Treball de Fi de Màster

Master's Degree in Automatic Control and Robotics (MUAR)

Safe Coordination of Autonomous Vehicles

MEMÒRIA

September 16, 2022

Autor: José Requena Gallego

Directors: Dr. Vicenç Puig Cayuela

Convocatòria: Setembre, 2022

Escola Tècnica Superior
d'Enginyeria Industrial de Barcelona

Abstract

As more and more autonomous vehicles enter our road, new mechanisms must be considered to ensure the safe coordination between the autonomous vehicles. Although many algorithms have been proposed to coordinate autonomous vehicles, few of them have considered the robustness of the solution against disturbances.

Therefore, in this master's thesis, a vehicle coordination algorithm that uses vehicle to vehicle (V2V) communication is design in order to achieve collision free trajectories, while rejecting disturbances. Specifically, a robust tube-based model predictive control (MPC) scheme is proposed in order to control the autonomous vehicle. This controller uses series of zonotopic reachable sets (also known as tube) to compute a set of state and input constraints, which ensure the robust feasibility of the problem. To reduce the computational burden of the MPC optimization problem, the vehicle model is reformulated into a pseudo-linear model by transforming its non-linear equations into the linear parameter varying (LPV) form. The disturbance rejection is performed by a H_∞-optimal corrective controller. Finally, the collision avoidance is achieved by a V2V coordination algorithm, in which the lateral bounds of a collision free path are computed.

To validate the proposed control scheme, a series of simulations have been performed to test the disturbance rejection of the corrective controller, as well as the vehicle coordination capabilities. The results from these tests show that the proposed controller is effective in coordinating a multiple overtaking maneuver, while rejecting the disturbances.

Key words: Autonomous vehicles, vehicle coordination, tube-based MPC, LPV modelling, H_∞-optimal control.

To all the friends
that I have made
in Barcelona.

José Requena Gallego

Acknowledgements

I would like to express my deep and sincere gratitude to my master's thesis supervisor Dr. Vicenç Puig Cayuela, for his support, patience and suggestions throughout the development of this work.

I would also like to thank my family and friends, who have supported me to complete this master's thesis directly or indirectly.

Table of Contents

ETSEIB

List of Figures

List of Tables

List of Abbreviations

V2V	Vehicle to Vehicle
COM	Control-Oriented Model
SOM	Simulation-Oriented Model
ODE	Ordinary Differential Equations
CG	Centre of Gravity
LTI	Linear Time Invariant
LPV	Linear Parameter Varying
SS	State Space
CT	Continuous Time
DT	Discrete Time
LMI	Linear Matrix Inequality
PCA	Principal Component Analysis
LQR	Linear Quadratic Regulator
SPOF	Single Point of Failure
MPC	Model Predictive Control
T-MPC	Tube-based Model Predictive Control

List of Symbols

Kinematic Models

X_G	Global X coordinate of the vehicle
Y_G	Global Y coordinate of the vehicle
θ_G	Global θ coordinate of the vehicle
v_x	Longitudinal linear velocity of the vehicle
v_x	Lateral linear velocity of the vehicle
ω	Angular velocity of the vehicle
s	Distance travelled along a curvilinear trajectory
y_e	Lateral error of the vehicle
θ_e	Orientation error of the vehicle
v_{x_c}	Velocity parallel to the trajectory's slope
v_{y_c}	Velocity perpendicular to the trajectory's slope
ω_c	Desired angular velocity
R	Radius of the trajectory at a distance s
κ	Curvature of the trajectory

Dynamic Models

F_{y_f}	Front lateral tire force
F_{y_r}	Rear lateral tire force
F_{motor}	Force generated by the vehicle's motor or breaks
F_{res_x}	Longitudinal resistive force
F_{res_y}	Lateral resistive force
F_f	Rolling resistive force due to tire friction
F_{d_x}	Longitudinal aerodynamic drag force
F_{d_y}	Lateral aerodynamic drag force
F_g	Gravitational force

a_x	Inertial acceleration in the longitudinal direction
a_y	Inertial acceleration in the lateral direction
δ_i	Steering angle of the inner wheel of the Ackermann model
δ_o	Steering angle of the outer wheel of the Ackermann model
δ	Steering angle of the bicycle model
a	Rear wheel's acceleration
α_f	Front tire slip angle
α_r	Rear tire slip angle
v_f	Front tire's velocity vector
v_r	Rear tire's velocity vector
θ_{v_f}	Front tire's velocity orientation w.r.t. the longitudinal axis
θ_{v_r}	Rear tire's velocity orientation w.r.t. the longitudinal axis
W	Vehicle's width
L	Vehicle's length
l_f	Distances from the Centre of Gravity (CG) to the front tire
l_r	Distances from the CG to the rear tire
C_f	Front tire cornering stiffness
C_r	Rear tire cornering stiffness
B	Stiffness factor of Pacejka's Magic Formula
C	Shape factor of Pacejka's Magic Formula
D	Peak value of Pacejka's Magic Formula
E	Curvature factor of Pacejka's Magic Formula
m	Vehicle's mass
μ	Static friction coefficient
g	Gravitational constant
v_{w_x}	Wind velocity in the longitudinal direction
v_{w_y}	Wind velocity in the lateral direction
$C_{d_{long}}$	Aerodynamic drag coefficient in the longitudinal direction
$C_{d_{lat}}$	Aerodynamic drag coefficient in the lateral direction
ρ	Mass density of air
$A_{r_{long}}$	Frontal area of the vehicle
$A_{r_{lat}}$	Lateral area of the vehicle
ϕ_{road}	Angle of inclination of the road

I_z	Vehicle's moment of inertia
l_w	Longitudinal distance from the CG to the point where wind is exerting its force

LPV Models

x	States of the LPV model
u	Control actions of the LPV model
ζ	LPV scheduling variables
θ	Varying parameter vector of the LPV model
Θ	Convex polytope bounding the varying parameter vector
$\hat{\Theta}$	Reduced convex polytope bounding the varying parameter vector
$C(\alpha)$	Tire cornering stiffness approximate function
ϵ	Small constant used to avoid singularities
$\text{co}\{\cdot\}$	Convex hull operator
$\mu_i(\theta)$	Vertex membership function
Υ	Trajectory of scheduling variables
Ξ	Trajectory of varying parameters
Ξ^{norm}	Normalized trajectory of varying parameters
ψ	Reduced varying parameters vector of the LPV model
Ψ	Convex polytope bounding reduced varying parameters vector

H_∞-LPV Corrective Controller

z	Performance output vector
w	Exogenous input vector
d	Disturbance vector
P	Generalized plant
K	Corrective feedback controller gain

Vehicle Coordination

s^{vect}	Performance output vector
y_e^{vect}	Exogenous input vector
W_{road}	Width of the road

MPC Formulation

E	Zonotopic reachable set of the mismatch error system
X	Zonotope bounding the states
$\tilde{X}$	Tightened zonotope bounding the states
U	Zonotope bounding the inputs
$\tilde{U}$	Tightened zonotope bounding the inputs
W	Zonotope bounding the disturbances
Q	Tuning matrix for the reference error
R	Tuning matrix for the variation of the control inputs

Part I

Introduction

Chapter 1

Introduction and Overview

1.1 Motivation

Conventional means of transportation face a series of issues, such as traffic accidents and air pollution. According to the World Health Organization, approximately 1.35 million people die each year worldwide as a result of road traffic crashes, and between 20 and 50 million more suffer non-fatal injuries, with many incurring a disability as a result of their injury [1]. Of these accidents, at least 97.5% of them were attributed to human factor causes [2]. Similarly, it is estimated that 4.2 million people die worldwide due to ambient air pollution [3], of which 27% can be attributed to transportation [4]. Other problems present in transportation include traffic congestion, which in the USA can entail an annual loss of 74.5 billion dollars [5].

Figure 1.1: Autonomous vehicles coordinating in an intersection. Source [6]

The extended use of autonomous vehicles is not only expected to reduce most of the car accidents attributed to human factors [7], but also decrease traffic congestion in cities, which will reduce air pollution [8]. Some studies have estimated that autonomous vehicle will account for up to 87.2% of vehicles by the year 2045 [9].

However, as more and more autonomous vehicles enter our road, new V2V communication protocols must be considered to ensure the safe coordination between the autonomous vehicles, especially in the execution of mobility tasks, e.g. platooning and ramp merging, or in complex scenarios with shared resources, e.g. parking slots and intersections [10], as presented in Figure 1.1.

This is the context in which the project developed in this thesis is framed. Specifically, a vehicle coordination algorithm that uses V2V communication is design in order to achieve collision free trajectories.

1.2 Objective

The main objective of this thesis is to design a real-time autonomous vehicle coordination algorithm using a MPC approach, capable of eluding collisions while rejecting disturbances. Based on this key objective, the following short-term goals have been defined:

- Study of the vehicle dynamics, as well as its mathematical model.
- Derive a LPV model of the vehicle.
- Design of a corrective controller capable of rejecting the disturbances affecting the vehicle.
- Propose an autonomous vehicle coordination algorithm that avoids collisions.
- Design a robust tube-based MPC controller.
- Reduce the computational burden of the MPC optimization problem by considering the LPV model of the vehicle.
- Perform simulation tests to validate the proposed control scheme.

1.3 Document Structure

The document is divided into the following chapters:

- **Chapter 2: Vehicle Modelling**. This chapter presents the kinematic and dynamic models of the vehicle.
- **Chapter 3: LPV Representation of the Vehicle Models**. In this chapter, the LPV representation of the vehicle models are derived.
- **Chapter 4: Polytopic LPV Representation of the Vehicle Models**. In this chapter, the polytopic LPV representation of the dynamic vehicle model is

computed using an approximate method of lower order.

