, the study should remain realistic, meaning that very long horizons that are not feasible from the sensing perspective should be avoided. Secondly, to avoid large sampling intervals, Δs, which can be problematic for the dynamics discretization, the horizon length needs to increase with the look-ahead distance of the controller. That implies longer computational times and higher dimensionality of the problem, representing a trade-off between look-ahead distance and computational speed and robustness. The update rate of the controller, look-ahead distance in meters, S_f, and the horizon length, N, with which the following results of this section were taken are as presented in Table 6.2.

Finally, the reference state vector, $\bar{\mathbf{x}}(s)$ is set as

$$\bar{\mathbf{x}}(s) = \begin{bmatrix} 0 & 0 & \bar{v}_x(s) & 0 & 0 & 0 & 0 \end{bmatrix}$$

Table 6.2: Path tracking MPC tuning of update rate and prediction horizon

Software (Solver)	S_f [m]	N	Update rate
casADi (IPM)	50	25	5 Hz
acados (SQP)	50	25	20 Hz
acados (SQP-RTI)	50	25	40 Hz

where $\bar{v}_x(s)$ is the reference longitudinal velocity. For high velocities the vehicle's lateral acceleration on the chosen test scenario, assuming perfect tracking of the center of the line, can reach values very much higher than 3 m/s^2. One can take the example of the last corner, with the highest curvature, and assume steady state cornering conditions with $v_x = 70$ km/h. For this road geometry and vehicle velocity, the vehicle would be subject to a lateral acceleration of slightly over 10m/s^2. Since the vehicle's acceleration is constrained to 3 m/s^2, in order to attain feasibility in the optimization problem, the vehicle's trajectory needs either to deviate from the path's center line to reduce the curvature of its trajectory or reduce the longitudinal velocity, which will equivalently decrease lateral acceleration. Note that the former is not desirable for the path tracking application, meaning that longitudinal control is then the remaining degree of freedom to ensure the feasibility of trajectories. Given that fast computations of MPC are highly dependent on the initial guess and how close it is to the optimum, in order to enhance the solver performance, $\bar{v}_x$ is therefore updated for each node of the horizon as follows:

$$\bar{v}_x(s) = \min \left\{ \bar{V}_x, \sqrt{\frac{a_{y,\max}}{\|\kappa(s)\|}} \right\} , \tag{6.9}$$

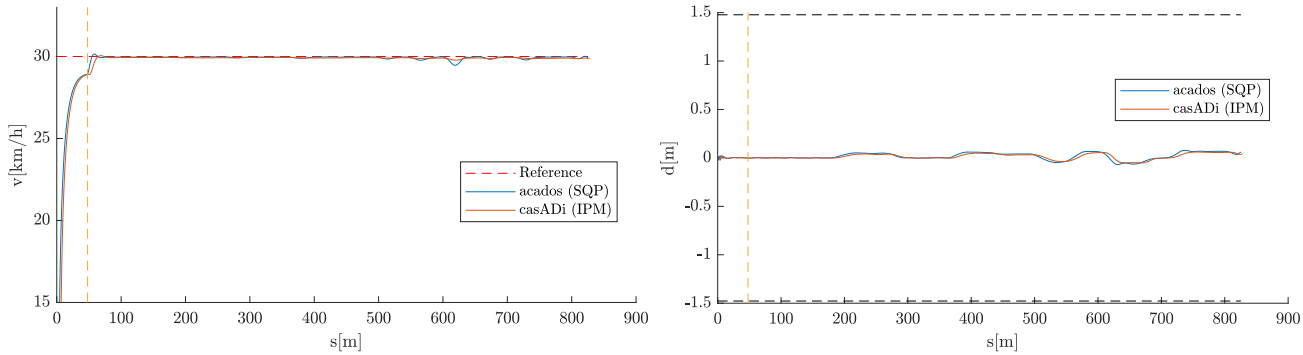
where $\bar{V}_x$ represents the global reference velocity, set as constant for each test. In such manner, assuming the vehicle tracks the path, the reference acts as offline prior knowledge of the trajectory described by the vehicle and computational efficiency is enhanced.

6.2.1 Comparison between casADi (IPM) and acados (SQP)

Results for two velocities are presented to illustrate two different testing scenarios: $\bar{V}_x = 30$ km/h and $\bar{V}_x = 70$ km/h, simulating smooth and more dynamic driving, respectively. The test procedure is as follows. Due to the spatial formulation, MPC cannot be run at zero or near-zero longitudinal velocity. Thus, the IPG CarMaker Driver Model, which can control both longitudinal and lateral vehicle dynamics, is initially activated and drives the vehicle until it reaches the reference longitudinal velocity. After that, MPC is activated and assumes the control for the rest of the simulation time. To highlight the MPC triggering time instant, all plots below contain a vertical line indicating it.

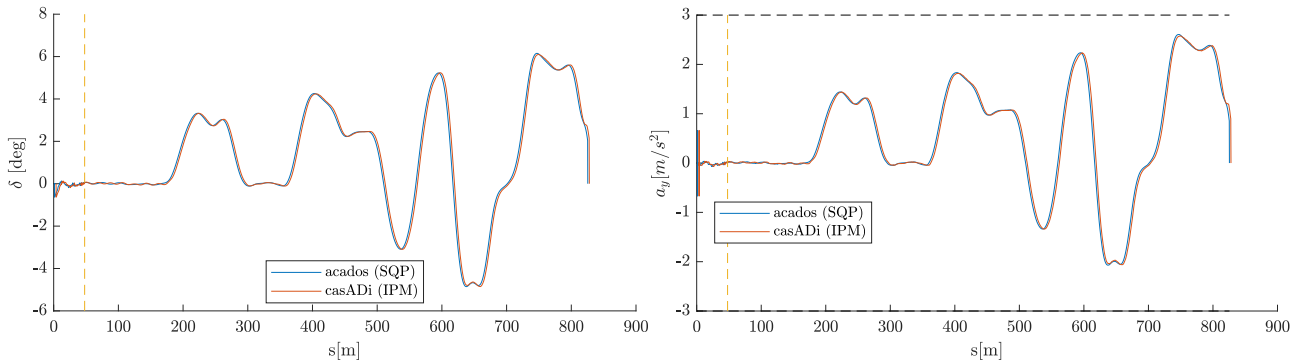
Starting with results at $\bar{V}_x = 30$ km/h, those are presented in Fig. 6.3. Note that the tracking is quite accurate in both metrics. No relevant differences between implementations can be observed, mainly due to the fact that the vehicle is not operating near the constrains and the velocity profile is constant. Then, Fig. 6.4 and 6.5 present an analysis on lateral and longitudinal control, respectively. Smooth performance is observed in both figures and for both implementations, as expected. Finally, Fig. 6.6 presents a histogram of computational time per feedback loop iteration. Here, conversely, the difference among solvers is significant, acados performing approximately 10 times faster than casADi. As discussed in Section 3, this difference is as expected. The casADi solver, IPOPT, is proper for large-scale nonlinear programming,

whereas acados employs the HPIPM solver for each of the QP from the SQP algorithm, which is tailored for efficiently solving small to medium sized QP arising from optimal-control problems. Therefore, this being a medium size problem, in specific having $N(n_x + n_s) + (N-1)n_u = 20 \times (7+3) + 19 \times 2 = 238$ variables, it is then possible to argue that casADi slower performance is as expected.



(a) Longitudinal velocity **(b)** Lateral displacement

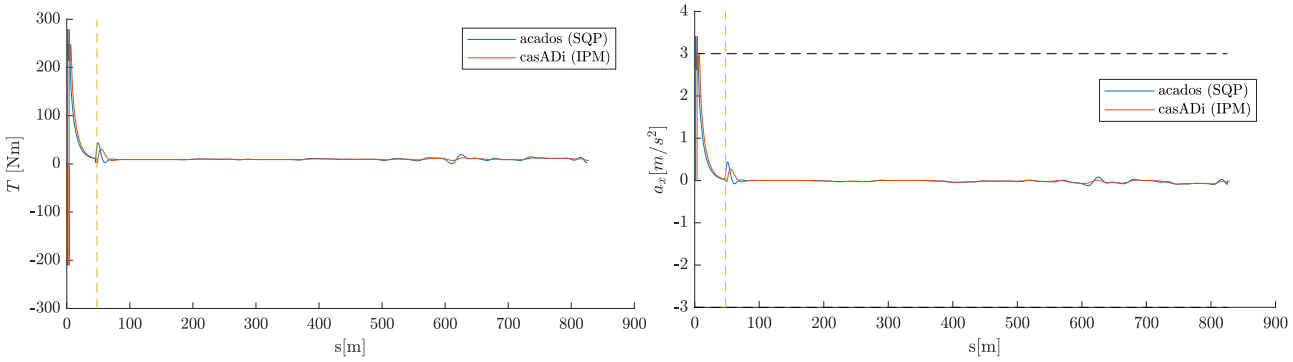
Figure 6.3: IPM vs SQP: Reference tracking at $\bar{V}_x = 30$ km/h.



(a) Steering angle **(b)** Lateral acceleration

Figure 6.4: IPM vs SQP: Lateral dynamics at $\bar{V}_x = 30$ km/h.

Next, more dynamic maneuvers were tested by setting $\bar{V}_x = 70$ km/h. The same set of results as for the previous case is now presented. Note that the tracking results in Fig. 6.7 have now a more dynamic behaviour, specially the longitudinal velocity where the reference is now constrained as in (6.9). Such reference is not accurately tracked along the entire path, by none of the implementations. This is due to the longitudinal acceleration constraint, as illustrated in Fig. 6.9b. Looking at the section of the path around 200m, relative to the first corner as it can be understood from Fig. 6.1, the controller brakes in advance in order to respect the -3 m/s^2 constraint on longitudinal acceleration. This is thus as expected and validates the implementation of the controller. Regarding the tracking of the center of the path, the vehicle now faces a more difficult scenario, where longitudinal dynamics and lateral dynamics come simultaneously into play. At the expense of worse tracking of reference longitudinal velocity in corners, see Fig. 6.7a, casADi implementation yields less deviation from the center of the



(a) Motor torque **(b)** Longitudinal acceleration

Figure 6.5: IPM vs SQP: Longitudinal dynamics at $\bar{V}_x = 30$ km/h.

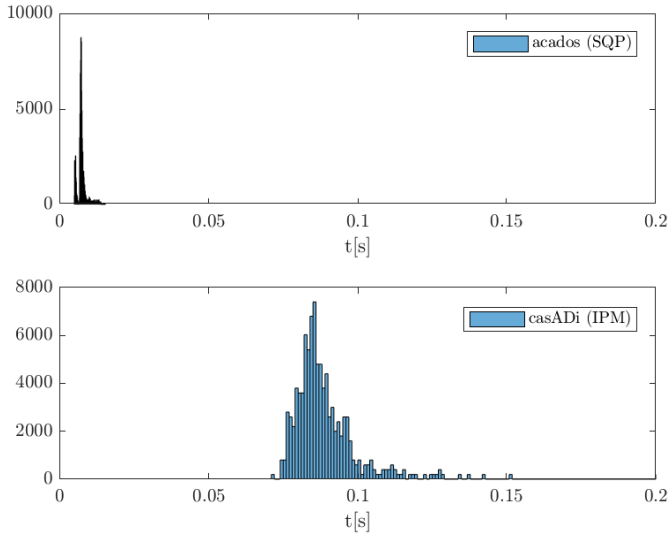
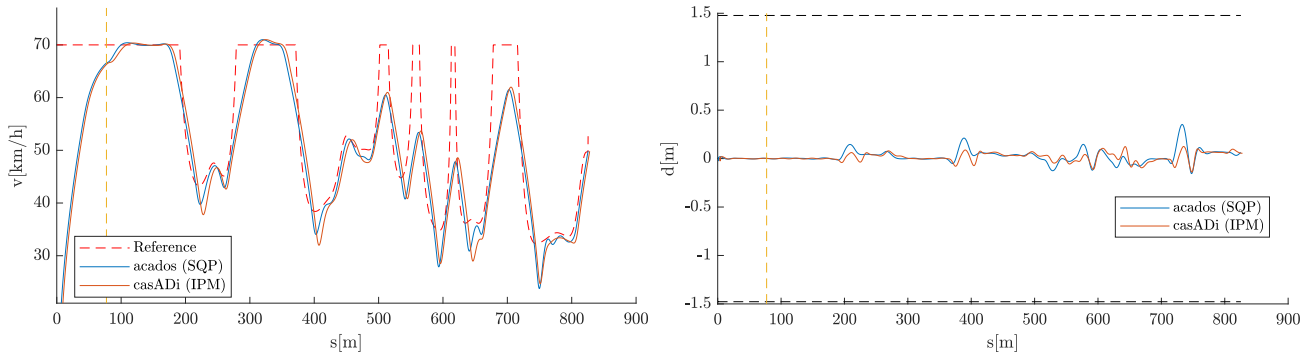


Figure 6.6: IPM vs SQP: Computation time histogram at $\bar{V}_x = 30$ km/h

path than its counterpart. Conversely, results with acados yielded better tracking in velocity, at the expense of more corner cutting. However, in overall, one can say the solutions are rather similar in terms of reference tracking.

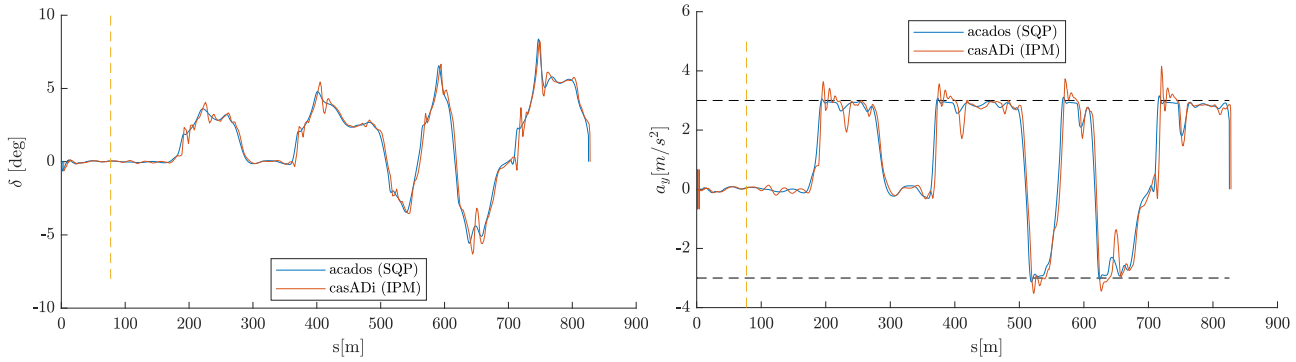
Instead, from the vehicle dynamics perspective, the implementations differ significantly mainly regarding lateral control. Looking at Fig. 6.8, one can conclude that the casADi implementation yields less stable steering angle usage than its counterpart. This is well understood both in the first straight-road section, before 200 m, as well as when it hits the constraints, as in Fig. 6.8. The same conclusion applies when looking at longitudinal dynamics in Fig. 6.9. Such difference is motivated mainly by the slower update rate of the casADi implementation, since, in case of model mismatch, too slow update rates can make the system's feedback loop too slow to compensate from deviation from the reference leading to an oscillatory behaviour.

Comparing both scenarios, one can then conclude that acados is better suited for this problem given the faster computational time and, consequently, the faster controller update rate.



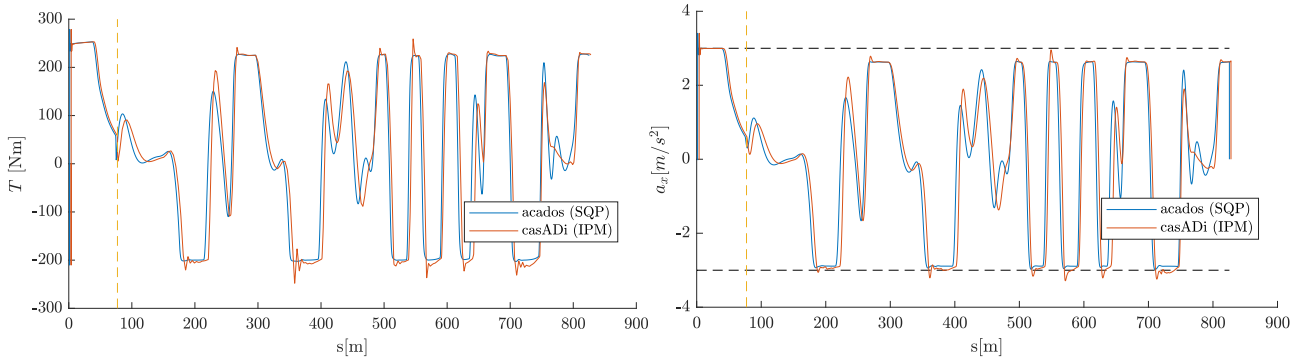
(a) Longitudinal velocity **(b)** Lateral displacement

Figure 6.7: IPM vs SQP: Reference tracking at $\bar{V}_x = 70$ km/h.



(a) Steering angle **(b)** Lateral acceleration

Figure 6.8: IPM vs SQP: Lateral dynamics at $\bar{V}_x = 70$ km/h.



(a) Motor torque **(b)** Longitudinal acceleration

Figure 6.9: IPM vs SQP: Longitudinal dynamics at $\bar{V}_x = 70$ km/h.

6.2.2 Comparison between SQP and RTI

The present thesis has the main focus of achieving fast computational times in the computation of MPC problems. In the previous section, acados results achieved the promising mark of around 10 ms per iteration, in average. This section builds on top of that and presents a

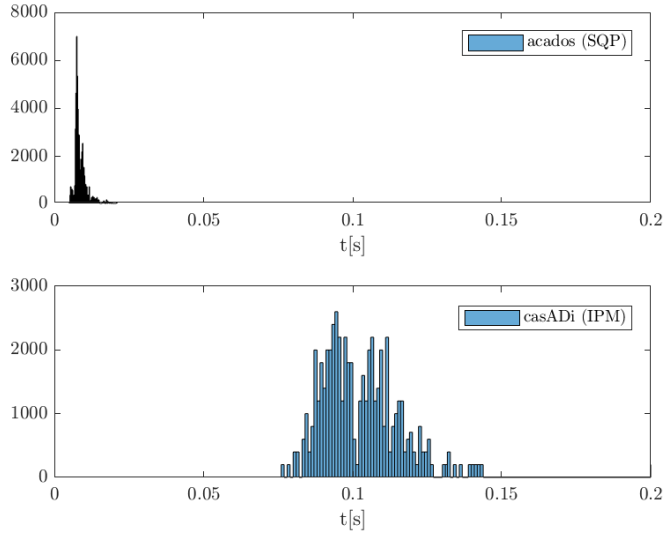


Figure 6.10: IPM vs SQP: Computation time histogram at $\bar{V}_x = 70$ km/h

comparison between SQP and RTI as solver methods and what's the trade-off between optimality and computational time. Recall RTI represents essentially the same solving method as SQP, being the solution of the first QP of the SQP method. It is therefore expected that RTI is considerably faster.

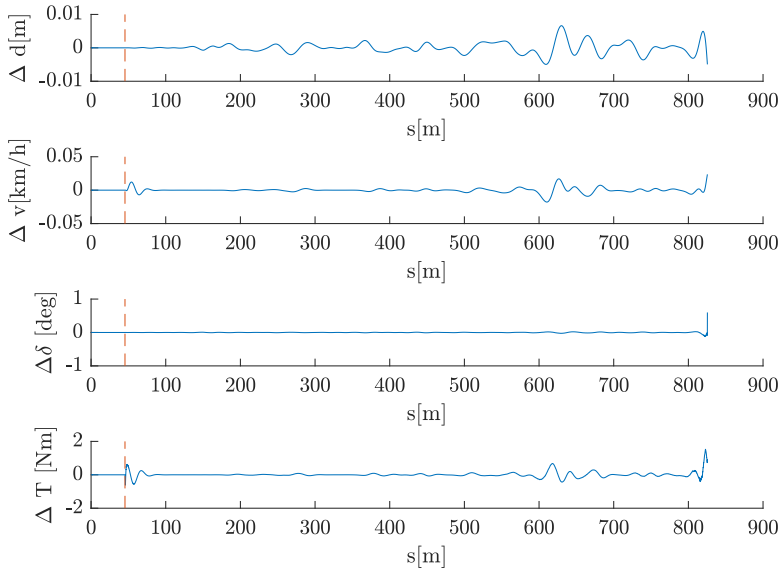


Figure 6.11: Comparison between SQP and RTI solutions at $\bar{V}_x = 30$ km/h

The results obtained are similar between both implementations. For that reason, this section presents the results in the form of the absolute deviation of the RTI solution relative to the SQP one, illustrated in Fig. 6.11 and 6.13, for $\bar{V}_x = 30$ km/h and $\bar{V}_x = 70$ km/h, respectively. Looking at both figures, one concludes that deviations are rather small, in both cases, apart from a slight difference in longitudinal control at $\bar{V}_x = 70$ km/h. The great advantage of this method is thus the reduced average computation time per iteration without major loss of

optimality. In the test bench used, which the authors are aware to be of very high computational power, RTI computational times were below 5ms, remarkable for MPC online computations. This presents a possibly viable solution to implement in a real vehicle, where the available computation power is constrained.

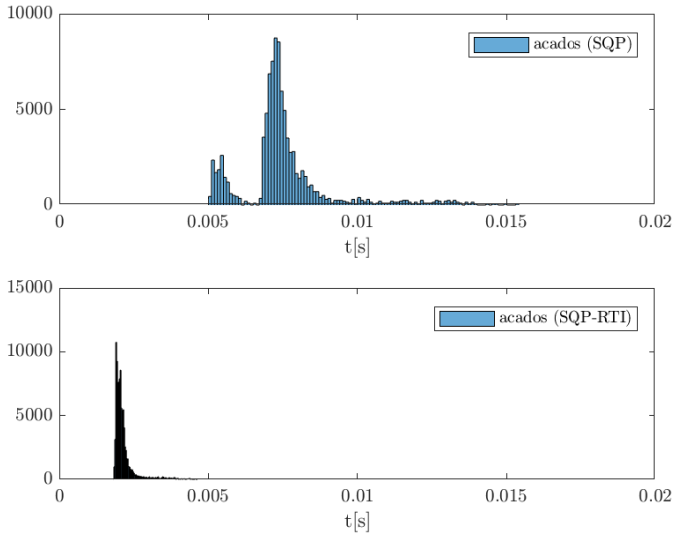


Figure 6.12: Comparison between SQP and RTI computation time histogram at $\bar{V}_x = 30$ km/h

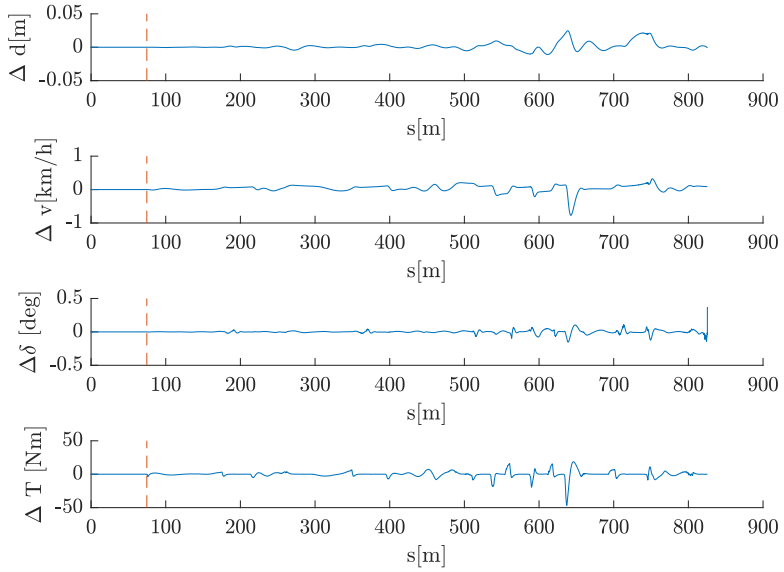


Figure 6.13: Comparison between SQP and RTI solutions at $\bar{V}_x = 70$ km/h

6.2.3 Comparison between MPC and Pure Pursuit Control

For the large majority of lateral vehicle control of nowadays applications, Pure Pursuit Control (PPC) is probably the most commonly implemented algorithm. Its main advantage is

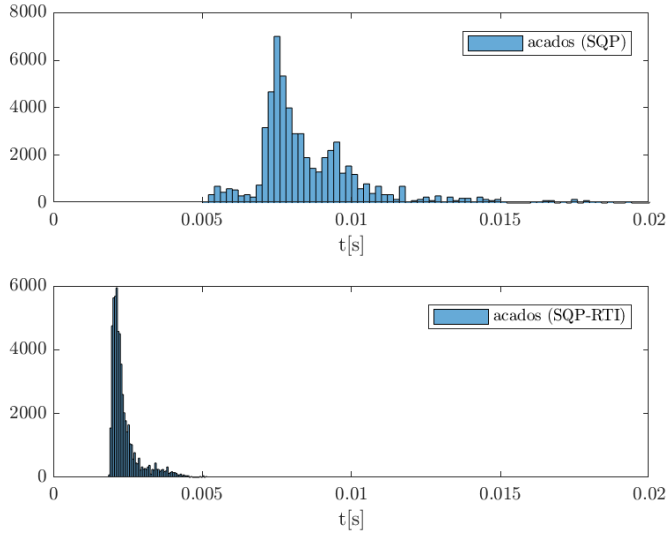


Figure 6.14: Comparison between SQP and RTI computation time histogram at $\bar{V}_x = 70$ km/h

its computational and algorithmic simplicity and capacity of delivering smooth performance. In essence, PPC works as follows. A look-ahead time is defined and based on the current velocity and the path coordinates, a point ahead in the path is selected as the target point of the maneuver at a certain sampling instant. Then, based on that, a PI-type controller can be set to drive the vehicle to this reference point.

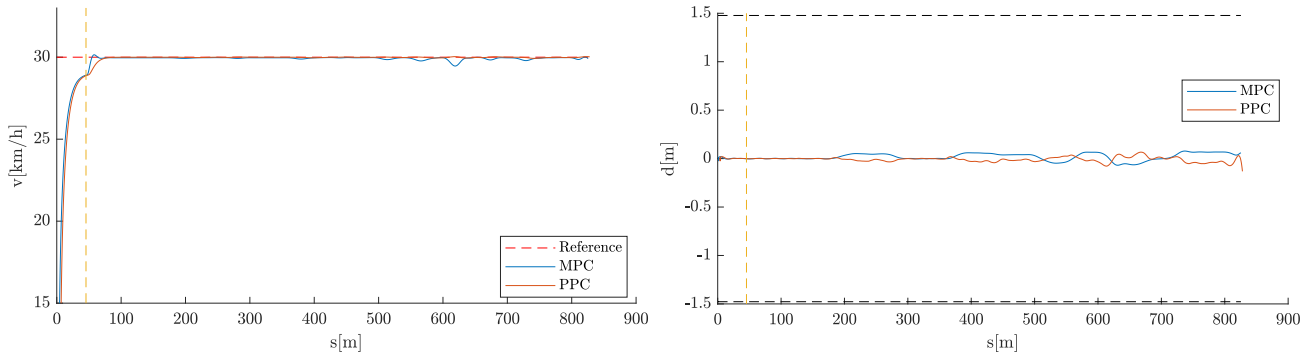
Previous studies at Volvo Cars Cooperation implemented PPC for lateral control, e.g. [34] and [16]. For longitudinal control, when tested in the IPG CarMaker environment, these works used its embedded Driver Model responsible to define and track the reference longitudinal velocity. The velocity planning algorithm relies on constraining the speed profile for the full track, i.e. it considers global a priori knowledge of the test scenario. It first saturates the initial $\bar{V}_x$ as it is done in (6.9) according to the constraints on acceleration, and then smooths the curve to attain comfort driving, given jerk constraints. Then, the new reference is tracked with a PI-type of controller. For this test, although MPC does not include jerks, the limits on acceleration are set equivalently on both implementations.

Conversely to PPC, MPC optimises the longitudinal velocity based on a limited prediction horizon, which is therefore sub-optimal since only limited information is available. To understand how MPC performance compares with PPC, results are presented below analogously to the aforementioned ones, and, for simplicity of analysis, only the acados (SQP) implementation is presented.

The analysis is in every sense analogous to what was previously presented, the plotting structure and organization being the same. Fig. 6.15 to 6.17 present the analysis at $\bar{V}_x = 30$ km/h while Fig. 6.18 to 6.20 are relative to $\bar{V}_x = 70$ km/h.