- **Chapter 5: Robust LPV Corrective Controller**. This chapter presents the design of a robust LPV corrective controller capable of rejecting the disturbances affecting the system.
- **Chapter 6: Autonomous Vehicle Coordination**. This chapter presents an autonomous vehicle coordination algorithm by which collisions are avoided.
- **Chapter 7: Robust Tube-Based MPC Controller for Autonomous Driving**. In this chapter, a robust tube-based MPC controller is derived to achieve robust feasibility.
- **Chapter 8: Tests and Results**. This chapter presents a series of tests performed to validate the proposed control scheme, as well as their results.
- **Chapter 9: Social, Economic and Environmental Impact**. This chapter presents a sustainability study of the project, with special emphasis on its social, economic and environmental impact.
- **Chapter 10: Project Budget**. This chapter presents the budget associated to the project, considering the material costs, personnel expenses and the amortization of the used equipment.
- **Chapter 11: Conclusions and Future Work**. This chapter presents the conclusions derived from the development of the project, as well as future work proposals.

Part II

Development

Chapter 2

Vehicle Modelling

The dynamics of a road vehicle, such as a car, can be modelled by a set of first order differential equations that describe its mechanical properties in a mathematical way.

These differential equations can be grouped into two major groups: kinematic and dynamic models. The first one describes the motion of the vehicle, such as the position, velocity or acceleration, without taking into account the forces that affected this motion, whilst the second one describes this motion by considering these forces as the cause of movement.

Deciding on a kinematic or dynamic model depends on the chosen vehicle and the control application. In vehicles operating at low velocities, Kinematic models have been used for control purposes, e.g. [11], as the lateral forces affecting the vehicle are negligible. Nevertheless, when these forces are not negligible, such as in high velocity vehicles, accurate dynamic models are necessary, e.g. [12].

The following sections describe the kinematic and dynamic models that are used throughout the project, both for simulation and controlling objectives.

2.1 Kinematic Model

Kinematic models present the advantage of being simpler and easier to derive than dynamic models, as they only depend on the geometrical relationship between the vehicle and its surroundings.

There are many kinematics models available to choose from, each of them corresponding to different types of vehicle geometries (mass-point, differential drive, bicycle, Ackermann, etc.) or the trajectory's description (curvature-based, reference-based, etc.) [13], [14] & [15].

The following subsection presents the mass-point kinematic model, which is widely used for its simplicity.

2.1.1 Mass-Point Kinematic Model

The mass-point kinematic model uses the linear and angular velocities of a vehicle expressed in local coordinates, to compute the motion of the CG of the vehicle expressed in global coordinates.

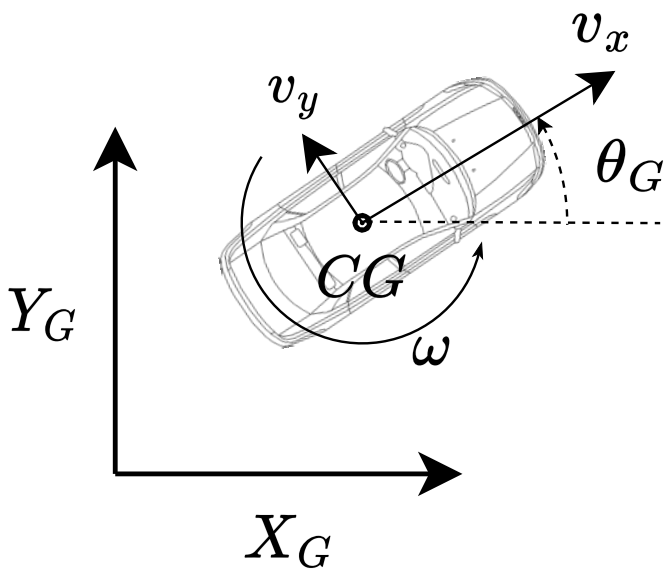


Figure 2.1: Diagram of the mass-point kinematic model.

To better illustrate this model, the diagram presented in Figure 2.1 shows the local and global coordinate system of the vehicle, as well as the positive directions of the linear and angular velocities of said vehicle.

From this diagram, the differential equations describing the motion of the CG of the vehicle can be obtained by applying trigonometric relationships, as shown in (2.1).

$$\begin{cases} \dot{X}_G = v_x \cos\theta_G - v_y \sin\theta_G & \text{(2.1a)} \\ \dot{Y}_G = v_x \sin\theta_G + v_y \cos\theta_G & \text{(2.1b)} \\ \dot{\theta}_G = \omega & \text{(2.1c)} \end{cases}$$

Where X_G, Y_G and θ_G express the global position and orientation of the CG of the vehicle, and v_x, v_x and ω represent respectively, the longitudinal and the lateral linear velocity and the angular velocity of the vehicle.

This model is useful for simulation purposes as the global position and orientation of the CG of the vehicle can be readily computed from the velocities obtained from the dynamic bicycle model presented in Section 2.2.

However, from a control point of view focused on vehicle-coordination, the motion described by (2.1) does not offer any practical information, as constraining the X_G, Y_G coordinates throughout an arbitrary trajectory would prove highly complex and computationally costly. For this reason, the next kinematic model is proposed.

2.1.2 Curvature-Based Kinematic Model

The necessity to express the vehicle's road constraints in an efficient manner justifies the use of the curvature-based kinematic model. This model uses the linear and angular velocities of a vehicle expressed in local coordinates to compute the motion of the CG of the vehicle with respect to a curvilinear trajectory.

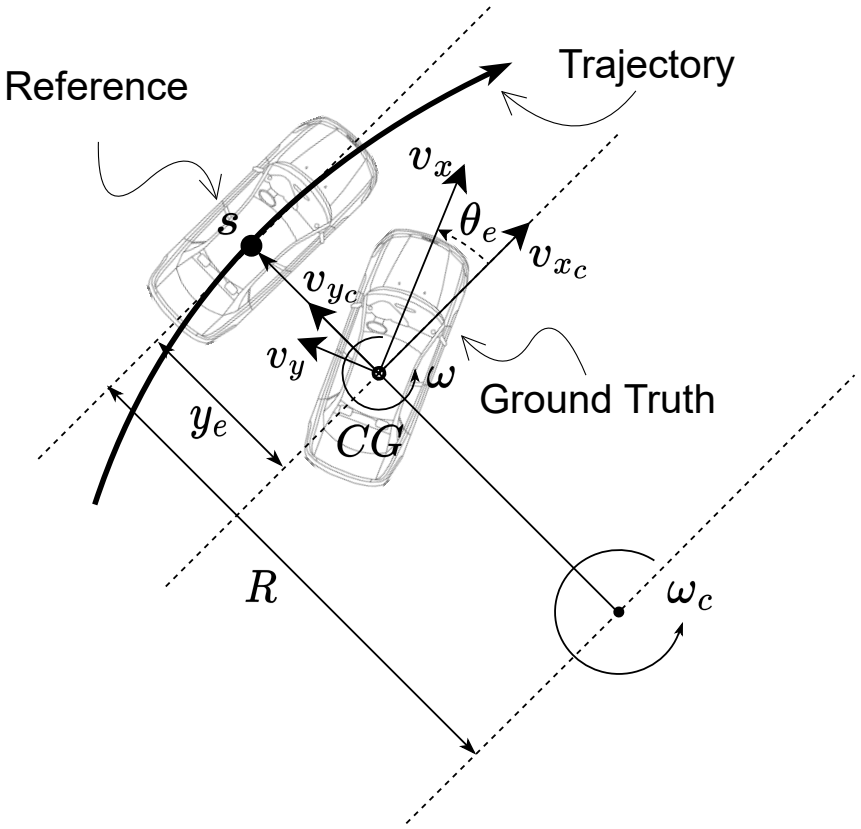


Figure 2.2: Diagram of the curvature-based kinematic model.

To better illustrate this model, the diagram presented in Figure 2.2 shows the linear and angular velocities of a vehicle (v_x, v_x and ω) expressed in local coordinates, as well as the variables used to represent the motion of the vehicle throughout the curvilinear trajectory:

- s: Represents the distance travelled along the curvilinear trajectory.
- y_e: Represents the lateral error of the vehicle, which is perpendicular to the slope of the trajectory at a distance s.

- θ_e: Represents the orientation error between the ground truth orientation of the vehicle and the desired one, which is determined by the angle of the slope of the trajectory at a distance s.