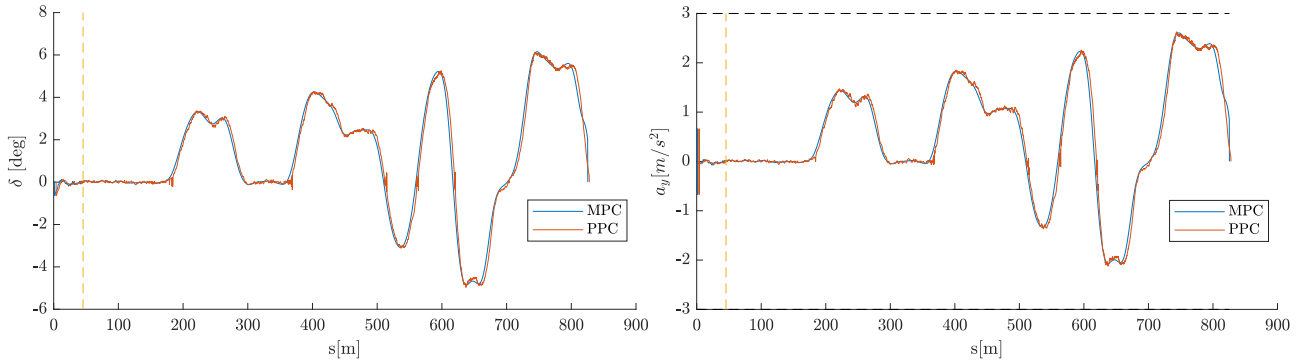
In the normal driving scenario, at $\bar{V}_x = 30$, the results of both control strategies are similar both in terms of tracking and in terms of vehicle dynamics performance. One important point to highlight is the slight difference in lateral displacement from the center of the track, in Fig. 6.15a. Note that MPC results deviate to the inner side of every corner, given its penalisation on steering rate. Therefore, given its predictive and optimal nature, it presents this minimal corner

cutting effect. Conversely, PPC acts like a PI-type of controller, and therefore deviations are not of the same type. Nevertheless, it is important to note that the magnitude of such deviations is small. On another point, PPC results present a noisier steering actuation, see Fig. 6.16. This is mainly due to the nature of the controller, acting proportionally on the lateral displacement, without any penalisation on steering rate. Common filtering strategies like low-pass filtering would surely solve this problem, however results are shown in their raw form to highlight the difference between methods.



(a) Longitudinal velocity **(b)** Lateral displacement

Figure 6.15: MPC vs PPC: Reference tracking at $\bar{V}_x = 30$ km/h.

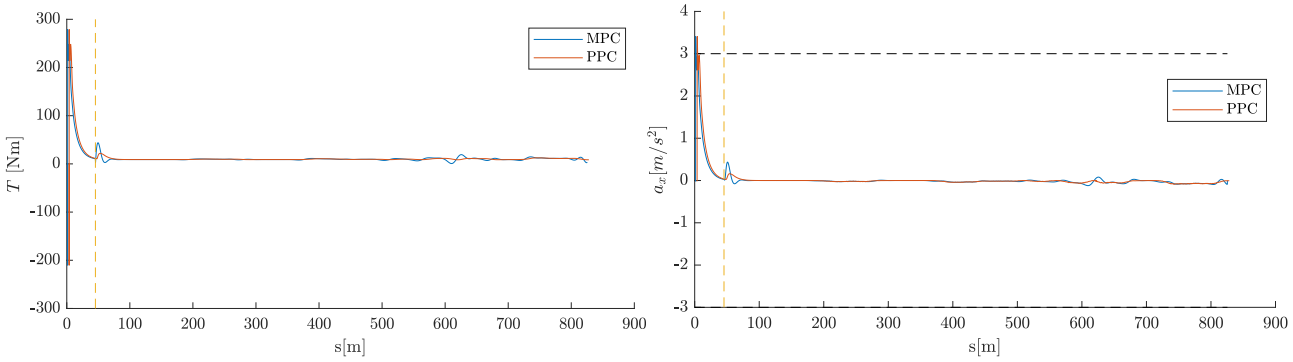


(a) Steering angle **(b)** Lateral acceleration

Figure 6.16: MPC vs PPC: Lateral dynamics at $\bar{V}_x = 30$ km/h.

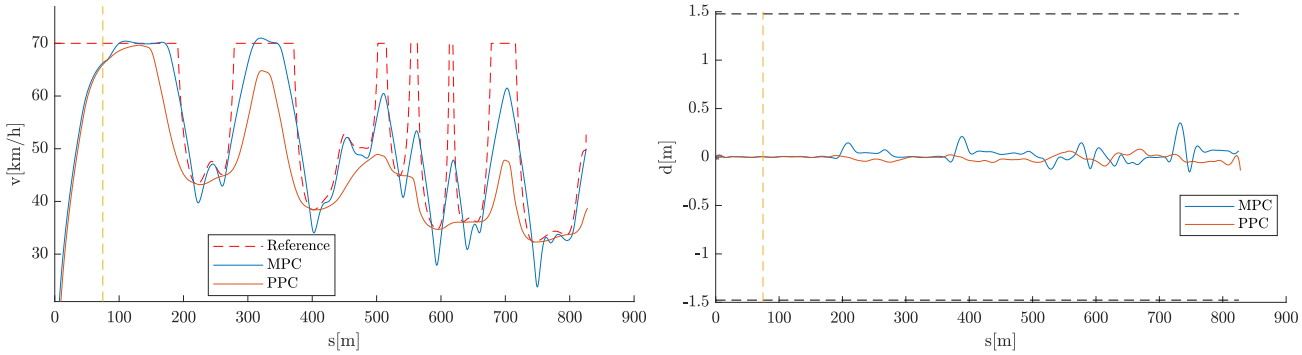
In the more dynamic driving scenario, at $\bar{V}_x = 70$, the results of both control strategies are now contrasting from different points of view. Starting from the reference tracking results, in Fig. 6.18, it is clear that the PPC longitudinal control is much more conservative than the MPC one. This implies, as illustrated in Fig. 6.19, that lateral control in corners is simplified from the vehicle dynamics perspective, resulting in smoother steering angle actuation and lower lateral accelerations.

In sum, MPC performed equivalently to PPC in the tracking of center of the path, with solely small deviations. The two implementations had different longitudinal control strategies, therefore a fair comparison between tracking of the reference longitudinal velocity is not possible. Nevertheless, it is possible to conclude that MPC solutions, being sub-optimal due to the



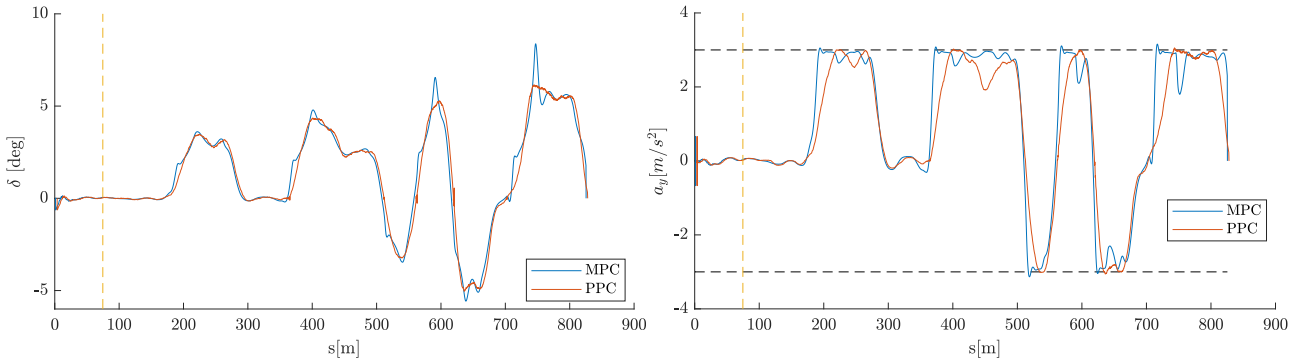
(a) Motor torque **(b)** Longitudinal acceleration

Figure 6.17: MPC vs PPC: Longitudinal dynamics at $\bar{V}_x = 30$ km/h.



(a) Longitudinal velocity **(b)** Lateral displacement

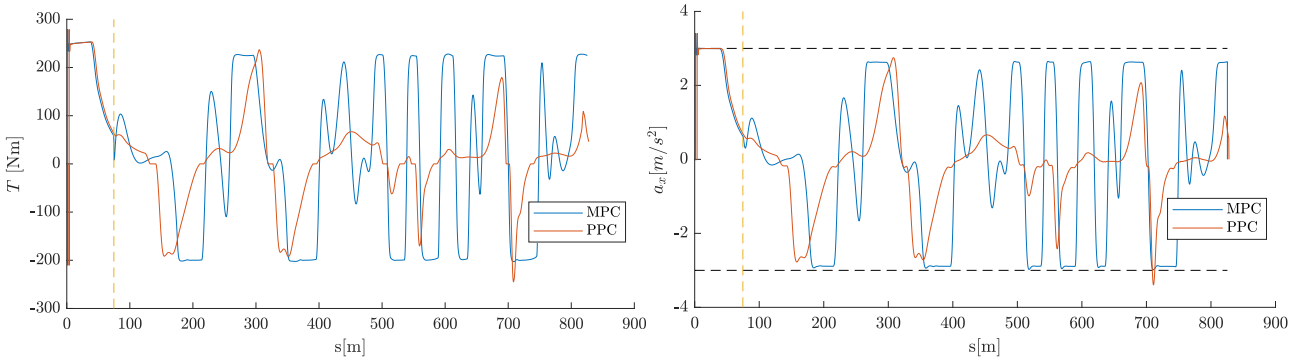
Figure 6.18: MPC vs PPC: Reference tracking at $\bar{V}_x = 70$ km/h.



(a) Steering angle **(b)** Lateral acceleration

Figure 6.19: MPC vs PPC: Lateral dynamics at $\bar{V}_x = 70$ km/h.

limited prediction horizon and having a more aggressive longitudinal control strategy, were still able to provide a good tracking of the center of the path. The method thus proves powerful to combine longitudinal and lateral control for the case of limited a priori knowledge of the entire track, as it is the case for autonomous driving applications. Furthermore, if comfort driving is to be achieved in the MPC implementation, acceleration penalisation can be included, which



(a) Motor torque **(b)** MPC vs PPC: Longitudinal acceleration

Figure 6.20: MPC vs PPC: Longitudinal dynamics at $\bar{V}_x = 70$ km/h.

would approximate the solutions of both implementations.

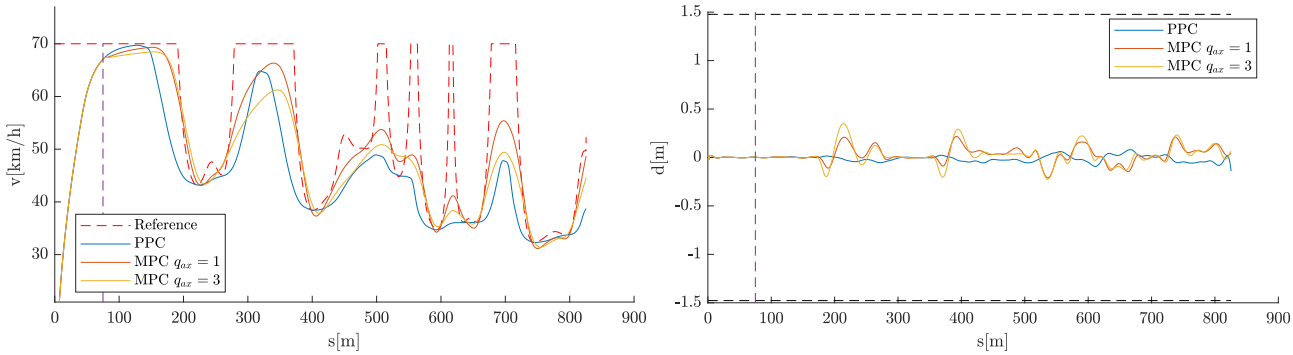
6.2.3.1 Additional penalty for comfort driving

Considering the suggestion at the end of the previous section, additional tests were conducted including a penalty on longitudinal acceleration, with the respective weight q_{ax}, the cost function being now:

$$l(\mathbf{x}, \mathbf{u}, \epsilon, s) = (\mathbf{x} - \bar{\mathbf{x}}(s))^T Q(\mathbf{x} - \bar{\mathbf{x}}(s)) + \mathbf{u}^T R\mathbf{u} + \epsilon^T Z\epsilon + q_{\text{ax}} a_x^2 \tag{6.10}$$

$$m(\mathbf{x}, \epsilon, s) = (\mathbf{x} - \bar{\mathbf{x}}(s))^T Q_f(\mathbf{x} - \bar{\mathbf{x}}(s)) + \epsilon^T Z\epsilon \ . \tag{6.11}$$

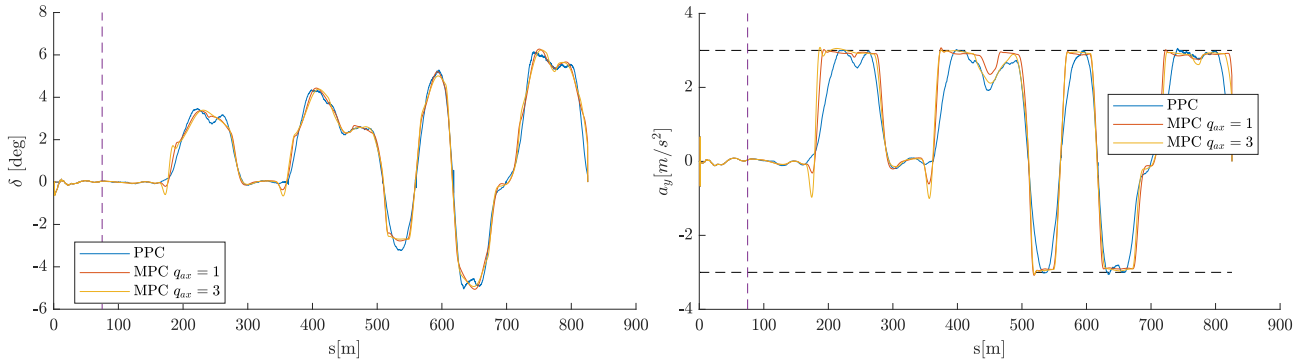
Fig. 6.21 to 6.23 illustrate the obtained results for the most dynamic scenario at $\bar{V}_x = 70$ km/h, considering $q_{\text{ax}} = \{1, 3\}$ for comparison. As expected, the velocity profile is now more similar to PPC than what it was in Fig. 6.18a, being however still more accurate in tracking the reference. Additionally, center of the line tracking is now slightly less accurate, as in Fig. 6.23b, since trajectories now take into account acceleration penalization.



(a) Longitudinal velocity **(b)** Lateral displacement

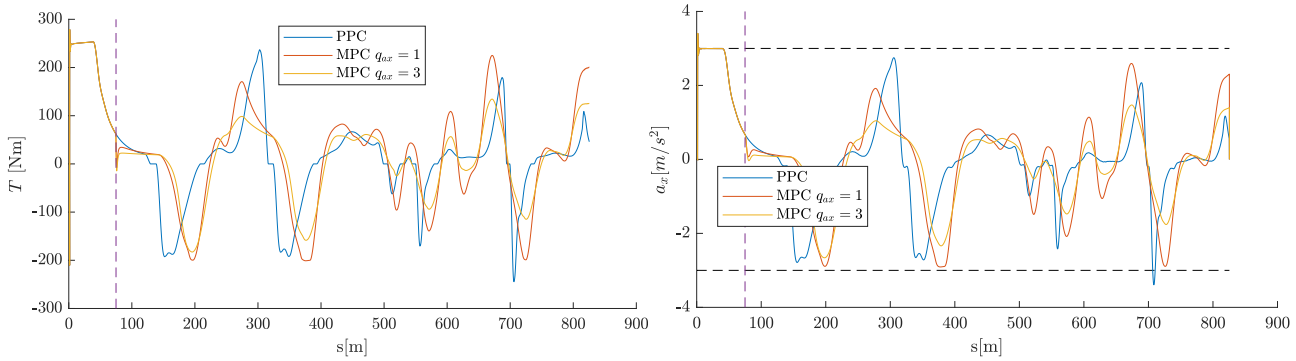
Figure 6.21: MPC (with acceleration penalization) vs PPC: Reference tracking at $\bar{V}_x = 70$ km/h.