The derivation of the differential equations describing the curvature-based kinematic model are obtained from [16]. This deduction starts by computing the velocity headed in the direction of the trajectory's slope, v_{x_c}, as shown in (2.2).

$$v_{x_c} = (R - y_e)\omega_c = v_x \cos\theta_e - v_y \sin\theta_e \tag{2.2}$$

Where R is the radius of the trajectory curve at a distance s and ω_c is the desired angular velocity. By defining $\kappa = 1/\ R$ as the curvature of the road at a distance s, ω_c can be isolated from (2.2), as shown in (2.3).

$$\omega_c = \frac{v_x \cos\theta_e - v_y \sin\theta_e}{1 - y_e\kappa}\kappa \tag{2.3}$$

Thus, the time derivative of the orientation error, θ_e, can be deduced as shown in (2.4).

$$\dot{\theta}_e = \omega - \omega_c = \omega - \frac{v_x \cos\theta_e - v_y \sin\theta_e}{1 - y_e\kappa}\kappa \tag{2.4}$$

Also, the time derivative of the distance travelled along the curvilinear trajectory can be deduced from ω_c, as shown in (2.5).

$$\dot{s} = R\omega_c = \frac{v_x \cos\theta_e - v_y \sin\theta_e}{1 - y_e\kappa} \tag{2.5}$$

Finally, the time derivative of the lateral error, y_e, can be directly obtained by computing v_{y_c}, the velocity headed in the direction perpendicular to the trajectory slope, as shown in (2.6).

$$\dot{y}_e = v_{y_c} = v_x \sin\theta_e + v_y \cos\theta_e \tag{2.6}$$

Therefore, the curvature-based kinematic model is given by clustering Equations (2.4), (2.5) & (2.6), as shown in (2.7).

$$\begin{cases} \dot{y}_e = v_x \sin\theta_e + v_y \cos\theta_e & \text{(2.7a)} \\ \dot{\theta}_e = \omega - \dfrac{v_x \cos\theta_e - v_y \sin\theta_e}{1 - y_e\kappa}\kappa & \text{(2.7b)} \\ \dot{s} = \dfrac{v_x \cos\theta_e - v_y \sin\theta_e}{1 - y_e\kappa} & \text{(2.7c)} \end{cases}$$

2.2 Dynamic Model

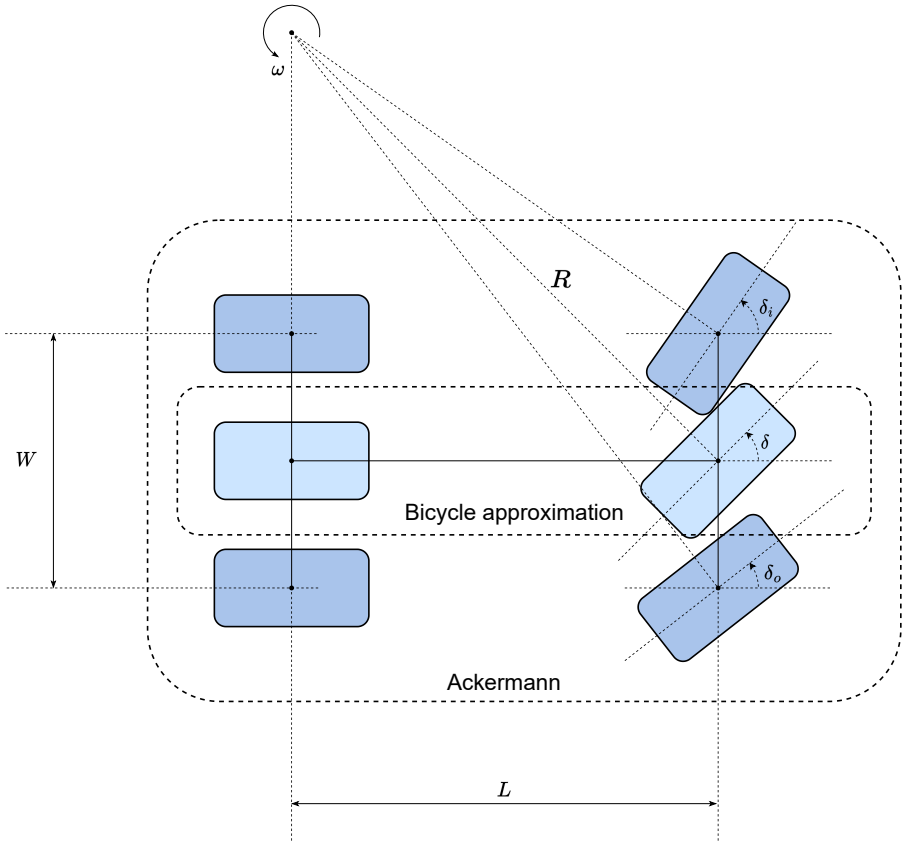


Figure 2.3: Diagram of the Ackermann model and its bicycle model approximation.

Most cars can be modelled after the Ackerman dynamic model, in which the two front wheels are steered, while the two rear wheels are fixed. The angle of each front wheel, while approximately close, are not equal due to each wheel travelling through an arc of different radius, as shown in Figure 2.3.

For small wheel slip angles, the inner and outer steering angles, δ_i and δ_o, can be approximately computed by (2.8), where the effective radius of each wheel has been considered.

$$\delta_i \approx \frac{L}{R - \frac{W}{2}} \tag{2.8a}$$

$$\delta_o \approx \frac{L}{R + \frac{W}{2}} \tag{2.8b}$$

Where L, W and R are the vehicle's length, width and turning radius, as shown in Figure 2.3.The average and difference between steering angle can be computed using (2.9) [13].

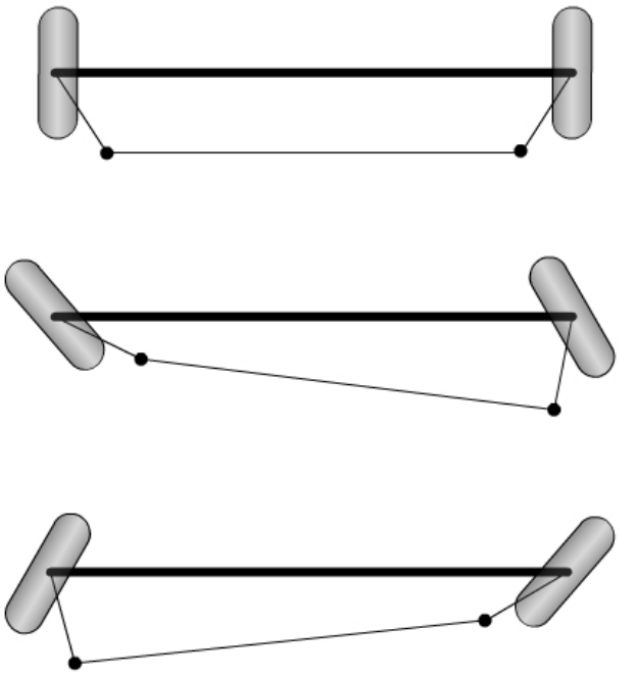


Figure 2.4: Differential steer from a trapezoidal tie-rod arrangement [13]

.

$$\delta_{Ack_{average}} = \frac{\delta_i + \delta_o}{2} \approx \frac{L}{R} \tag{2.9a}$$

$$\delta_{Ack_{difference}} = \delta_i - \delta_o \approx \delta^2_{Ack_{average}} \frac{W}{L} \tag{2.9b}$$

In practice, this steering angle difference is accomplished by using a trapezoidal tie rod arrangement [13], as shown in Figure 2.4, where the inner wheel always turns with a larger steering angle.

In many applications, the Ackermann model can be approximated by using the bicycle model, as shown in Figure 2.3, in which the two front and two rear wheels are reduced to one front and one rear wheel. This approximation is common as the steering angle derived from the bicycle model, shown in (2.10), is approximately equal for small wheel angle slips to the average steering angle of the Ackermann model, shown in (2.9a) [13].

$$\delta_{bicycle} \approx \delta_{Ack_{average}} \approx \frac{L}{R} \tag{2.10}$$

Therefore, the dynamic model assumed in this thesis will be the bicycle dynamic model, of which a complete diagram is shown in Figure 2.5. Unlike the kinematic models previously presented, the bicycle dynamic model is more appropriate for working with higher velocities due to its non-assumption of negligi-

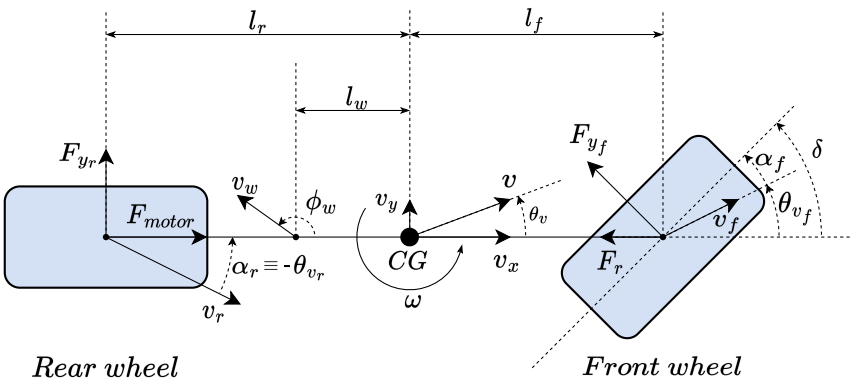


Figure 2.5: Diagram of the Cartesian dynamic model.

ble lateral tire forces acting on the front and rear wheels. This assumption can be made due to the inclusion of the of the lateral tire force model.

2.2.1 Lateral Tire Force Model

The lateral tire force model is a crucial element in high velocity control application since it is linked with the vehicle's traction to the road. High lateral tire forces could lead to a complete loss of traction, which could prove fatal. The lateral force in a tire is a function of the tire slip angle, which is defined as the angle between the orientation of the tire and the orientation of the velocity vectors of the wheels [13].

The diagram presented in Figure 2.5 shows the lateral force vectors, F_{y_f} and F_{y_r}, the front and rear slip angles, α_f and α_r, as well as the velocity vectors of each wheel, v_f and v_r. The slip angles can be computed using (2.11).

$$\alpha_f = \delta - \theta_{v_f} = \delta - \arctan\left(\frac{v_y + l_f\omega}{v_x}\right) \tag{2.11a}$$

$$\alpha_r = -\theta_{v_r} = -\arctan\left(\frac{v_y - l_r\omega}{v_x}\right) \tag{2.11b}$$

Where δ is the steering angle, θ_{v_f} and θ_{v_r} are the angles that the velocity vectors of each wheel forms with the bicycle's longitudinal axis, l_f and l_r are the distances from the CG to the front and rear tires, and v_x, v_x and ω are the longitudinal, lateral and angular velocities respectively.

As previously commented, the lateral tire forces depend on the tire slip angle, as shown in Figure 2.6 [13]. This dependency is approximately linear for small slip angles but non-linear as the slip angle increases. As such, two lateral tire force models can be considered, depending on the expected slip angles or the level of fidelity.