These results prove the possibility of tuning MPC controllers to mimic currently implemented methods in vehicles, a feature that is not available in simpler implementations such as PPC.



(a) Steering angle **(b)** Lateral acceleration

Figure 6.22: MPC (with acceleration penalization) vs PPC: Lateral dynamics at $\bar{V}_x = 70$ km/h.



(a) Motor torque **(b)** MPC vs PPC: Longitudinal acceleration

Figure 6.23: MPC (with acceleration penalization) vs PPC: Longitudinal dynamics at $\bar{V}_x = 70$ km/h.

6.3 Energy consumption minimization

The problem complexity increases when energy consumption is taken into consideration. In fact, adding energy consumption to the cost function originates another type of control problem: economic (nonlinear) model predictive control (EMPC). The term economic arises from the fact that energy is a physical quantity that describes the cost of vehicle trajectory by itself. The resultant trajectory is thus a balanced trade-off between the consumption of energy and the penalization of other performance terms, such as tracking of reference velocity and path.

EMPC is in general more intricate to solve than problems such as the one presented in the previous section, mainly due to nonlinearities introduced by the presence of economic terms in the cost function. Note that e.g. (6.1) and (6.2) are quadratic convex functions, the nonconvexity of the problem being located only at the equality constraints related to the dynamics of the system. Energy terms will therefore make the cost function more nonlinear and definitely nonconvex. Moreover, recall that the problem can be of high dimension ($N(n_x + n_s) + (N-1)n_u$), specially for large horizons. When solving the problem, computing search directions can become more cumbersome if the solution space is large or if well-defined search

directions cannot be found. In the specific case of energy, different decision variables affect energy consumption in different ways, e.g. speed, lateral displacement, and actuator values, and the solver might need multiple iterations with costly computations to find the optimal search direction for convergence.

Throughout the development of this work, the authors encountered multiple difficulties when tuning the controller parameters, which led to multiple changes in the problem formulation. Firstly, the horizon length was increased to $N = 50$ which proved useful in helping the solver dealing with nonlinearities. The look-ahead distance and the sampling rate were instead kept the same. Then, in order for the cost function terms to not differ from each other by many orders of magnitude, which is oftentimes the case for EMPC, the different terms of the cost function were scaled. Finally, with the assumption that energy optimization smooths out actuation by itself, and in order to avoid that trajectories are chosen mainly dictated by actuation penalization, the penalty on steering rate was decreased. Building upon the conclusions on the previous sections, only the acados (SQP) implementation was considered since the already high computation times of CasADi (IPM) increased to an average of over 0.2s, a too slow time mark for real-time implementation.

The EMPC formulation is now presented. The feasible set is as defined in (6.4)-(6.8). The cost function is set as:

$$l(\mathbf{x}, \mathbf{u}, \epsilon, s) = (\mathbf{x} - \bar{\mathbf{x}}(s))^T Q (\mathbf{x} - \bar{\mathbf{x}}(s)) + \mathbf{u}^T R \mathbf{u} + Q_E E + \epsilon^T Z \epsilon \tag{6.12}$$

$$m(\mathbf{x}, \epsilon, s) = (\mathbf{x} - \bar{\mathbf{x}}(s))^T Q_f (\mathbf{x} - \bar{\mathbf{x}}(s)) + \epsilon^T Z \epsilon \ , \tag{6.13}$$

where E stands for energy consumption and Q_e is an additional tuning penalty of the same term. For the case of EMPC, the tuning differs from test to test and will be presented accordingly. Nevertheless, the matrices Q, R, and Q_e have a common structure among tests and are diagonal matrices with nonzero terms:

$$\begin{aligned}
Q_{1,1} &= \frac{q_d}{\max(d)} \\
Q_{3,3} &= \frac{q_v}{\max(\Delta v_x)} \\
R_{1,1} &= \frac{q_{\dot{\delta}}}{\max\left(\dot{\delta}\right)} \\
R_{2,2} &= \frac{q_{\dot{T}}}{\max\left(\dot{T}\right)} \\
Q_E &= \frac{q_e}{\frac{\bar{v}_x(s)}{R} I T_{\max}} \\
Q_f &= Q \ .
\end{aligned}$$

By dividing each term of the cost function by its maximum value, each term will lie within the range of $[-1, 1]$. This is extremely useful for the solver, as it helps it avoiding ill-conditioned matrices. Then, additional tuning factors, $q_{(\cdot)}$, are considered to define the relative weights between terms. Note as well Q_E varies with the reference speed. This is critical for the case when the reference longitudinal speed profile varies significantly, being necessary to adjust this scaling to not under penalize energy consumption when travelling slower. Regarding the calculation of energy E, the following relation is used given that time is no more the independent

variable neither it is part of the state-space:

$$E = P\Delta t = P\frac{\Delta s}{\dot{s}} \ , \tag{6.14}$$

where

$$P = P_{\text{mech}} + P_{\text{el.loss}} \ . \tag{6.15}$$

On another point, it is relevant to point out that the reference saturation in (6.9) is kept. Note, however, that the center of the path is not anymore such an informative guess of the vehicle trajectory. This means that tracking the reference speed while deviating from the center of the path leads to sub-optimal solutions. It was however decided to keep this saturation to enhance solver computations, with the assumption that, if energy is penalized highly enough, the solver adapts the longitudinal velocity accordingly.

The tests are conducted assuming the solver can take two major directions to optimize energy consumption. Recall the cost function penalizes the deviation from reference longitudinal velocity and lateral displacement. The solver thus needs to compromise one of the tracking objectives, or both, to attain energy savings. For that reason, results will demonstrate what is the individual impact of relaxing each of the tracking objectives and quantify the possible savings each one allows for, and conclude by describing what would be the total saving potential when relaxing both tracking objectives. The forthcoming sections divide the conclusions into energy saving potential in path tracking, i.e. when tracking the center of the line, and trajectory planning, i.e., when deviations from the center line of the path are allowed.

6.3.1 Energy-optimal path tracking

As highlighted throughout the analysis of the results in Section 6.2, the test scenarios involve reference longitudinal velocity profiles that, although complying with the lateral acceleration constraints, still require aggressive longitudinal control. To track such references, it is necessary to heavily penalize deviations from them. If energy minimization is added on top of such tracking performance standards, the margin for energy consumption reduction is low since little compromise is allowed in reference tracking. Instead, if less aggressive maneuvering is desired, the optimization problem can be adapted to take that into account, as Section 6.2.3 highlighted, relaxing reference tracking, mainly the one of longitudinal velocity. The trajectory thus gets not only smoother but also has more freedom to attain the pursued savings. To illustrate the energy saving potential in this case, the reference longitudinal velocity is set as $\bar{V}_x = 70$ km/h.

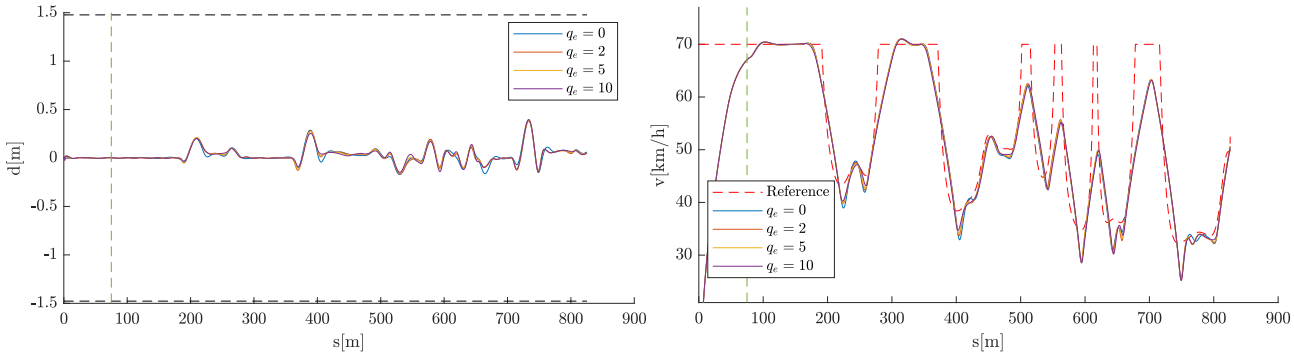
6.3.1.1 Accurate reference tracking

For accurate reference tracking, the cost function is as in (6.12)-(6.13), with tuning:

$$q_d = 10,\ q_v = 1,\ q_e = \{2, 5, 10\},\ q_{\dot{\delta}} = 0.1,\ q_{\dot{T}} = 0.05,\ Z_{1,1} = Z_{2,2} = Z_{3,3} = 10 \ , \tag{6.16}$$

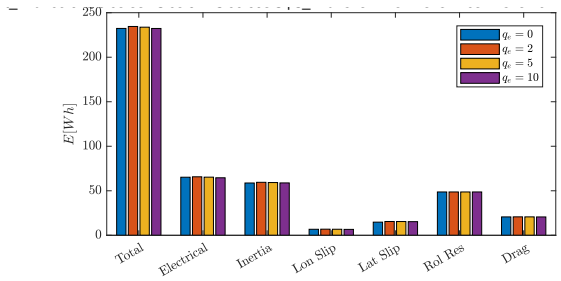
where the range of q_e is chosen as high as possible so that the solver converges successfully within low computational times and the tracking objective is not impacted unreasonably. The results follow. Fig. 6.24 presents a comparison between the energy consumption from the results in Section 6.2.3 and the energy consumption with the tuning presented in this section. The reference longitudinal velocity profile is not significantly different across different q_e, given the

high penalty on deviations from reference. This, in turn leads to no savings, where consumption remains quite similar among all results, as illustrated in Fig. 6.24c and summarized in Table 6.3. In the table, the notation MAE Δv and MAE d hold for mean absolute error of tracking of longitudinal velocity reference, $\Delta v = v_x(s) - \bar{V}_x$, and zero lateral displacement, respectively.

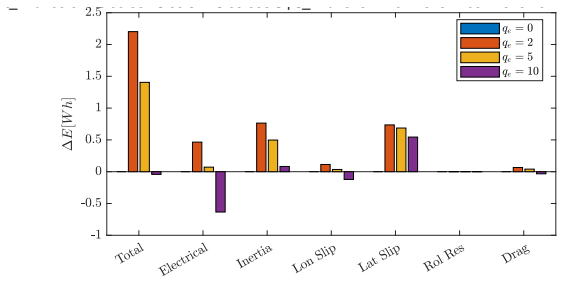


(a) Lateral Displacement

(b) Longitudinal velocity



(c) Total energy consumption



(d) Difference in energy consumption

Figure 6.24: Performance comparison at $\bar{V}_x = 70$ km/h for $q_d = 10$ and $q_e = \{0, 2, 5, 10\}$.

Table 6.3: Key performance indicators at $\bar{V}_x = 70$ km/h for $q_d = 10$ and $q_e = \{0, 2, 5, 10\}$.

	$q_e = 0$	$q_e = 2$	$q_e = 5$	$q_e = 10$
Lap time [s]	66.0	65.9	65.9	66.0
Lap length [m]	823.3	823.3	823.3	823.3
Total energy [Wh]	232.3	234.5	233.7	232.3
Relative energy saving [%]	-	-0.9	-0.6	0.02
MAE Δv [km/h]	25.1	25.1	25.1	25.1
MAE d [m]	0.06	0.05	0.05	0.05
Mean computational time [ms]	7.7	15.0	15.2	15.2
Max computational time [ms]	15.0	22.5	23.4	21.4

6.3.1.2 Additional penalty for comfort driving

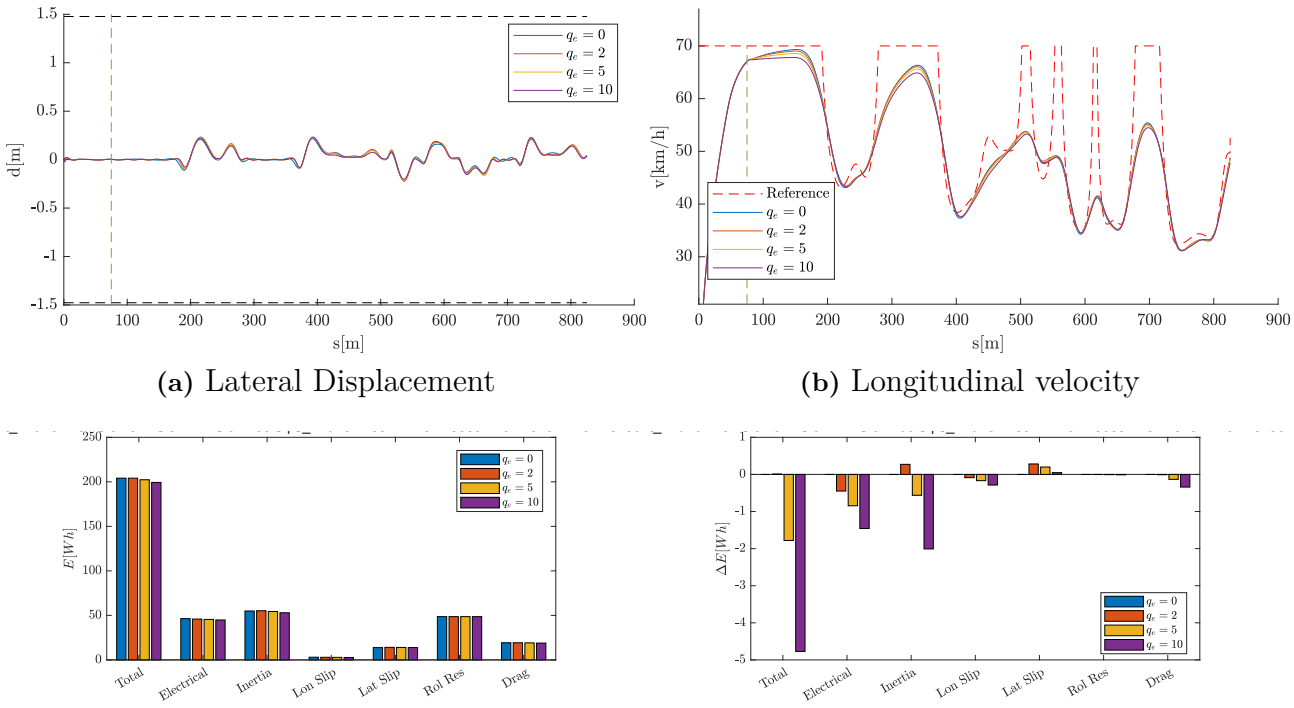
When considering comfort driving, as described in Section 6.2.3.1, longitudinal acceleration is also penalized. As a result, the overall penalty on deviation from reference longitudinal

velocity is lower. Subsequently, the ratio between that penalty and the energy consumption one is decreased, leaving more space for energy consumption reduction. In other words, the controller has now more flexibility to balance the economic cost and the tracking performance penalties in the cost function. For this scenario, the cost function is thus

$$l(\mathbf{x}, \mathbf{u}, \epsilon, s) = (\mathbf{x} - \bar{\mathbf{x}}(s))^T Q(\mathbf{x} - \bar{\mathbf{x}}(s)) + \mathbf{u}^T R\mathbf{u} + Q_e E + q_{\text{ax}}(\frac{a_x}{a_{x,\max}})^2 + \epsilon^T Z\epsilon \tag{6.17}$$

$$m(\mathbf{x}, \epsilon, s) = (\mathbf{x} - \bar{\mathbf{x}}(s))^T Q_f(\mathbf{x} - \bar{\mathbf{x}}(s)) + \epsilon^T Z\epsilon \ , \tag{6.18}$$

with tuning equal to (6.16) and $q_{\text{ax}} = 1$. The results are shown in Fig. 6.25 and summarized in Table 6.4. Conversely to the previous case, increasing energy penalty resulted in energy consumption reduction at an expense of just a slight decrease in average velocity along the path. Such decrease in average is mainly noticeable in corners' entries and exits, whereas within corners the same offset is not visible. This indicates smoother longitudinal control for higher q_e, which is correlated with lower torque actuation, leading in turn to lower inertial power and electrical power losses.