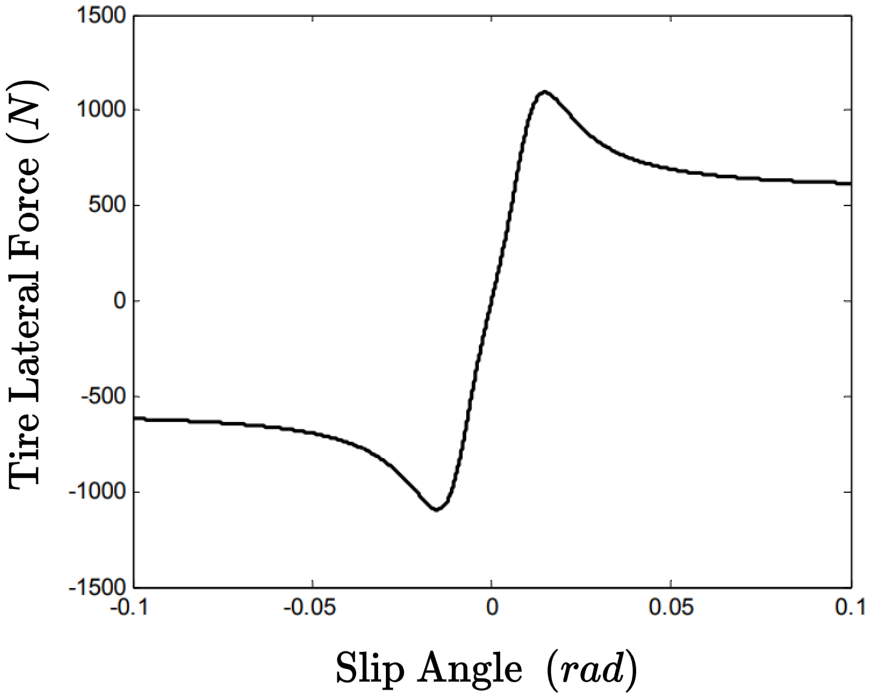


Figure 2.6: Lateral tire force curve obtained from Pacejka's Magic Formula [13].

Lateral Tire Force Model for Small Slip Angles.

For small slip angles, the front and rear lateral tire forces, can be computed by the linear function shown in (2.12).

$$F_{y_f} = C_f \alpha_f \tag{2.12a}$$

$$F_{y_r} = C_r \alpha_r \tag{2.12b}$$

Where C_f and C_r are the front and rear cornering stiffness.

Lateral Tire Force Model for Large Slip Angles.

In cases where the slip angles become larger a more sophisticated model is required. However, since analytical tire models do not always lead to quantitatively accurate results, semi-empirical tire models such as Dugoff's tire model or Pacejka's Magic Formula have been used [17]. In this thesis, the Magic Formula shown in (2.13)[18] will be considered as the lateral tire force model.

$$F_y(\alpha) = D \sin\left(C \arctan\left(B\alpha - E\left(B\alpha - \arctan\left(B\alpha\right)\right)\right)\right) \tag{2.13}$$

Where the model parameters B, C, D and E have the following nomenclature (a detailed explanation of each parameter is found in [18]):

- B: Stiffness factor.
- C: Shape factor.
- D: Peak value.
- E: Curvature factor.

2.2.2 Bicycle Dynamic Model

Having defined the lateral tire dynamics, the bicycle dynamic model can be derived. This model is composed of the bicycle's longitudinal, lateral and yaw dynamics.

Longitudinal Dynamic Model

The bicycle's longitudinal dynamics in the CG can be obtained by applying Newton's second law in the longitudinal direction, as shown in (2.14).

$$a_x = \dot{v}_x - \omega v_y = \frac{\sum F_x}{m} \tag{2.14}$$

Where a_x is the inertial acceleration in the longitudinal direction, m is the vehicles mass and v_x, v_x and ω are the longitudinal, lateral and angular velocities respectively. The sum of forces acting on the longitudinal direction is shown in (2.15).

$$\sum F_x = F_{motor} - F_{y_f} \sin\delta - F_{res_x} \tag{2.15}$$

Where F_{motor} is the force generated by the vehicle's motor or breaks, F_{y_f} is the front lateral tire force, which can be computed using (2.12a) or (2.13), and F_{res_x} is the longitudinal resistive force.

The force generated by the vehicle can be computed using (2.16), where a is the rear wheel's acceleration.

$$F_{motor} = ma \tag{2.16}$$

The longitudinal resistive force is composed of F_f, the force due to the rolling resistance at the tires due to friction, F_{d_x}, the longitudinal aerodynamic drag force and F_g, the gravitational force due to the grade of the road. This resistive force can be computed using (2.17).

$$F_{res_x} = F_f + F_{d_x} + F_g = \tag{2.17a}$$
$$F_f = m\mu g \tag{2.17b}$$
$$F_{d_x} = \operatorname{sgn}(v_x - v_{w_x})\frac{1}{2}\rho C_{d_{long}} A_{r_{long}} (v_x - v_{w_x})^2 \tag{2.17c}$$
$$F_g = mg\sin\phi_{road} \tag{2.17d}$$

Where μ is the static friction coefficient, g is the gravitational constant, v_{w_x} is the longitudinal wind velocity, $C_{d_{long}}$ is the aerodynamic drag coefficient in the longitudinal direction, ρ is the mass density of air, $A_{r_{long}}$ is the frontal area of the vehicle and ϕ_{road} is the angle of inclination of the road on which the vehicle is traveling.

By substituting Equations (2.15), (2.16) and (2.17) in (2.14) and then isolating $\dot{v}_x$, we obtain the longitudinal dynamic model shown in (2.18).

$$\dot{v}_x = a - \frac{F_{y_f}\sin\delta}{m} - \frac{F_{res_x}}{m} + \omega v_y \tag{2.18}$$

Lateral Dynamic Model

The bicycle's lateral dynamics in the CG can be obtained by applying Newton's second law in the lateral direction, as shown in (2.19).

$$a_y = \dot{v}_y + \omega v_x = \frac{\sum F_y}{m} \tag{2.19}$$

Where a_y is the inertial acceleration in the lateral direction. The sum of forces acting on the lateral direction is shown in (2.20).

$$\sum F_y = F_{y_f}\cos\delta + F_{y_r} - F_{res_y} \tag{2.20}$$

Where F_{y_f} and F_{y_r} are the front and rear lateral tire forces, which can be computed using (2.12) or (2.13), and F_{res_y} is the lateral resistive force.

The lateral resistive force is only composed of F_{d_y}, the lateral aerodynamic drag force, which can be computed using (2.21).

$$F_{res_y} = F_{d_y} = \operatorname{sgn}(v_y - v_{w_y})\frac{1}{2}\rho C_{d_{lat}} A_{r_{lat}} (v_y - v_{w_y})^2 \tag{2.21}$$

Where v_{w_y} is the lateral wind velocity, $C_{d_{lat}}$ is the aerodynamic drag coefficient in the lateral direction and $A_{r_{lat}}$ is the lateral area of the vehicle.

By substituting Equations (2.20) and (2.21) in (2.19) and then isolating $\dot{v}_y$, we obtain the lateral dynamic model shown in (2.22).

$$\dot{v}_y = \frac{F_{y_f}\cos\delta}{m} + \frac{F_{r_y}}{m} - \frac{F_{res_y}}{m} - \omega v_x \tag{2.22}$$

Yaw Dynamic Model

The bicycle's yaw dynamics in the CG can be obtained by applying Newton's second law for rotation in the axis perpendicular to the plane of the road, as shown in (2.23).

$$\dot{\omega} = \frac{\sum \tau}{I_z} \tag{2.23}$$

Where I_z is the moment of inertia of the vehicle in the CG. The sum of torques acting on the vehicle is shown in (2.24).

$$\sum \tau = F_{y_f} l_f \cos\delta + F_{res_y} l_w - F_{y_r} l_r \tag{2.24}$$

Where l_w is the distance in the longitudinal direction from the CG to the point where the wind is exerting its force, as shown in the diagram presented in Figure 2.5.

By substituting (2.24) in (2.23), we obtain the yaw dynamic model shown in (2.25).

$$\dot{\omega} = \frac{F_{y_f} l_f \cos\delta + F_{res_y} l_w - F_{y_r} l_r}{I_z} \tag{2.25}$$

Finally, the complete bicycle dynamic model can be computed by combining Equations (2.18), (2.22) and (2.25), as shown in (2.26).

$$\begin{cases} \dot{v}_x = a - \dfrac{F_{y_f} \sin\delta}{m} - \dfrac{F_{res_x}}{m} + \omega v_y & \text{(2.26a)} \\ \dot{v}_y = \dfrac{F_{y_f} \cos\delta}{m} + \dfrac{F_{r_y}}{m} - \dfrac{F_{res_y}}{m} - \omega v_x & \text{(2.26b)} \\ \dot{\omega} = \dfrac{F_{y_f} l_f \cos\delta + F_{res_y} l_w - F_{y_r} l_r}{I_z} & \text{(2.26c)} \end{cases}$$

2.3 Simulation and Control Oriented Models

In this section, two different models will be defined for control and simulation purposes.