(a) Lateral Displacement

(b) Longitudinal velocity

(c) Total energy consumption

(d) Difference in energy consumption

Figure 6.25: Performance comparison at $\bar{V}_x = 70$ km/h for $q_d = 10$, $q_{\text{ax}} = 1$, and $q_e = \{0, 2, 5, 10\}$.

A natural question that could arise at this point is what happens for larger q_e. From testing, it was found that for $q_e > 10$, the straight-road longitudinal velocity tracking was compromised, due to the fact that the relative cost of tracking the reference velocity was higher than the total energy consumed to do so. This is undesirable for this application given that reference tracking is still the main objective in the optimization problem.

Table 6.4: Key performance indicators at $\bar{V}_x = 70$ km/h for $q_d = 10$, $q_{\text{ax}} = 1$, and $q_e = \{0, 2, 5, 10\}$.

	$q_e = 0$	$q_d = 2$	$q_d = 5$	$q_d = 10$
Lap time [s]	67.2	67.1	67.2	67.5
Lap length [m]	823.0	823.0	823.0	823.0
Total energy [Wh]	204.1	204.1	202.4	199.4
Relative energy saving [%]	-	0.0	0.9	2.4
MAE Δv [km/h]	25.9	25.8	25.9	26.1
MAE d [m]	0.06	0.06	0.06	0.06
Mean computational time [ms]	7.4	14.5	14.7	14.6
Max computational time [ms]	13.8	25.9	24.9	24.0

6.3.2 Energy-optimal trajectory planning

Building upon the results of the previous section, further exploration of path planning possibility was conducted taking comfort driving into account. Not only this optimization setting is more realistic from the product development point of view, but also it allows for further savings in energy. Regarding controller tuning, the cost function remains as in (6.17)-(6.18), with similar tuning apart from q_d that is now let free to evaluate the impact of relaxing the center-of-the-path tracking on energy consumption. Regarding the slack variables, note that they are now smaller than for the path tracking scenario with no energy minimization. The reason why is because near-constraints trajectories are now more common, both regarding acceleration and path limits, and solver convergence is more intricate. Initially, a wide range of q_d was selected and the energy consumption over one lap of the test track was analysed. Together with the MAE of both the tracking of reference longitudinal velocity, $\bar{V}_x$, and center of the path, $d = 0$, the figures below present the energy consumption for the logarithmic range $q_d \in [10^{-4}, 10^2]$ and the range of energy penalties $q_e = \{2, 5, 10\}$ for different $\bar{V}_x$.

Starting with $\bar{V}_x = 30$ km/h, Fig. 6.26 demonstrates the energy saving results whereas the respective tracking performance is illustrated in Fig. 6.27 comparing two extreme cases, $q_d = 10^{-4}$ and $q_d = 10$, for a high penalty on energy $q_e = 10$. The rightmost extreme case of $q_d = 100$ was not selected since it can include noisy actuation due to very high penalty on d. Looking at the energy consumption trend, as q_d decreases, the vehicle deviates more from the center of the path in a corner-cutting fashion, as illustrated in Fig. 6.27a, reducing the travelled distance along the path. Instead, reference longitudinal velocity tracking is not compromised, given the dynamics involved in the maneuver are low. The trend is similar for fixed values of q_e. On the other hand, as q_e increases for constant q_d, the energy consumption decreases due to higher deviation from the reference longitudinal velocity. Fig. 6.27c and 6.27d illustrate the consumption of energy decomposed into the relevant sources of energy loss together with inertia. Overall, the total savings for this scenario are low, being mainly due to the shorter travelled distance. This is visible in the generalized, but low-magnitude reduction in energy losses across all the terms. Table 6.5 sums up the key performance indicators of this example and the energy savings attained.

Results with $\bar{V}_x = 50$ km/h follows, illustrated in Fig. 6.28 and 6.29. Regarding the latter, the analysis is now conducted for $q_e = 2$, since higher values compromise straight-road velocity tracking, i.e. the vehicle travels slower in straights to save energy. Looking at the former,

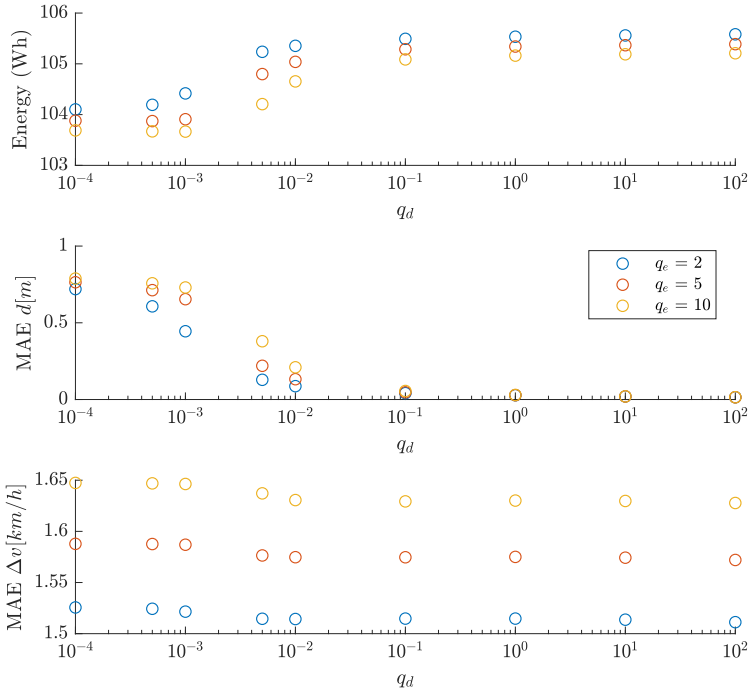
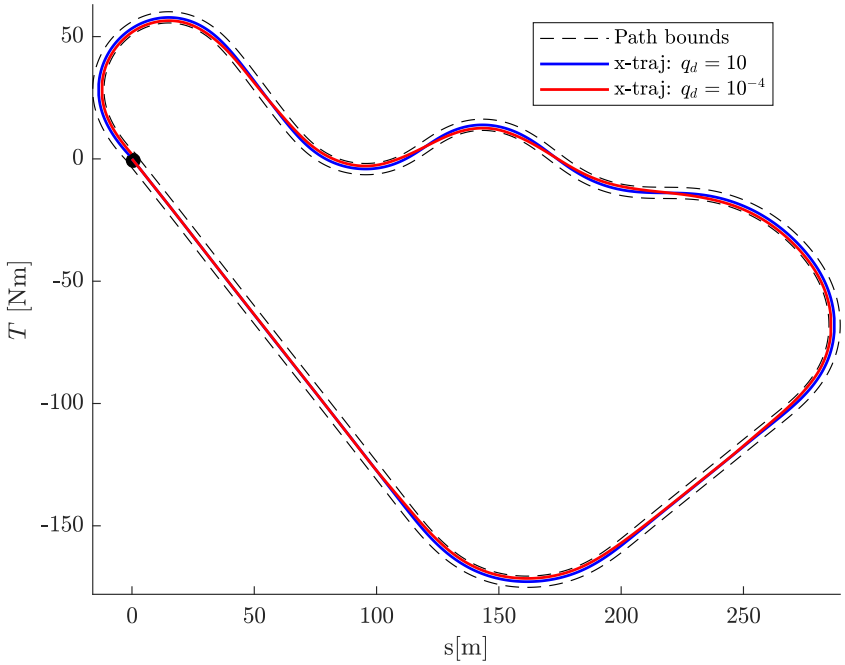


Figure 6.26: Energy consumption at $\bar{V}_x = 30$ km/h for $q_d \in [10^{-4}, 10^2]$ and $q_e = \{2, 5, 10\}$.

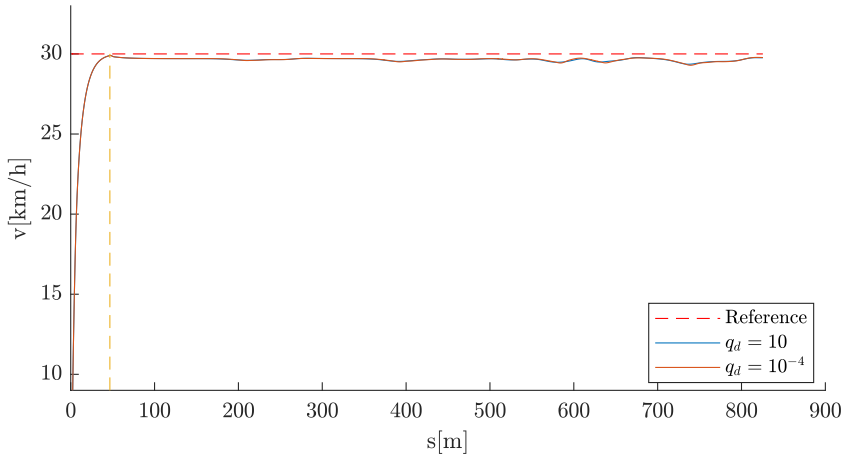
Table 6.5: Key performance indicators at $\bar{V}_x = 30$ km/h for $q_d = \{10^{-4}, 10\}$ and $q_e = 10$.

	$q_d = 10^{-4}$	$q_d = 10$
Lap time [s]	102.9	105.0
Lap length [m]	810.7	823.6
Total energy [Wh]	103.7	105.2
Relative energy saving [%]	1.4	-
MAE Δv [km/h]	1.65	1.63
MAE d [m]	0.02	0.8
Mean computational time [ms]	15.5	17.0
Max computational time [ms]	23.5	31.1

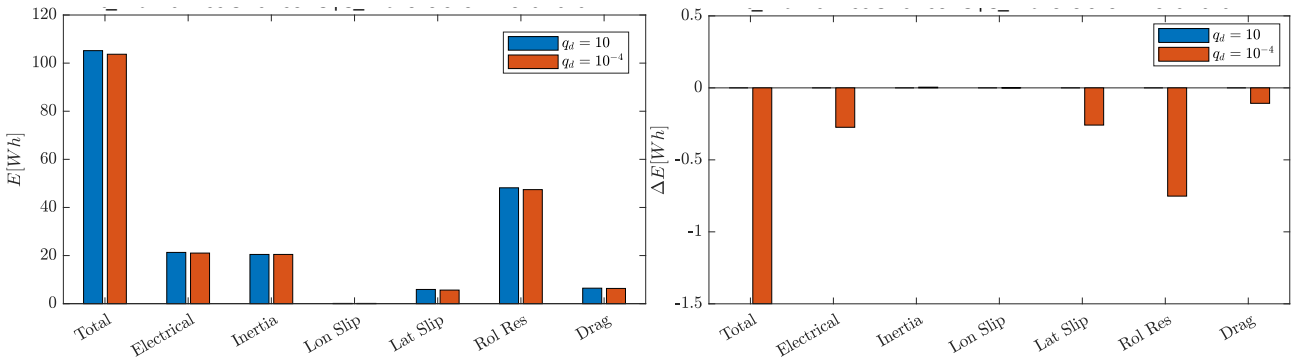
one can observe a double trade-off in reference tracking, where both lateral displacement and longitudinal velocity vary along the range of q_d. Looking at Fig. 6.29a, one can observe that the path profile is now different than at $\bar{V}_x = 30$ km/h, since corners are now cut in an initial stage while the vehicle opens it trajectory when leaving them. Note as well that such behaviour allows for better tracking of $\bar{V}_x$. The findings show little or negligible energy savings can be attained, due to the referred reference tracking trade-off. The potential savings from relaxing the each tracking objective is compromised by its counterpart, thus cancelling each other mutually. Focusing on the energy consumption decomposition, in Fig. 6.29c and 6.29d, where again the same two extreme cases are compared, such trade-off is well understood looking at energy consumption sources. The smoother v_x profile is a synonym of smoother actuation, i.e. smoother acceleration, which yields less electric losses as a result of reduced torque. Conversely, such smoother profile, even if it results in lower longitudinal acceleration, yields higher inertial



(a) Vehicle trajectory



(b) Longitudinal velocity



(c) Total energy consumption

(d) Difference in energy consumption

Figure 6.27: Energy consumption at $\bar{V}_x = 30$ km/h for $q_d = \{10^{-4}, 10\}$ and $q_e = 10$.

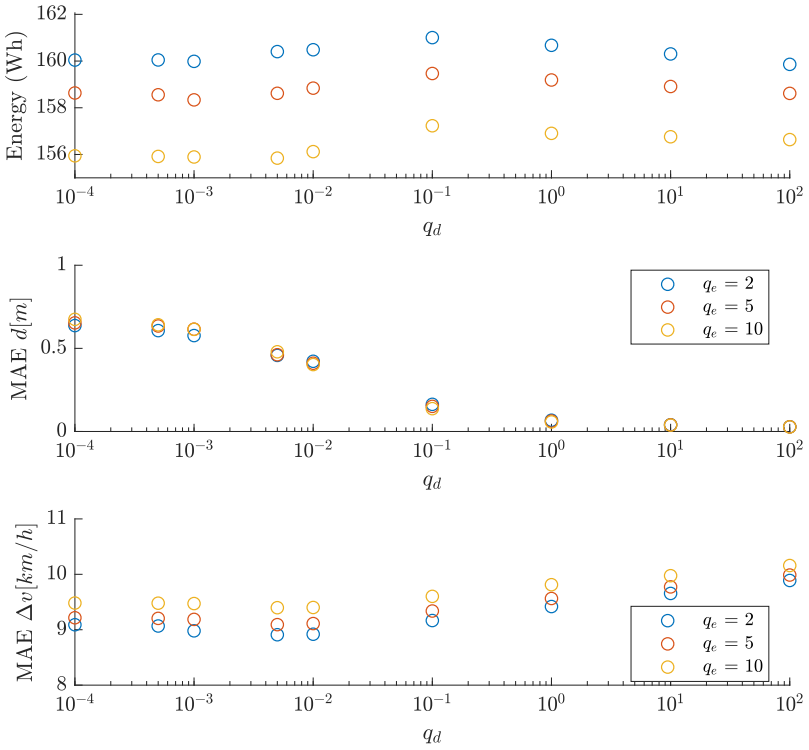


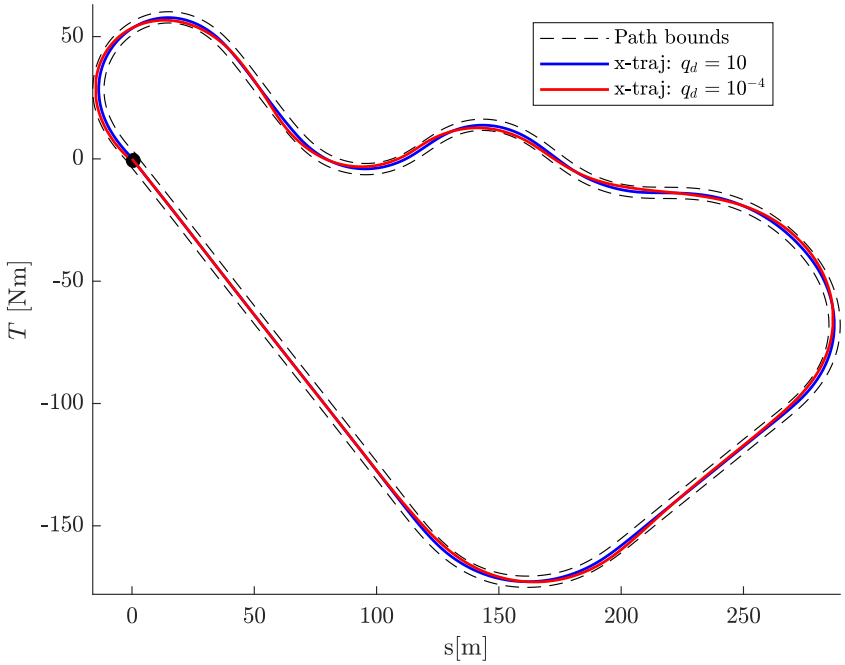
Figure 6.28: Energy consumption at $\bar{V}_x = 50$ km/h for $q_d \in [10^{-4}, 10^2]$ and $q_e = \{2, 5, 10\}$.

energy simply because the longitudinal velocity is higher. Note, additionally, that lateral slip and drag losses also increase as a result of higher longitudinal velocity. To sum-up, Table 6.6 presents the key performance indicators for this second scenario.

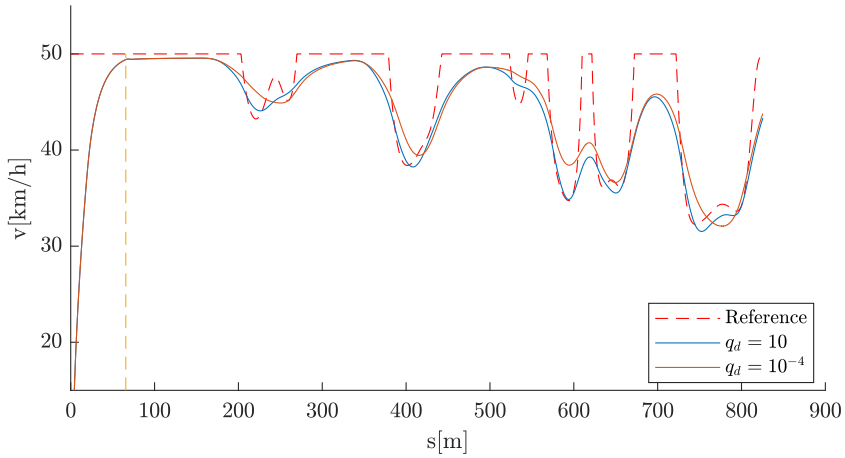
Table 6.6: Key performance indicators at $\bar{V}_x = 50$ km/h for $q_d = \{10^{-4}, 10\}$ and $q_e = 2$.

	$q_d = 10^{-4}$	$q_d = 10$
Lap time [s]	72.3	73.5
Lap length [m]	822.1	823.2
Total energy [Wh]	160.0	160.3
Relative energy saving [%]	0.17	-
MAE Δv [km/h]	9.1	9.7
MAE d [m]	0.64	0.04
Mean computational time [ms]	13.7	14.0
Max computational time [ms]	25.8	38.1

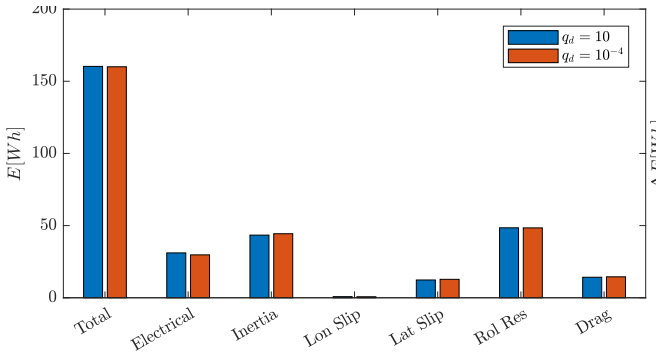
The last scenario involves an even more dynamic maneuver, at $\bar{V}_x = 70$ km/h. Analogously to previous scenarios, the same set of plots is presented, extending the detailed comparison to the cases of $q_d = \{10, 10^2, 10^{-4}\}$, setting $q_e = 2$ for the same reason as for $\bar{V}_x = 50$ km/h. Equivalently to the analysis of the results at that $\bar{V}_x$, there is also a double trade-off present in this results. For high q_d in order to better track the center line of the path the vehicle travels in average at a lower longitudinal velocity, and vice-versa for lower q_d. In what regards the geometry of the trajectory, it is interesting to observe that, given the higher entry velocity of



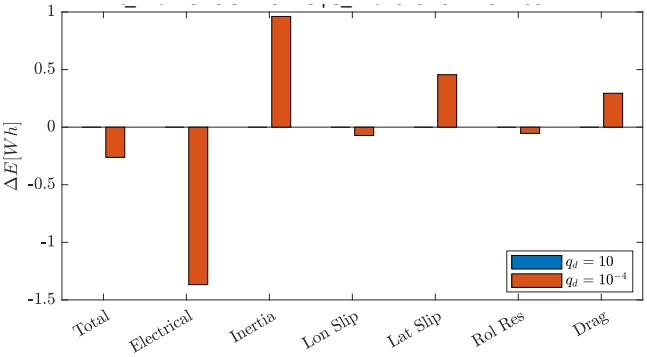
(a) Vehicle trajectory



(b) Longitudinal velocity



(c) Total energy consumption



(d) Difference in energy consumption

Figure 6.29: Performance comparison at $\bar{V}_x = 50$ km/h for $q_d = \{10^{-4}, 10\}$ and $q_e = 2$.

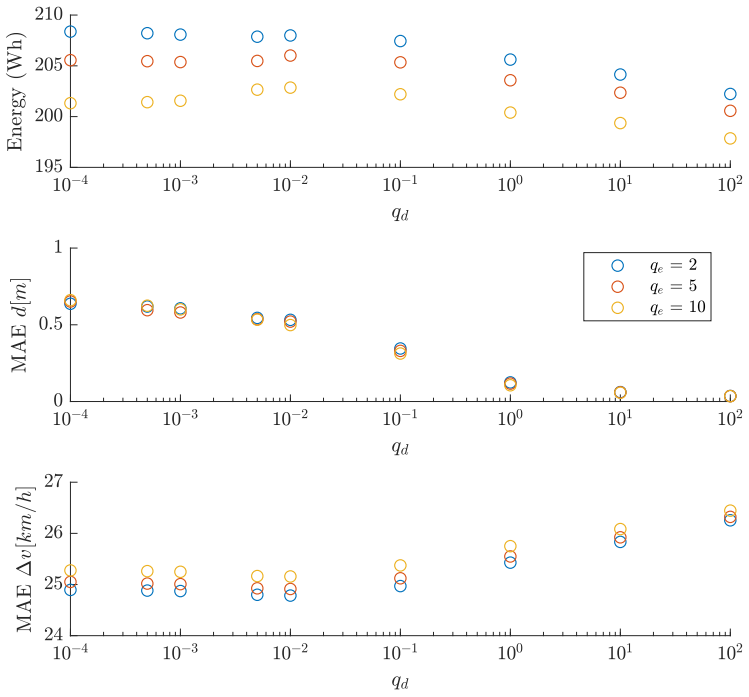
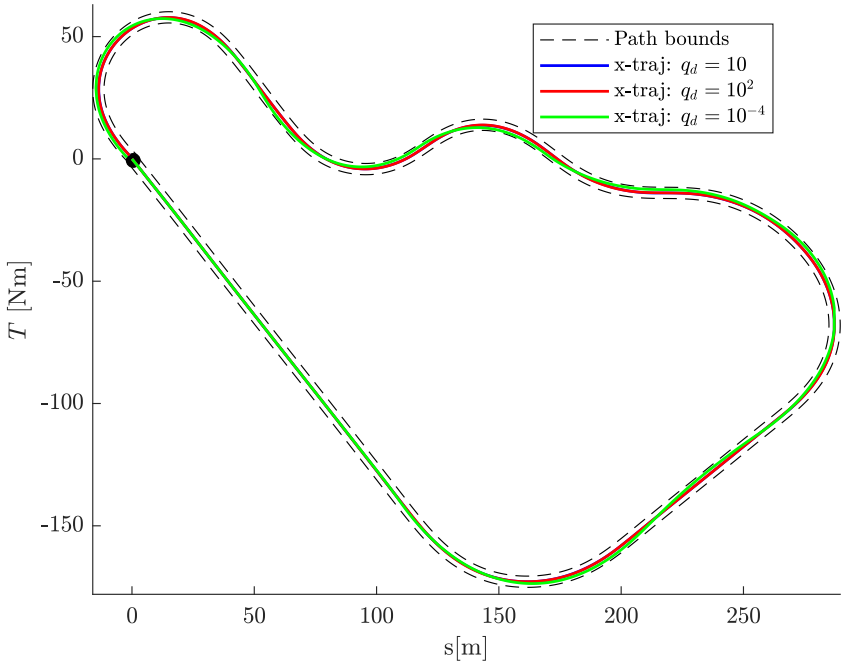


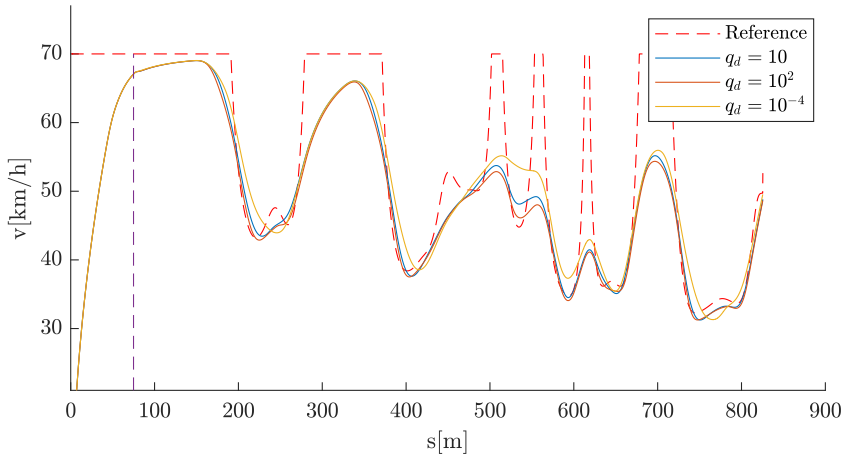
Figure 6.30: Energy consumption at $\bar{V}_x = 70$ km/h for $q_d \in [10^{-4}, 10^2]$ and $q_e = \{2, 5, 10\}$.

the vehicle into corners, the energy-optimal trajectory lies mostly on the outer part of the path. Note that approaching corners by the outer part increases the radius of curvature which in turn decreases the lateral acceleration. Although multiple factors should be taken into consideration, recall the lateral and longitudinal acceleration are kept constrained, and the solver weights the cost of respecting those constraints when it comes to corner cutting. The longitudinal acceleration, specially, is penalized, which also contributes to shaping the trajectories. From the energy perspective, looking at Fig. 6.30, it can be understood that relaxing q_d led to higher energy consumption. When highly penalizing q_d, instead, the longitudinal velocity is lower, specially at corners, saving indeed total energy consumption. This is illustrated both in Fig. 6.31b, where one can observe that $q_d = 10^{-4}$ results, on average, in higher longitudinal velocity, and in Fig. 6.31c and 6.31d, where instead the unique fact that the vehicle travels slower for $q_d = 10^2$ leads to overall reduction in energy in all the terms. The resultant key performance indicators are summarized in Table 6.7, where once again lap time and distance do not differ in great extent.

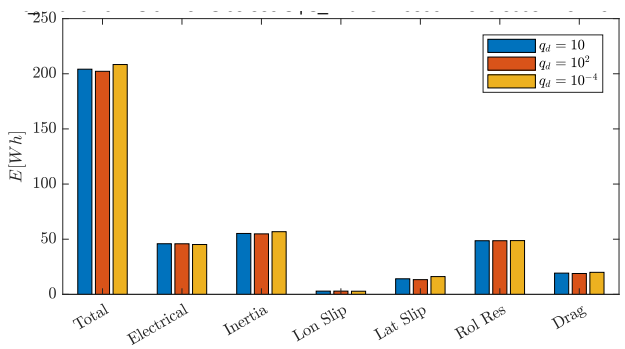
An additional study is conducted at $\bar{V}_x = 70$ km/h where different penalties on energy consumption, $q_e = \{2, 5, 10\}$, are compared for $q_d = 10^{-4}$ to better understand the difference in trajectory profile, which are illustrated for each q_e tuning in Fig. 6.32. Together with Fig. 6.33a and 6.33b, one can conclude on the direction the controller took in order to save energy. Firstly, note that, inline with previous findings, longitudinal control is smoother for higher q_e, resulting in lower entering and exiting corner longitudinal velocity. As a result, the vehicle cornering profile is different. Take the example of the first corner, at around 200m. While for $q_e = \{2, 5\}$ the vehicle opens slightly the corner in anticipation, $q_e = 10$ cuts the corner in the entry zone, at the expense of relaxed longitudinal control. In essence, the solver found that the most energy reduction came from reducing longitudinal velocity, and on top of that, further



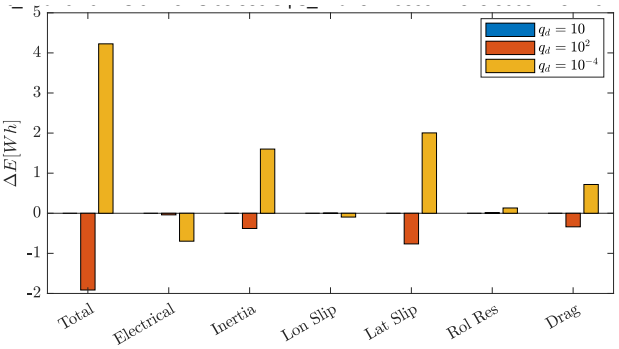
(a) Vehicle trajectory



(b) Longitudinal velocity



(c) Total energy consumption



(d) Difference in energy consumption

Figure 6.31: Performance comparison at $\bar{V}_x = 70$ km/h for $q_d = \{10^{-4}, 10, 10^2\}$

Table 6.7: Key performance indicators at $\bar{V}_x = 70$ km/h for for $q_d = \{10^{-4}, 10, 10^2\}$ and $q_e = 2$.