2.3.1 Simulation Oriented Model

The simulation-oriented model (SOM) presented in (2.27), is obtained by combining the mass-point kinematic model shown in (2.1), the curvature-based kinematic model shown in (2.7) and the bicycle dynamic model shown in (2.26). This non-linear model will simulate the full dynamics of the vehicle by means of an ordinary differential equations (ODE) solver.

$$
\begin{cases}
\dot{v}_x = a - \dfrac{F_{y_f}\sin\delta}{m} - \dfrac{F_{res_x}}{m} + \omega v_y & \text{(2.27a)}\\
\dot{v}_y = \dfrac{F_{y_f}\cos\delta}{m} + \dfrac{F_{r_y}}{m} - \dfrac{F_{res_y}}{m} - \omega v_x & \text{(2.27b)}\\
\dot{\omega} = \dfrac{F_{y_f} l_f \cos\delta + F_{res_y} l_w - F_{y_r} l_r}{I_z} & \text{(2.27c)}\\
\dot{y}_e = v_x \sin\theta_e + v_y \cos\theta_e & \text{(2.27d)}\\
\dot{\theta}_e = \omega - \dfrac{v_x\cos\theta_e - v_y\sin\theta_e}{1 - y_e\kappa}\kappa & \text{(2.27e)}\\
\dot{s} = \dfrac{v_x\cos\theta_e - v_y\sin\theta_e}{1 - y_e\kappa} & \text{(2.27f)}\\
\dot{X}_G = v_x\cos\theta_G - v_y\sin\theta_G & \text{(2.27g)}\\
\dot{Y}_G = v_x\sin\theta_G + v_y\cos\theta_G & \text{(2.27h)}\\
\dot{\theta}_G = \omega & \text{(2.27i)}
\end{cases}
$$

2.3.2 Control Oriented Model

The control-oriented model (COM) will be used by the vehicle coordination algorithm in order to safely maneuver the vehicle around the road, while avoiding other vehicles. This model will differ from the SOM in two main aspects:

- *Reduction of differential equations*: the COM will not require the mass-point kinematic model presented in (2.1), as it does not add any relevant information useful for the vehicle coordination algorithm.
- *Simplified dynamics*: the dynamics of the bicycle model presented in (2.26) will be simplified in order to test the robustness of the designed vehicle coordination algorithm under the perturbations caused by the unmodelled dynamics.

The simplified bicycle dynamic model is presented in (2.28).

$$
\begin{cases}
\dot{v}_x = a - \dfrac{F_{y_f}\sin\delta}{m} - \dfrac{\tilde{F}_{res_x}}{m} + \omega v_y & \text{(2.28a)}\\
\dot{v}_y = \dfrac{F_{y_f}\cos\delta}{m} + \dfrac{F_{r_y}}{m} - \omega v_x & \text{(2.28b)}\\
\dot{\omega} = \dfrac{F_{y_f} l_f \cos\delta - F_{y_r} l_r}{I_z} & \text{(2.28c)}
\end{cases}
$$

Where two simplifications have been made: the longitudinal resistive forces caused by the angle of inclination of the road and the wind are not considered, and the lateral resistive forces are neglected, as presented in (2.29).

$$F_{res_x} \approx \tilde{F}_{res_x} = F_f + \tilde{F}_{d_x} = m\mu g + \text{sgn}\,(v_x)\frac{1}{2}C_{d_{long}}\rho A_{r_{long}}\,(v_x)^2 \tag{2.29a}$$

$$F_{res_y} \approx 0 \tag{2.29b}$$

Therefore, the COM presented in (2.30), is obtained by combining the curvature-based kinematic model shown in (2.7) and a simplified dynamic of the bicycle dynamic model shown in (2.28).

$$\begin{cases} \dot{v}_x = a - \dfrac{F_{y_f}\sin\delta}{m} - \dfrac{\tilde{F}_{res_x}}{m} + \omega v_y & \text{(2.30a)} \\ \dot{v}_y = \dfrac{F_{y_f}\cos\delta}{m} + \dfrac{F_{r_y}}{m} - \dfrac{F_{res_y}}{m} - \omega v_x & \text{(2.30b)} \\ \dot{\omega} = \dfrac{F_{y_f} l_f \cos\delta + F_{res_y} l_w - F_{y_r} l_r}{I_z} & \text{(2.30c)} \\ \dot{y}_e = v_x \sin\theta_e + v_y \cos\theta_e & \text{(2.30d)} \\ \dot{\theta}_e = \omega - \dfrac{v_x\cos\theta_e - v_y\sin\theta_e}{1 - y_e\kappa}\kappa & \text{(2.30e)} \\ \dot{s} = \dfrac{v_x\cos\theta_e - v_y\sin\theta_e}{1 - y_e\kappa} & \text{(2.30f)} \end{cases}$$

Finally, the model parameters, as well as the bounds of the dynamic state used for the COM and SOM are presented in Table 2.1 and 2.2, which were obtained from the driverless UPC racing car presented in [19].

Parameter	Value	Parameter	Value
l_f	$0.902\ m$	L	$4.2\ m$
l_r	$0.638\ m$	W	$1.8\ m$
m	$196\ kg$	C_f	$25\cdot 10^3\ \frac{N}{rad}$
I_z	$93\ kg\cdot m^2$	C_r	$25\cdot 10^3\ \frac{N}{rad}$
$C_{d_{long}}\cdot A_{r_{long}}$	1.64	B	-17.3065
$C_{d_{lat}}\cdot A_{r_{lat}}$	1.82	C	1.1804
ρ	$1.225\ \frac{kg}{m^3}$	D	$1.2246\cdot 10^3 N$
μ	0.015	E	0
g	$9.81\ \frac{m}{s^2}$		

Table 2.1: Dynamic model parameters of the driverless UPC Car [19].

Variable	Lower Bound	Upper bound
v_x	$1\ \frac{m}{s}$	$15\ \frac{m}{s}$
v_y	$-1\ \frac{m}{s}$	$1\ \frac{m}{s}$
ω	$-\frac{\pi}{2}\ \frac{rad}{s}$	$\frac{\pi}{2}\ \frac{rad}{s}$
a	$-2\ \frac{m}{s^2}$	$13\ \frac{m}{s^2}$
δ	$-0.25\ rad$	$0.25\ rad$
Δa	$-0.5\ \frac{m}{s^2}$	$0.5\ \frac{m}{s^2}$
$\Delta\delta$	$-0.05\ rad$	$0.05\ rad$

Table 2.2: Lower and upper bounds of the states and inputs [19].

Where $\Delta\delta$ and Δa are the maximum increments allowed on the control actions during a time interval of $T_s = 30\ ms$. The bounds on the kinematic states of the COM and SOM are not presented in this Chapter, as they are defined by the road characteristics and not by the vehicle.

Chapter 3

LPV Representation of the Vehicle Models

In this chapter, the LPV representation of the vehicle models presented in Chapter 2 are formulated using the non-linear embedding approach. The goal of the this approach is to embed the non-linearities of the original equations into a series of varying parameters, so that the resulting LPV model is formulated in a state space (SS) representation, as shown in (3.1), in which the matrices are a function of these varying parameters.

$$\dot{x} = A\left(\theta\left(\zeta\right)\right) x + B\left(\theta\left(\zeta\right)\right) u \tag{3.1}$$

Where x and u are vectors containing the system's states and inputs, $A\left(\theta\left(\zeta\right)\right)$ and $B\left(\theta\left(\zeta\right)\right)$ are the LPV matrices, θ is a vector containing the matrices' varying parameters and ζ is a vector containing the scheduling variables used to compute these varying parameters.

3.1 Curvature-based Kinematic LPV Model

The kinematic states and inputs of the curvature-based kinematic LPV model are defined as in (3.2).

$$x = \begin{bmatrix} y_e \\ \theta_e \\ s \end{bmatrix}, \quad u = \begin{bmatrix} v_x \\ v_y \\ \omega \end{bmatrix} \tag{3.2}$$

Such that the SS representation obtained from the non-linear embedding approach can be expressed as in (3.1). In order to compute the LPV matrices, the non-linear embedding approach must be applied to the kinematic model presented in (2.7), as shown in (3.3).

$$\dot{x} = A(v_x, v_y, y_e, \kappa)x + B(y_e, \theta_e, \kappa)u \tag{3.3a}$$

Where the LPV matrices are presented in (3.3b).

$$A(v_x, v_y, y_e, \kappa) = \begin{bmatrix} 0 & A_{12} & 0 \\ 0 & A_{22} & 0 \\ 0 & A_{32} & 0 \end{bmatrix}, \quad B(y_e, \theta_e, \kappa) = \begin{bmatrix} B_{11} & B_{12} & 0 \\ B_{21} & B_{22} & 1 \\ B_{31} & B_{32} & 0 \end{bmatrix} \tag{3.3b}$$

And the varying parameters are defined in (3.3c).

$$\begin{aligned}
A_{12} &= \frac{1}{2}\frac{\sin(\theta_e)}{\theta_e}v_x \approx \frac{1}{2}v_x \\
A_{22} &= \frac{1}{2}\frac{v_y}{1 - y_e\kappa}\frac{\sin(\theta_e)}{\theta_e}\kappa \approx \frac{1}{2}\frac{v_y}{1 - y_e\kappa}\kappa \\
A_{32} &= -\frac{1}{2}\frac{v_y}{1 - y_e\kappa}\frac{\sin(\theta_e)}{\theta_e} \approx -\frac{1}{2}\frac{v_y}{1 - y_e\kappa} \\
B_{11} &= \frac{1}{2}\sin(\theta_e) \\
B_{12} &= \cos(\theta_e) \\
B_{21} &= -\frac{\cos(\theta_e)}{1 - y_e\kappa}\kappa \\
B_{22} &= \frac{1}{2}\frac{\sin(\theta_e)}{1 - y_e\kappa}\kappa \\
B_{31} &= \frac{\cos(\theta_e)}{1 - y_e\kappa} \\
B_{32} &= -\frac{1}{2}\frac{\sin(\theta_e)}{1 - y_e\kappa}
\end{aligned} \tag{3.3c}$$

It should be noted that, to avoid singularities in the matrix $A(v_x, v_y, y_e, \kappa)$ when $\theta_e = 0$, the small-angle approximation shown in (3.4) was applied to the varying parameters A_{12}, A_{22} and A_{32}. The error committed with this approximation is less than 1% for $-13.99° \leq \theta_e \leq 13.99°$ [20].

$$\frac{\sin(\theta_e)}{\theta_e} \approx 1 \tag{3.4}$$

3.2 Simplified Bicycle Dynamic LPV Model

The dynamic states and inputs of the simplified bicycle dynamic LPV model are defined as in (3.5).

$$x = \begin{bmatrix} v_x \\ v_y \\ \omega \end{bmatrix}, \quad u = \begin{bmatrix} a \\ \delta \end{bmatrix} \tag{3.5}$$

Such that the SS representation obtained from the non-linear embedding approach can be expressed as in (3.1). To compute the LPV matrices, the non-linear embedding approach must be applied to the simplified dynamic model presented in (2.28). However, this process is not trivial due to the strong non-linearities present in the lateral tire force model. As such, the following approximations are to be considered [21].

First, the tangent function present in (2.11), which is used to compute the slip angles α_f and α_r, is approximated by $\tan(\alpha) \approx \alpha$, as shown in (3.6).

$$\alpha_f = \delta - \arctan\left(\frac{v_y + l_f\omega}{v_x}\right) \approx \delta - \frac{v_y + l_f\omega}{v_x} \tag{3.6a}$$

$$\alpha_r = -\arctan\left(\frac{v_y - l_r\omega}{v_x}\right) \approx -\frac{v_y - l_r\omega}{v_x} \tag{3.6b}$$

The error committed with this approximation is less than 1% for $-9.91° \leq \delta \leq 9.91°$ [20]. This approximation is valid as typical slip angles remain within this interval [14].

Second, the Magic Formula presented in (2.13) is approximated by a n-order polynomial using a least-squares algorithm, as shown in (3.7).

$$F_y(\alpha) \approx p_n\alpha^n + p_{n-1}\alpha^{n-1} + \cdots + p_1\alpha + p_0 \tag{3.7}$$

Where p_i are the n coefficients of the induced polynomial, being n the polynomial's order.

With this polynomial, the non-linear embedding approach is applied to the Magic Formula as shown in (3.8).

$$F_y(\alpha) \approx C(\alpha)\alpha \tag{3.8a}$$

$$C(\alpha) = p_n\alpha^{n-1} + p_{n-1}\alpha^{n-2} + \cdots + p_1 + p_0\frac{1}{\alpha + \epsilon} \tag{3.8b}$$

Where $C(\alpha)$ is the tire cornering stiffness function obtained from the non-linear embedding approach and ϵ is a small constant used to avoid singularities in $C(\alpha)$ when $\alpha = 0$.

With the help of these two approximations, the simplified bicycle dynamic model presented in (2.28) can be approximated by (3.9), where the non-linearities have been embedded.

$$\begin{cases} \dot{v}_x = a - \dfrac{\tilde{F}_{res_x}}{mv_x}v_x + \dfrac{C_f \sin\delta}{mv_x}v_y + \left(\dfrac{C_f l_f \sin\delta}{mv_x} + v_y\right)\omega - \dfrac{C_f \sin\delta}{m}\delta & (3.9a) \\ \dot{v}_y = -\dfrac{C_f \cos\delta + C_r}{mv_x}v_y - \left(\dfrac{C_f l_f \cos\delta - C_r l_r}{mv_x} - v_x\right)\omega + \dfrac{C_f \cos\delta}{m}\delta & (3.9b) \\ \dot{\omega} = -\dfrac{C_f l_f \cos\delta - C_r l_r}{I_z v_x}v_y - \dfrac{C_f l_f^2 \cos\delta + C_r l_r^2}{I_z v_x}\omega + \dfrac{C_f l_f \cos\delta}{I_z}\delta & (3.9c) \end{cases}$$

Where the front and rear tire cornering stiffness parameters, C_f and C_r, are approximated by (3.8b). By using the approximations made to the dynamics equations shown in (3.9), the LPV model can be readily obtained, as shown in (3.10)

$$\dot{x} = A(v_x, v_y)x + B(\delta)u \tag{3.10a}$$

Where the LPV matrices are presented in (3.10b).

$$A(v_x, v_y) = \begin{bmatrix} A_{11} & A_{12} & A_{13} \\ 0 & A_{22} & A_{23} \\ 0 & A_{32} & A_{33} \end{bmatrix}, \quad B(\delta) = \begin{bmatrix} 1 & B_{12} \\ 0 & B_{22} \\ 0 & B_{32} \end{bmatrix} \tag{3.10b}$$

And the varying parameters are defined in (3.10c).

$$\begin{aligned} A_{11} &= -\frac{\tilde{F}_{res_x}}{mv_x} \\ A_{12} &= \frac{C_f \sin\delta}{mv_x} \\ A_{13} &= \frac{C_f l_f \sin\delta}{mv_x} + v_y \\ A_{22} &= -\frac{C_f \cos\delta + C_r}{mv_x} \\ A_{23} &= -\frac{C_f l_f \cos\delta - C_r l_r}{mv_x} - v_x \\ A_{32} &= -\frac{C_f l_f \cos\delta - C_r l_r}{I_z v_x} \\ A_{33} &= -\frac{C_f l_f^2 \cos\delta + C_r l_r^2}{I_z v_x} \\ B_{12} &= -\frac{C_f(\alpha) \sin\delta}{m} \\ B_{22} &= \frac{C_f(\alpha) \cos\delta}{m} \\ B_{32} &= \frac{C_f(\alpha) l_f \cos\delta}{I_z} \end{aligned} \tag{3.10c}$$

3.3 Control Oriented LPV Model

The COM shown in (2.30) can be transform into the LPV form by combining Equations (3.3) and (3.10), as presented in (3.11).

$$\dot{x} = A(v_x, v_y, y_e, \kappa)x + B(y_e, \theta_e, \kappa, \delta)u \tag{3.11a}$$

Where the state and input vectors are shown in (3.11b).

$$x = \begin{bmatrix} v_x \\ v_y \\ \omega \\ y_e \\ \theta_e \\ s \end{bmatrix}, \quad u = \begin{bmatrix} a \\ \delta \end{bmatrix} \tag{3.11b}$$

The LPV matrices are presented in (3.11c) and (3.11d).

$$A(v_x, v_y, y_e, \kappa) = \begin{bmatrix} A_{11} & A_{12} & A_{13} & 0 & 0 & 0 \\ 0 & A_{22} & A_{23} & 0 & 0 & 0 \\ 0 & A_{32} & A_{33} & 0 & 0 & 0 \\ A_{41} & A_{42} & 0 & 0 & A_{45} & 0 \\ A_{51} & A_{52} & 1 & 0 & A_{55} & 0 \\ A_{61} & A_{62} & 0 & 0 & A_{65} & 0 \end{bmatrix} \tag{3.11c}$$

$$B(y_e, \theta_e, \kappa, \delta) = \begin{bmatrix} 1 & B_{12} \\ 0 & B_{22} \\ 0 & B_{32} \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix} \tag{3.11d}$$

And the varying parameters are defined in (3.11e).

$$\begin{aligned} A_{11} &= -\frac{F_{res_x}}{m v_x} \\ A_{12} &= \frac{C_f \sin\delta}{m v_x} \\ A_{13} &= \frac{C_f l_f \sin\delta}{m v_x} + v_y \\ A_{22} &= -\frac{C_f \cos\delta + C_r}{m v_x} \\ A_{23} &= -\frac{C_f l_f \cos\delta - C_r l_r}{m v_x} - v_x \\ A_{32} &= -\frac{C_f l_f \cos\delta - C_r l_r}{I_z v_x} \end{aligned}$$

$$
\begin{aligned}
A_{33} &= -\frac{C_f l_f^2 \cos\delta + C_r l_r^2}{I_z v_x} \\
A_{45} &= \frac{1}{2} v_x \\
A_{55} &= \frac{1}{2}\frac{v_y}{1 - y_e\kappa}\kappa \\
A_{65} &= -\frac{1}{2}\frac{v_y}{1 - y_e\kappa} \\
A_{41} &= \frac{1}{2}\sin(\theta_e) \\
A_{42} &= \cos(\theta_e) \\
A_{51} &= -\frac{\cos(\theta_e)}{1 - y_e\kappa}\kappa \\
A_{52} &= \frac{1}{2}\frac{\sin(\theta_e)}{1 - y_e\kappa}\kappa \\
A_{61} &= \frac{\cos(\theta_e)}{1 - y_e\kappa} \\
A_{62} &= -\frac{1}{2}\frac{\sin(\theta_e)}{1 - y_e\kappa} \\
B_{12} &= -\frac{C_f(\alpha)\sin\delta}{m} \\
B_{22} &= \frac{C_f(\alpha)\cos\delta}{m} \\
B_{32} &= \frac{C_f(\alpha) l_f \cos\delta}{I_z}
\end{aligned}
\tag{3.11e}
$$

Finally, note that even though the presented LPV models were developed in continuous time (CT), their discrete time (DT) formulation can be obtained by using the Euler discretization method shown in (3.12).

$$
x[k+1] = A_d x[k] + B_d u[k] \approx (I + AT_s)x[k] + T_s B u[k] \tag{3.12}
$$

Where A_d and B_d are the DT SS matrices, A and B are the CT SS matrices and T_s is the sampling time.

Chapter 4

Polytopic LPV Representation of the Vehicle Models

The LPV models presented in Chapter 3 serve as a useful tool for obtaining linear models without linearization by embedding the non-linearities in the varying parameters ζ.

However, due to the unmodelled dynamics present in the COM shown in (2.30), which were the result of simplifying the dynamic bicycle model, a corrective controller is required for robustness.

This corrective controller can be formulated as a gain-scheduled multivariable controller, in which the stability and performance of the close-loop system is ensured by applying the Lyapunov theory. One approach to synthesized this controller is to apply linear matrix inequality (LMI) optimization techniques to the LPV COM system presented in (3.11).