	$q_d = 10^{-4}$	$q_d = 10$	$q_d = 10^2$
Lap time [s]	65.9	67.1	67.8
Lap length [m]	824.9	822.9	823.3
Total energy [Wh]	208.4	204.1	202.2
Relative energy saving [%]	-2.1	-	0.94
MAE Δv [km/h]	24.9	25.8	26.3
MAE d [m]	0.64	0.06	0.04
Mean computational time [ms]	14.9	14.5	16.3
Max computational time [ms]	30.2	25.9	33.8

reduction on total energy consumption could be attaining by cutting the corner. As expected, $q_e = 10$ resulted in energy savings that are illustrated in Fig. 6.33c and 6.33d and quantified in Table 6.8.

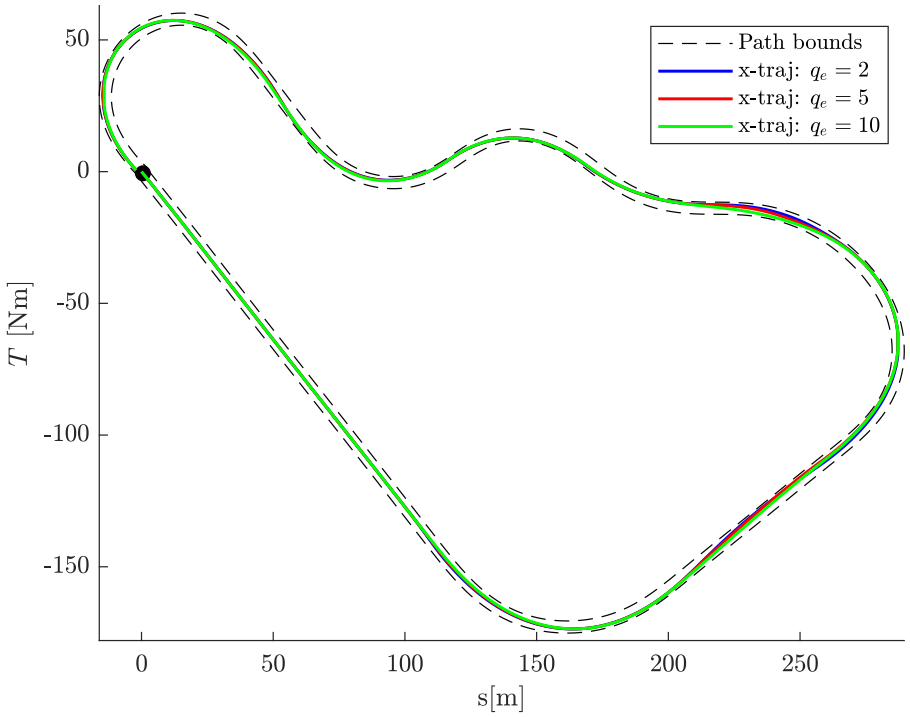
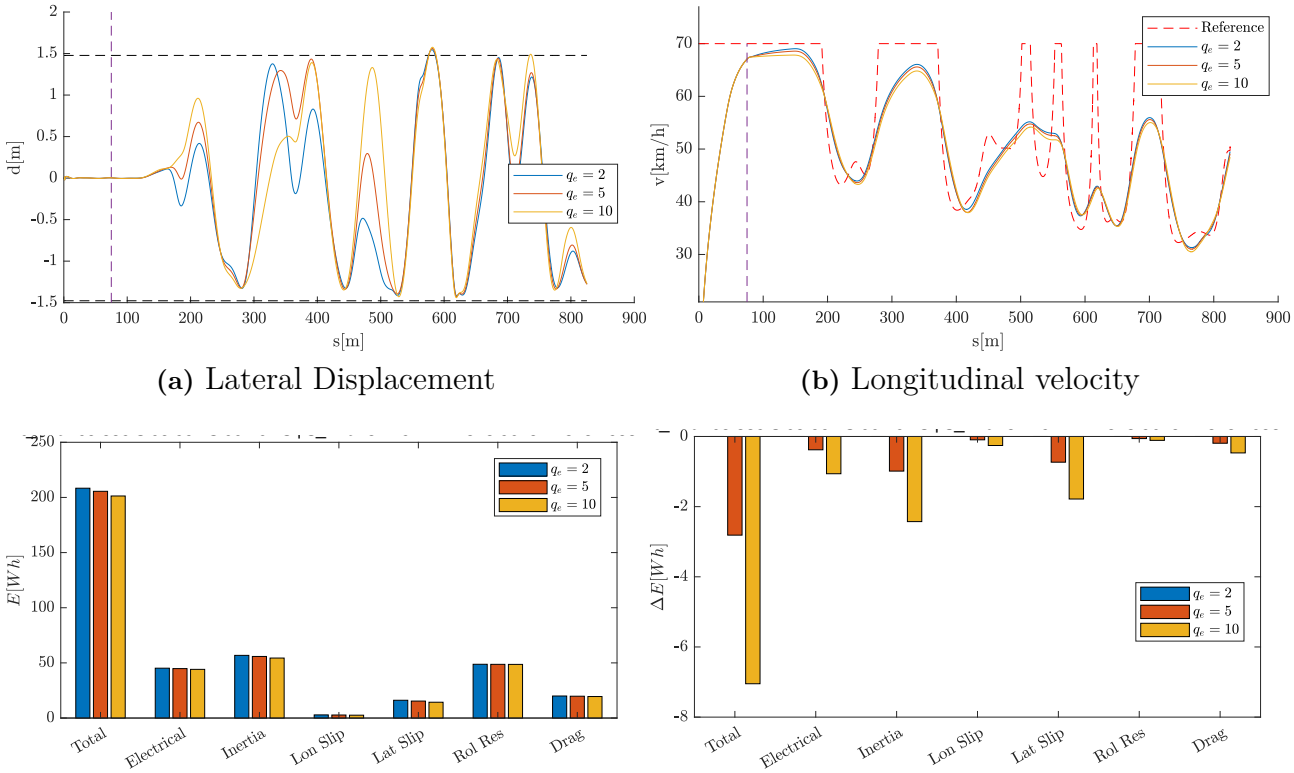


Figure 6.32: Vehicle trajectory at $\bar{V}_x = 70$ km/h for $q_d = 10^{-4}$ and $q_e = \{2, 5, 10\}$.

6.4 Discussion

This section summarizes the findings in Sections 6.2 and 6.3 and presents a general reflection on the implemented method, namely its strengths and weaknesses.

Results from Section 6.2 refer to path tracking applications, where MPC was implemented with different software and respective numerical solvers, and compared with PPC. Both IPM



(a) Lateral Displacement

(b) Longitudinal velocity

(c) Total energy consumption

(d) Difference in energy consumption

Figure 6.33: Performance comparison at $\bar{V}_x = 70$ km/h for $q_d = 10^{-4}$ and $q_e = \{2, 5, 10\}$.

Table 6.8: Key performance indicators at $\bar{V}_x = 70$ km/h for for $q_d = 10^{-4}$ and $q_e = \{2, 5, 10\}$.

	$q_e = 2$	$q_e = 5$	$q_e = 10$
Lap time [s]	65.9	66.0	66.3
Lap length [m]	824.9	824.0	823.2
Total energy [Wh]	208.4	205.6	201.3
Relative energy saving [%]	-	1.4	3.4
MAE Δv [km/h]	24.9	25.1	25.3
MAE d [m]	0.64	0.65	0.66
Mean computational time [ms]	14.9	15.3	14.9
Max computational time [ms]	30.2	27.5	23.2

and SQP performed accurately in terms of reference tracking performance for both smooth and dynamic maneuvering. It is important to highlight that the optimization problem at this point was of reduced complexity, the cost function being quadratic, with box inequality constraints and medium-size horizon and sampling interval. The solvers however differed in computational time, where acados (SQP) implementation was significantly faster than CasADi (IPM). The root justification of such difference is in the backbone theory of each numerical solver and the respective transcription method used. Considering the problem can be considered of medium size and informative initial guesses are given, SQP tends to outperform IPM. On top of that, QPs in SQP iterations are solved with HPIPM, which is a high-performance IPM solver oriented for QP formulated in an OCP setting. These two points combined resulted in acados largely

outperforming IPM in computational time. In turn, that led to faster feedback loop, and smoother trajectories were attained by acados. Further improvements of computational time can be attained by trading optimality for computational speed. Results showed that, for the simpler application of path tracking, RTI scheme outperformed SQP without major difference in performance results. Finally, the study of path tracking results finished by comparing with PPC combined with longitudinal control from IPG CarMaker Driver model, a common baseline in related works. It was shown that MPC was capable to perform similarly to PPC in tracking the center line of the path while centralizing longitudinal and lateral control, with high degree of flexibility for tuning different degrees of smoothness in trajectories. That validates the already known fact in the autonomous driving community that MPC is indeed a powerful method and viable solution for full autonomous vehicle control in the presence of limited, short-sighted information.

Results on energy minimization followed in Section 6.3. As a first step, energy saving potential was studied for path tracking, where the longitudinal control is the main degree of freedom for such objective. Comparing between comfort driving, where reference longitudinal velocity tracking is relaxed, and accurate path tracking, where little deviations from reference are allowed, only the former gave space for energy minimization, as expected. These results highlighted the potential of energy saving having smoother longitudinal control with nearly-negligible loss of reference tracking accuracy. Namely, for a dynamic reference profile with $\bar{V}_x = 70$ km/h, energy reduction around 2.4% was attained with a decrease of solely 0.2 km/h in average velocity. When center of the path tracking is relaxed, different conclusions apply according to the velocity the vehicle is travelling at. While at constant, low longitudinal velocity, gains of around 1.4% in energy efficiency were attained by cutting corners and reducing distance travelled, for dynamic reference longitudinal velocity profiles a trade-off was observed between tracking the different references: deviating from the center of the path allowed for higher longitudinal velocity, and vice-versa. As a result, the possible individual savings from relaxing each of the reference tracking penalties were cancelled mutually, which led to zero energy savings or even higher total consumption.

For all applications, it is of utmost importance to highlight the computational times achieved. They were not only of low value in average, allowing to run the controller as fast as 20 Hz, but also with low standard deviation, where the maximum computational time was consistently around two times the average, respecting in all cases the feedback loop time. Such performance was attained mainly due to relaxation of constraints, accurate initial guesses, and powerful optimization software framework.

The overall conclusion of the energy minimization study is that, for the considered acceleration profile, which limited maneuvers to normal driving applications with $a_{x,\max} = a_{y,\max} = 3$, the main contribution for energy consumption is longitudinal velocity, both in form of inertial power, electric losses, and drag (mainly for high velocities). Lateral control, thus, plays a secondary roll for this application, given that lateral slip losses and lateral inertial power are of low magnitude. Moreover, one can also conclude that MPC involves an intricate tuning trade-off between tracking of references and saving energy. In fact, that is indeed the difficulty associated with economic MPC applications. In this work, it was not yet clearly understood how to formulate EMPC such that deviations from the path could be combined with reduction in longitudinal velocity. To address this and other improvements, the following section describes future directions to take.

6.4.1 Future work

6.4.1.1 Reference longitudinal velocity profile

The reference longitudinal velocity profile used in the presented simulations is initially set as constant and afterwards saturated as a function of the maximum lateral acceleration. For high velocities, this results in very dynamic braking and acceleration requests that can be challenging to the controller. Further work would include changing the reference to track to avoid infeasible requests for longitudinal control.

6.4.1.2 MPC target cost

The target cost in the present work is selected equally to the running cost. Nevertheless, as it is known from MPC theory, target cost can play a crucial role in enhancing stability. Possible emphasis should be put on determining a better target cost to get further stability guarantees, specially if higher accelerations are considered. As a possible solution, the unconstrained MPC problem, i.e. the LQR equivalent, can be solved to obtain the Riccati matrix, which can later be used as a target cost.

Furthermore, the target cost can also be used to emulate higher horizons. From the energy-minimization perspective, since the controller plans the vehicle's trajectory, the longer the horizon the better. Understanding the full profile of the corner is critical for the controller to optimize the entry and exit speed and position of the vehicle. As an example, if the controller understands it has to accelerate again after exiting the corner, it can be optimal to drive faster the corner compromising instead in lateral displacement. Doing so, less torque would be necessary to recover the lost speed when braking and therefore less consumption of energy would be expected. MPC however solves a finite horizon problem in each iteration, with physical limitations on the horizon coming from the perception layer. One way to include further ahead information in the optimization problem would be to select the target cost according to the expected curvature after the corner, using information from offline maps.

6.4.1.3 Higher acceleration limits

The work of this thesis is incomplete if not enlarging the acceleration limits to consider more dynamic driving. To do so, higher complexity could be included in the controller dynamic model to include nonlinear tire model and, possibly, load transfer. This way, accelerations could be unconstrained without having the model mismatch present in this work.

For larger accelerations, namely lateral acceleration, power consumption related to lateral dynamics is expected to gain more relevance. This, in turn, is expected to affect trajectory geometry and the trade-off between lateral and longitudinal control. Further work should then include the study of how does lateral motion impact energy consumption along increasing ranges of lateral acceleration, and how does planned trajectories vary for that same range.

6.4.1.4 Torque Vectoring

In the sequence of the precedent point, it is known from related literature that energy consumption arising from lateral dynamics can be reduced with torque vectoring. Torque vectoring consists of allocating torque differently for each motor of the vehicle. This allows for optimization of such control allocation since the problem becomes redundant: the number

of control inputs is higher than the system's degrees of freedom. A further continuation of this work would therefore be to jointly optimize vehicle trajectory and torque allocation for energy minimization. Namely, it would be of utmost importance to compare centralized and decentralized approaches when it comes to including torque vectoring into the optimization problem, and understand how does trajectories are differently planned. This is expected to increase the complexity of the problem, namely solver convergence time and efficiency.