However, the LMI optimization problem cannot be directly applied to a LPV system, like the one shown in (3.1), due to the fact that the optimization problem should be evaluated for all the possible numerical values that the LPV matrices could adopt. As a result, the optimization problem would become an infinite constraint LMI problem.

To get around this problem, the infinite constraints are reduced to a finite number of constraints, obtained by evaluating the optimization problem in each vertex of the system's polytopic LPV model, as e.g. [22]. Equation (4.1) presents a standard polytopic LPV model of a system.

$$\dot{x} = \sum_{i=1}^{N} \mu_i(\theta)(A_i x + B_i u) \tag{4.1}$$

Where the pair of matrices (A_i, B_i) represents one of the N vertices of the polytope that bounds all the numerical values of the LPV matrices, and $\mu_i(\theta)$ is the vertex membership function, which indicates the membership degree of the LPV model $\big(A(\theta), B(\theta)\big)$ in the (A_i, B_i) vertex. This function must satisfy the constraints

shown in (4.2).

$$\sum_{i=1}^{N} \mu_i(\theta) = 1, \quad \mu_i(\theta) \geq 0 \tag{4.2}$$

The next section will present algorithms capable of generating polytopic LPV models.

4.1 Generation of Polytopic LPV Model

Multiple algorithms have been proposed to compute the polytopic LPV model of a system, of which two will be presented in this section. The first one is the traditional bounding box method [23], which proves inefficient as a result of the conservatism introduced due to overbounding. While the second one is the principal component analysis (PCA) based parameter set mapping method [24], which is more efficient, as it yields tighter polytopes.

4.1.1 Bounding Box Method

The bounding box method starts by assuming that the measurement of each varying parameter of the LPV model, θ_j, is available in real-time and that their range is known a priori, as shown in (4.3).

$$\underline{\theta}_j \leq \theta_j \leq \overline{\theta}_j, \quad j = 1, \ldots, n_\theta \tag{4.3a}$$

$$\overline{\theta}_j = \max_{\zeta \in [\underline{\zeta}, \overline{\zeta}]} \theta_j(\zeta), \quad \underline{\theta}_j = \min_{\zeta \in [\underline{\zeta}, \overline{\zeta}]} \theta_j(\zeta) \tag{4.3b}$$

Where $\underline{\theta}_j$ and $\overline{\theta}_j$ are the lower and upper bounds of θ_j, found withing the scheduling variable's range $\zeta \in \left[\underline{\zeta}, \overline{\zeta}\right]$, and n_θ is the number of varying parameters.

If these assumptions are applicable, then the polytope Θ bounding the varying parameter vector can be defined as shown in (4.4).

$$\theta \in \Theta \coloneqq \mathrm{co}\left\{\Theta_1, \Theta_2, \ldots \Theta_N\right\} \tag{4.4}$$

Where each vertex Θ_i is formed by considering one of the $N = 2^{n_\theta}$ possible permutations of $\underline{\theta}_j$ and $\overline{\theta}_j$ for $j = 1, \ldots, n_\theta$ and $\mathrm{co}\{\cdot\}$ is defined as the convex hull operator of said vertices. This method is referred to as bounding box since the resulting bounding polytope has the shape of a hyperrectangle.

By using this bounding polytope, the varying parameter vector can be expressed in polytopic form, as shown in (4.5).

$$\theta = \left\{ \sum_{i=1}^{N} \mu_i(\theta)\Theta_i \;\middle|\; \sum_{i=1}^{N} \mu_i(\theta) = 1, \;\; \mu_i(\theta) \geq 0 \right\} \tag{4.5}$$

Note that if an affine function is applied to the varying parameter vector: $\delta(\theta)$, then its new bounding polytope can be obtained by computing the image of Θ under $\delta(\cdot)$, as shown in (4.6).

$$\delta(\theta) \in \delta(\Theta) \coloneqq \text{co}\left\{\delta(\Theta_1), \delta(\Theta_2), \ldots \delta(\Theta_N)\right\} \tag{4.6}$$

Since all the LPV matrices, $A(\theta)$ and $B(\theta)$, shown in Chapter 3, presented an affine relation with the varying parameter vector θ, the polytope bounding each LPV matrix can be computed as the image of Θ under $A(\cdot)$ and $B(\cdot)$, as shown in (4.7).

$$A(\theta) \in A(\Theta) \coloneqq \text{co}\left\{A_1, A_2, \ldots A_N\right\} \tag{4.7a}$$

$$B(\theta) \in B(\Theta) \coloneqq \text{co}\left\{B_1, B_2, \ldots B_N\right\} \tag{4.7b}$$

Where $A_i = A(\Theta_i)$ and $B_i = B(\Theta_i)$, for $i = 1, \ldots, N$. By using these two bounding polytopes, the LPV matrices can be expressed in polytopic form, as shown in (4.8).

$$A(\theta) = \left\{ \sum_{i=1}^{N} \mu_i(\theta)A_i \;\middle|\; \sum_{i=1}^{N} \mu_i(\theta) = 1, \;\; \mu_i(\theta) \geq 0 \right\} \tag{4.8a}$$

$$B(\theta) = \left\{ \sum_{i=1}^{N} \mu_i(\theta)B_i \;\middle|\; \sum_{i=1}^{N} \mu_i(\theta) = 1, \;\; \mu_i(\theta) \geq 0 \right\} \tag{4.8b}$$

Therefore, by substituting (4.8) in (3.1), as presented in (4.9), the desired LPV polytopic model shown in (4.1) is obtained.

$$\begin{aligned} \dot{x} &= A(\theta)x + B(\theta)u \\ &= \left(\sum_{i=1}^{N} \mu_i(\theta)A_i \right) x + \left(\sum_{i=1}^{N} \mu_i(\theta)B_i \right) u \\ &= \sum_{i=1}^{N} \mu_i(\theta)(A_i x + B_i u) \end{aligned} \tag{4.9}$$

Finally, the vertex membership function $\mu_i(\theta)$ can be computed as shown in (4.10) [21].

$$\mu_i(\theta) = \prod_{j=1}^{n_\theta} \xi_{ij}\left(\underline{\eta}_j, \overline{\eta}_j\right), \quad \forall i = 1, \ldots, N = 2^{n_\theta} \tag{4.10a}$$

$$\underline{\eta}_j = \frac{\overline{\theta}_j - \theta_j}{\overline{\theta}_j - \underline{\theta}_j} \tag{4.10b}$$

$$\overline{\eta}_j = \frac{\theta_j - \underline{\theta}_j}{\overline{\theta}_j - \underline{\theta}_j} = 1 - \eta_0^j \tag{4.10c}$$

Where $\xi_{ij}\left(\underline{\eta}_j, \overline{\eta}_j\right)$ is a function that alternates between $\underline{\eta}_j$ and $\overline{\eta}_j$, in order to compute the N possible combinations.

4.1.2 PCA-Based Parameter Set Mapping

The bounding box method, presented in the previous section, is a simple and straight forward method for computing the polytopic LPV model of a system. However, it presents a series of drawbacks [24]:

- *Poorly scalable*: the number of vertices of the polytopic model increases exponentially with the number of varying parameters: $N = 2^{n_\theta}$. Consequently, many non-linear systems with high number of varying parameters resort to ad hoc simplifications (such as neglecting dynamic coupling or fixing parameter dependent variables) in order to reduce the total number of varying parameters. Ultimately, a compromise between model accuracy and model tractability is considered.
- *Excessive conservatism*: Since the bounding polytope presented in (4.4) does not take into account the inherent couplings between different varying parameters, many of the polytope's vertices are unreachable by the system, which results in unnecessary overbounding.

To amend these disadvantages, algorithms like the PCA-based parameter set mapping [24] have been proposed, which try to mitigate the aforementioned drawbacks by reducing the number of varying parameters and tightening the bounding polytope so that only reachable vertices are considered.

The steps involve in the PCA-based parameter set mapping algorithm are: the generation of the trajectories of the varying parameters, the application of the PCA algorithm to these trajectories to reduce the number of varying parameters and finally, the computation of the reduced polytopic LPV model.

Generation of Varying Parameter Trajectories

First, this algorithm requires the generation of typical trajectories of the scheduling variables ζ, obtained by either measurements or simulation. These trajectories should cover the expected range of operation, which in the case of road vehicles consists of different driving maneuver such as: accelerating, braking or making

tight or wide turns. This step could also provide information about any inherent couplings present between the scheduling variables.

Equation (4.11) shows a trajectory formed by N_s measurements of the scheduling variable that have been sampled at a frequency of $f = 1/T_s$.

$$\Upsilon = \{\zeta[0],\ \zeta[T_s],\ \ldots,\ \zeta[kT_s],\ \zeta[(N_s-1)T_s]\} \tag{4.11}$$

The trajectories of the varying parameters can be generated by computing the image of the trajectory of scheduling variables, Υ, under the varying parameter mapping $\theta(\zeta)$, as shown in (4.12). These new trajectories could give information about any possible inherent couplings present between the varying parameters, due to scheduling variables appearing repeatedly in the mapping of different varying parameters.

$$\begin{aligned}\Xi &= \{\theta[0],\ \theta[T_s],\ \ldots,\ \theta[kT_s],\ \theta[(N_s-1)T_s]\} \\ &= \{\theta(\zeta[0]),\ \theta(\zeta[T_s]),\ \ldots,\ \theta(\zeta[kT_s]),\ \theta(\zeta[(N_s-1)T_s]\}\end{aligned} \tag{4.12}$$

In order to weight each varying parameter equally, each row of the trajectory matrix Ξ is normalize, as shown in (4.13), to achieve scaled - zero mean trajectories.

$$\Xi_i^{norm} = \mathcal{N}_i(\Xi_i) = \frac{\Xi_i - \mu_{\Xi_i}}{\sigma_{\Xi_i}} \tag{4.13}$$

Where Ξ_i is the i^{th} row of the trajectory matrix Ξ and μ_{Ξ_i} and σ_{Ξ_i} are, respectively, the mean and standard deviation of Ξ_i.