To overcome that, an efficiency-oriented control architecture could be designed, inspired in the work of [34]. Here, the authors optimized a torque-vectoring induced yaw moment, that is then tracked by low-level controllers. Such architecture has multiple advantages, namely that the number of optimization variables in the MPC problem is reduced, and thus solving complexity is also reduced. In addition, an architecture that separates planning and tracking can ensure that wrong inputs from model mismatch are avoided.

6.4.1.5 Study RTI for energy minimization

As suggested in the path tracking section, the RTI variant of SQP is a promising strategy that, by trading optimality for computational speed, can unlock real-time implementation of complex controllers. As a result of tests throughout this thesis, RTI proved to be less stable in convergence as SQP. A possible improvement could be to start with SQP so that MPC converge to optimal solution in the first set of iterations, and later, when informed initial guesses are available, reduce the number of QP solved, possibly gradually.

6.4.1.6 Vehicle implementation

The logic next step is to validate the current results in the vehicle. The implementation of the MPC scheme considers the use of signals provided by sensors present on the vehicle to operate. Amongst these signals, there is the wheel torque, the wheel steering angle, the position and heading of the vehicle (provided by a GPS, for instance), longitudinal and lateral velocity, and yaw rate. However, one thing to note is that it might be the case that GPS units provide a noisy and inaccurate lateral velocity measurement. In this regard, other means to estimate this, or any other signal that is noisy and/or inaccurate, shall be considered for the proper operation of the MPC. The signals coming from the sensors shall feed the controller at the update rate at which is intended to operate in real-time.

The implementation shall consider the hardware (HW) limitations in terms of computing power, the problem complexity, and its feasibility for real-time applications. In this regard, some implementation tasks with acados have been performed with the MicroAutoBox II but, if for any reason a different HW is used, the implementation shall consider that the procedure to deploy the MPC framework requires that the target HW has an acados supported toolchain file.

In a more practical sense, the implementation shall guarantee that the MPC is always being fed-up with a non-zero speed, due to the singularity implicit in the spatial formulation. Therefore, if starting from standstill, different strategies, such as replacing the true longitudinal velocity measurement by low-magnitude non-zero constant value until a certain threshold is met, shall be considered. Also, the implementation shall consider that other tasks, such as communication tasks of the CAN controller, are being performed by the HW besides the one associated to the MPC. Therefore, task scheduling may be critical to achieve a proper operation of the controller. Depending on the workload associated to other tasks, even parallel computing

could be considered in order to achieve fast update rates, in case the HW allows for it.

6.4.1.7 Sensitive analysis on vehicle control unit

Once implemented in the vehicle control unit, an interesting study can be conducted to decide on number of discretization steps in the horizon, i.e. horizon length N. The computational power available should be taken into consideration when deciding the problem dimensionality and complexity. To fully understand the correlation between horizon length, trajectory optimality, and computational time, the authors suggest a sensitive analysis on the trade-off of these three terms. The idea would consist of increasing gradually N and to measure the average computational time over a test scenario together with the closed-loop cost of the vehicle trajectory. It is expected that a Pareto-type of optimal choice can be made, balancing all the aforementioned factors. If N is too low, computational time can be fast at the cost of loss of optimality, whereas if it increases significantly, optimality is enhanced but the higher dimension of the problem leads to slower computations. Moreover, too low values of N can lead to convergence problems whereas too high values of N is expected to have a limited effect on increasing optimality of trajectories after a certain threshold.

7
Conclusion

This thesis investigates the joint problem of energy optimal longitudinal and lateral control of an electric vehicle, given a predetermined reference path and longitudinal velocity profile. To conduct high-fidelity testing, vehicle simulations are performed using IPG CarMaker. The approach adopted in this research involves the implementation of Model Predictive Control (MPC) for both path tracking and energy-optimal trajectory planning, where the latter can be regarded as an economic MPC problem. Consistent with related literature, the control problem is formulated in the spatial domain, and the path information is translated into curvature coordinates.

The controller is first tested for path tracking applications without energy minimization where the base performance of MPC is assessed. Both CasADi and acados software packages are implemented in order to compare the performance of the respective numerical solvers: Interior Point Method (IPM) and Sequential Quadratic Programming (SQP). Moreover, MPC implementation is compared with Pure Pursuit Control, and MPC comfort-oriented driving is described. For this first section, results indicate that acados implementation outperforms the CasADi one, both in terms of computational time and accuracy. The reduced computational times of SQP suggest this implementation is suitable for real-time implementation. Furthermore, MPC performed equivalent performance to Pure Pursuit Control for path tracking, with the added advantage of centralizing longitudinal and lateral control.

Subsequently, the energy term is incorporated into the cost function, and the potential for energy savings is explored by relaxing the emphasis on reference tracking. The results demonstrate that the greatest energy saving potential for normal driving, i.e. for constrained acceleration, lies on optimizing longitudinal control. Particularly, when tracking the center of the path given a dynamic reference longitudinal velocity profile, results show great potential for energy consumption reduction by smoothing longitudinal control, with over relevant energy savings as a result of little reduction on average longitudinal velocity. However, the results also highlight that the tuning of the economic MPC involves a cumbersome trade-off between the economic cost, the energy, and the reference tracking cost. In certain cases, this led to sub-optimal solutions where near-zero savings or even an increase in energy consumption when the vehicle deviated from the center of the path. Despite the inclusion of the energy term increasing the computational time per iteration of MPC, the computational times remained low comparing with standards and once again suggest that is possible to implement the controller into a real vehicle.

To conclude, the proposed controller has proved to be efficient and robust for energy minimization trajectory planning for electric vehicles, with the developed MPC framework providing a solid foundation for future advancements in this field.

Bibliography

[1] Joel A E Andersson and Moritz Diehl. *User Documentation for CasADi v2.2.0*. Online. 2015. URL: https://casadi.sourceforge.net/v2.2.0/users_guide/html/.

[2] Joel A E Andersson et al. "CasADi – A software framework for nonlinear optimization and optimal control". In: *Mathematical Programming Computation* (2018).

[3] Peter Bearman. "Bluff Body Flow Research with Application to Road Vehicles". In: *The Aerodynamics of Heavy Vehicles II: Trucks, Buses, and Trains*. Ed. by Fred Browand, Rose McCallen, and James Ross. Springer Berlin Heidelberg, 2009.

[4] Karl Berntorp et al. "Models and methodology for optimal trajectory generation in safety-critical road–vehicle manoeuvres". In: *Vehicle System Dynamics* 52.10 (2014), pp. 1304–1332. DOI: 10.1080/00423114.2014.939094.

[5] Dimitri Bertsekas. *Dynamic programming and optimal control: Volume I*. Vol. 1. Athena scientific, 2012.

[6] Lorenz T Biegler. "Solution of dynamic optimization problems by successive quadratic programming and orthogonal collocation". In: *Computers & chemical engineering* 8.3-4 (1984), pp. 243–247.

[7] Hans Georg Bock and Karl-Josef Plitt. "A multiple shooting algorithm for direct solution of optimal control problems". In: *IFAC Proceedings Volumes* 17.2 (1984), pp. 1603–1608.

[8] Stefano Di Cairano. "An Industry Perspective on MPC in Large Volumes Applications: Potential Benefits and Open Challenges". In: *IFAC Proceedings Volumes* 45.17 (2012). 4th IFAC Conference on Nonlinear Model Predictive Control, pp. 52–59. ISSN: 1474-6670. DOI: https://doi.org/10.3182/20120823-5-NL-3013.00040.

[9] Daniele Casanova. "On minimum time vehicle manoeuvring: The theoretical optimal lap". In: (2000).

[10] C Chatzikomis et al. "An energy-efficient torque-vectoring algorithm for electric vehicles with multiple motors". In: *Mechanical systems and signal processing* 128 (2019), pp. 655–673.

[11] R Craig Coulter. *Implementation of the pure pursuit path tracking algorithm*. Tech. rep. Carnegie-Mellon UNIV Pittsburgh PA Robotics INST, 1992.

[12] Giovanni De Filippis et al. "Energy-efficient torque-vectoring control of electric vehicles with multiple drivetrains". In: *IEEE Transactions on Vehicular Technology* 67.6 (2018), pp. 4702–4715.

[13] Moritz Diehl, Hans Joachim Ferreau, and Niels Haverbeke. "Efficient numerical methods for nonlinear MPC and moving horizon estimation". In: *Nonlinear model predictive control: towards new challenging applications* (2009), pp. 391–417.

[14] Moritz Diehl et al. "Fast direct multiple shooting algorithms for optimal robot control". In: *Fast motions in biomechanics and robotics: optimization and feedback control* (2006), pp. 65–93.

[15] Yuanli Ding et al. “Automotive Li-ion batteries: current status and future perspectives”. In: *Electrochemical Energy Reviews* 2 (2019), pp. 1–28.

[16] Isak Erval and Ziad George. “Centralised yaw and lateral motioncontrol for future electric vehicles”. Chalmers University of Technology, 2022.

[17] Paolo Falcone et al. “A linear time varying model predictive control approach to the integrated vehicle dynamics control problem in autonomous systems”. In: *2007 46th IEEE Conference on Decision and Control*. IEEE. 2007, pp. 2980–2985.

[18] Paolo Falcone et al. “Predictive active steering control for autonomous vehicle systems”. In: *IEEE Transactions on control systems technology* 15.3 (2007), pp. 566–580.

[19] Hans Joachim Ferreau et al. “qpOASES: A parametric active-set algorithm for quadratic programming”. In: *Mathematical Programming Computation* 6 (2014), pp. 327–363.

[20] Janick V Frasch et al. “An auto-generated nonlinear MPC algorithm for real-time obstacle avoidance of ground vehicles”. In: *2013 European Control Conference (ECC)*. IEEE. 2013, pp. 4136–4141.

[21] Gianluca Frison and Moritz Diehl. “HPIPM: a high-performance quadratic programming framework for model predictive control”. In: *IFAC-PapersOnLine* 53.2 (2020), pp. 6563–6569.

[22] Yiqi Gao et al. “Spatial predictive control for agile semi-autonomous ground vehicles”. In: *Proceedings of the 11th international symposium on advanced vehicle control*. 2. 2012, pp. 1–6.

[23] Shengbo Li et al. “Model Predictive Multi-Objective Vehicular Adaptive Cruise Control”. In: *IEEE Trans. Contr. Sys. Techn.* 19 (May 2011), pp. 556–566. DOI: 10.1109/TCST.2010.2049203.

[24] Gijs van Lookeren Campagne. “A Nonlinear Model Predictive Control based Evasive Manoeuvre Assist Function”. TU Delft Mechanical, Maritime and Materials Engineering, 2019.

[25] Suvarna M. and Prabhu A. “Numerical optimization methods for improving energy efficiency in Battery Electric Vehicles using wheel torques and steering angles”. Chalmers University of Technology, 2021.

[26] Hans Pacejka. *Tire and vehicle dynamics*. Elsevier, 2012.

[27] Alberto Parra et al. “On Nonlinear Model Predictive Control for Energy-Efficient Torque-Vectoring”. In: *IEEE Transactions on Vehicular Technology* 70.1 (2021), pp. 173–188. DOI: 10.1109/TVT.2020.3022022.

[28] Lev Semenovich Pontryagin. *Mathematical theory of optimal processes*. CRC press, 1987.

[29] James Blake Rawlings, David Q Mayne, and Moritz Diehl. *Model predictive control: theory, computation, and design*. Vol. 2. Nob Hill Publishing Madison, WI, 2017.

[30] Hannah Ritchie. *Cars, planes, trains: Where do CO2 emissions from transport come from?* Accessed: May 9, 2023. Oct. 2020. URL: https://ourworldindata.org/co2-emissions-from-transport.

[31] RWH Sargent and GR Sullivan. “The development of an efficient optimal control package”. In: *Optimization Techniques: Proceedings of the 8th IFIP Conference on Optimization Techniques Würzburg, September 5–9, 1977*. Springer. 1978, pp. 158–168.

[32] Efstathios Siampis, Efstathios Velenis, and Stefano Longo. “Rear wheel torque vectoring model predictive control with velocity regulation for electric vehicles”. In: *Vehicle System Dynamics* 53.11 (2015), pp. 1555–1579.

[33] Efstathios Siampis et al. “A real-time nonlinear model predictive control strategy for stabilization of an electric vehicle at the limits of handling”. In: *IEEE Transactions on Control Systems Technology* 26.6 (2017), pp. 1982–1994.

[34] Sirish Srinivasan, Sebastian Nicolas Giles, and Alexander Liniger. “A holistic motion planning and control solution to challenge a professional racecar driver”. In: *IEEE Robotics and Automation Letters* 6.4 (2021), pp. 7854–7860.

[35] M Švec et al. “Predictive Torque Vectoring Vehicle Control Based on a Linear Time Varying Model”. In: *2019 42nd International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO)*. IEEE. 2019, pp. 960–965.

[36] Jacob Svendenius. “Tire Models for Use in Braking Applications”. PhD thesis. Department of Automatic Control, 2003.

[37] Juliette Torinsson et al. “Energy reduction by power loss minimisation through wheel torque allocation in electric vehicles: a simulation-based approach”. In: *Vehicle System Dynamics* 60.5 (2022), pp. 1488–1511.

[38] Mahyar Vajedi and Nasser L. Azad. “Ecological Adaptive Cruise Controller for Plug-In Hybrid Electric Vehicles Using Nonlinear Model Predictive Control”. In: *IEEE Transactions on Intelligent Transportation Systems* 17.1 (2016), pp. 113–122. DOI: 10.1109/TITS.2015.2462843.

[39] Robin Verschueren et al. “acados—a modular open-source framework for fast embedded optimal control”. In: *Mathematical Programming Computation* 14.1 (2022), pp. 147–183.

[40] *Volvo Cars to be fully electric by 2030*. Accessed: May 9, 2023. Mar. 2021. URL: https://www.media.volvocars.com/global/en-gb/media/pressreleases/277409/volvo-cars-to-be-fully-electric-by-2030.

[41] Andreas Wächter and Lorenz T Biegler. “On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming”. In: *Mathematical programming* 106 (2006), pp. 25–57.

[42] Jo Yung Wong. *Theory of ground vehicles*. John Wiley & Sons, 2022.

[43] Stephen J Wright. *Primal-dual interior-point methods*. SIAM, 1997.

[44] Jiapeng Wu and Lihe Chen. “Numerical Strategies for Energy Optimization in Battery Electric Vehicles”. Chalmers University of Technology, 2022.

DEPARTMENT OF SOME SUBJECT OR TECHNOLOGY
CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden
www.chalmers.se

CHALMERS
UNIVERSITY OF TECHNOLOGY