Reduction of the Number of Varying Parameter

The next step involves applying the PCA, in order to reduce the number of varying parameters. To do so, the singular value decomposition of the normalized trajectory matrix Ξ_i^{norm} is performed, as shown in (4.14). If these trajectories are correlated, then some of the singular values will be small in comparison with others and as such, they can be neglected.

$$\begin{aligned}\Xi^{norm} &= U\Sigma V \\ &= [U_s U_n] \begin{bmatrix} \Sigma_s & 0 & 0 \\ 0 & \Sigma_n & 0 \end{bmatrix} \begin{bmatrix} V_s^\mathsf{T} \\ V_n^\mathsf{T} \end{bmatrix}\end{aligned} \tag{4.14}$$

Where the matrices U_s, Σ_s and V_s correspond to the m most significant singular values, such that Ξ_i^{norm} can be approximated by (4.15).

$$\Xi_i^{norm} \approx \hat{\Xi}^{norm} = U_s \Sigma V_s^\mathsf{T} \tag{4.15}$$

By using the U_s matrix and the normalization presented in (4.13), the reduced vector of varying parameters ψ can be computed as shown in (4.16), where the number of varying parameters has been reduced from n_θ to $n_\psi = m$, being m the chosen number of most significant singular values.

$$\psi(\theta) = U_s^{\mathsf{T}} \mathcal{N}\left(\theta(\zeta)\right) \tag{4.16}$$

As a quality measurement of this approximation, the fraction of total variation shown in (4.17) can be applied.

$$v_m = \frac{\sum_{i=1}^{n_\psi} \sigma_i}{\sum_{i=1}^{n_\theta} \sigma_i} 100\% \tag{4.17}$$

Where σ_i is the i^{th} most significant singular value. The performed approximation should achieve a trade-off between a high percentage of the fraction of total variation and an effective reduction of the number of varying parameters.

Finally, Equation (4.18) is used to transform from the reduced varying parameter vector $\psi(\zeta)$ to the original varying parameter vector $\theta(\zeta)$.

$$\hat{\theta}(\psi) = \mathcal{N}^{-1}\left(U_s \psi(\zeta)\right) \tag{4.18}$$

Where $\mathcal{N}^{-1}(\cdot)$ rescales the data down to the original mean μ_Ξ and standard deviation σ_Ξ presented in (4.13).

Reduced Polytopic LPV Model

To compute the reduced polytopic LPV model, the upper and lower bounds of the reduced varying parameter vector $\psi(\zeta)$ are computed. However, unlike in the bounding box method shown in (4.3), where the bounds were computed ignoring the possible inherent coupling between parameters, these new bounds are searched throughout the generated trajectories of the scheduling variable Υ, as presented in (4.19). As a result, these new bounds are less conservative as unreachable dynamics are not considered in the bound's computation.

$$\underline{\psi}_j^{\Upsilon} \leq \psi_j \leq \overline{\psi}_j^{\Upsilon}, \quad j = 1, \ldots, n_\psi \tag{4.19a}$$

$$\overline{\psi}_j^{\Upsilon} = \max_k \psi_j\left(\zeta\left[kT_s\right]\right), \quad \underline{\psi}_j^{\Upsilon} = \min_k \psi_j\left(\zeta\left[kT_s\right]\right), \quad k = 0, \ldots, (N_s - 1) \tag{4.19b}$$

With these bounds, a new polytope Ψ bounding the reduced varying parameter vector can be defined as shown in (4.20).

$$\psi \in \Psi := \mathrm{co}\left\{\Psi_1^{\Upsilon}, \Psi_2^{\Upsilon}, \ldots \Psi_M^{\Upsilon}\right\} \tag{4.20}$$

Where each vertex Ψ_i^{Υ} is formed by considering one of the $M = 2^{n_\psi}$ possible permutations of $\underline{\psi}_j^{\Upsilon}$ and $\overline{\psi}_j^{\Upsilon}$ for $j = 1, \ldots, n_\psi$. Then, applying the transformation shown in (4.18) to the vertices Ψ_i^{Υ} leads to the polytope shown in (4.21), which is a lower order approximation of Θ, the overbounding polytope computed in (4.4) using the bounding box method.

$$\theta \in \hat{\Theta} \coloneqq \text{co}\left\{\hat{\Theta}_1, \hat{\Theta}_2, \ldots \hat{\Theta}_M\right\} = \text{co}\left\{\hat{\theta}(\Psi_1^{\Upsilon}), \hat{\theta}(\Psi_2^{\Upsilon}), \ldots \hat{\theta}(\Psi_M^{\Upsilon})\right\} \tag{4.21}$$

Finally, the reduced polytopic LPV model can be obtained by applying (4.22).

$$\dot{x} = \sum_{i=1}^{M} \mu_i(\theta)(\hat{A}_i x + \hat{B}_i u) \tag{4.22}$$

Where the vertex matrices $(\hat{A}_i, \hat{B}_i)$ can be computed as the image of $\hat{\Theta}$ under the affine functions $A(\cdot)$ and $B(\cdot)$, as shown in (4.23), where $\hat{A}_i = A(\hat{\Theta}_i)$ and $\hat{B}_i = B(\hat{\Theta}_i)$, for $i = 1, \ldots, M$.

$$A(\theta) \in A(\hat{\Theta}) \coloneqq \text{co}\left\{\hat{A}_1, \hat{A}_2, \ldots \hat{A}_M\right\} \tag{4.23a}$$

$$B(\theta) \in B(\hat{\Theta}) \coloneqq \text{co}\left\{\hat{B}_1, \hat{B}_2, \ldots \hat{B}_M\right\} \tag{4.23b}$$

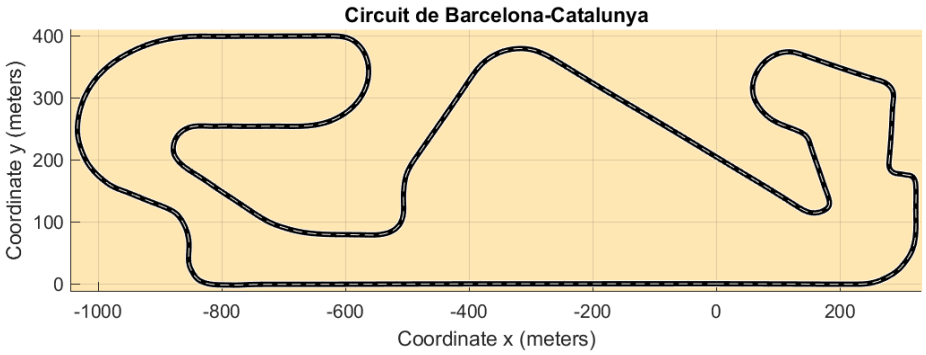
Thanks to the PCA-based parameter set mapping algorithm, the total number of vertices can be reduced by a factor of $2^{n_\theta - n_\psi}$

4.2 Polytopic LPV Model of the Bicycle Dynamic Model

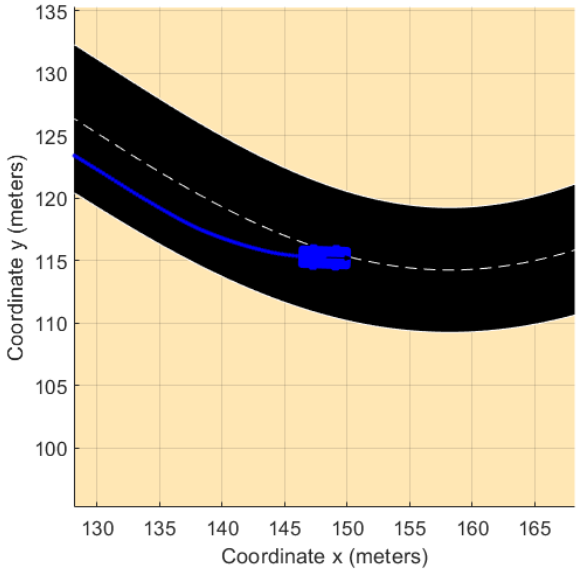
The polytopic LPV model of the bicycle dynamic model will be necessary in order to synthesized a robust LPV corrective controller using LMI optimization techniques, as presented in Chapter 5.

Since the LPV model of the bicycle dynamic system presented in (3.10) had ten varying parameters, the polynomic LPV model would present $N = 2^{10} = 1024$ vertices if the bounding box method was used to compute the bounding polytope.

Thus, to obtain the robust corrective controller, an optimization problem with at least 1024 constraints would have to be solved. Since this would result unfeasible or at least present a poor performance due to the excessive conservatism, the number of constraints is reduced by considering the vertices of the reduced polytope computed by the PCA parameter set mapping algorithm.



(a) The track *Circuit de Barcelona-Catalunya* used to generate the trajectories of the scheduling variable.



(b) Data acquisition obtained by manually driving the car.

Figure 4.1: Generation of the trajectories of scheduling variables.

As described in Section 4.1.2, the first step necessary to apply the PCA parameter set mapping algorithm is to compute a trajectory of the scheduling variables that covers their expected range of operation.

To do so, a simulation is performed using the SOM, with the parameters shown in Tables 2.1, in which the vehicle was manually driven through a circuit, as presented in Figure 4.1. This circuit has been designed by using the real-life coordinates of the track *Circuit de Barcelona-Catalunya*, which were obtained from [25].

The generated trajectories of the scheduling variables $\zeta = [v_x, v_y, \delta]^\mathsf{T}$ are shown in Figure 4.2. These trajectories contained a total number of 18619 samples which were sampled at a frequency of $30\ Hz$.

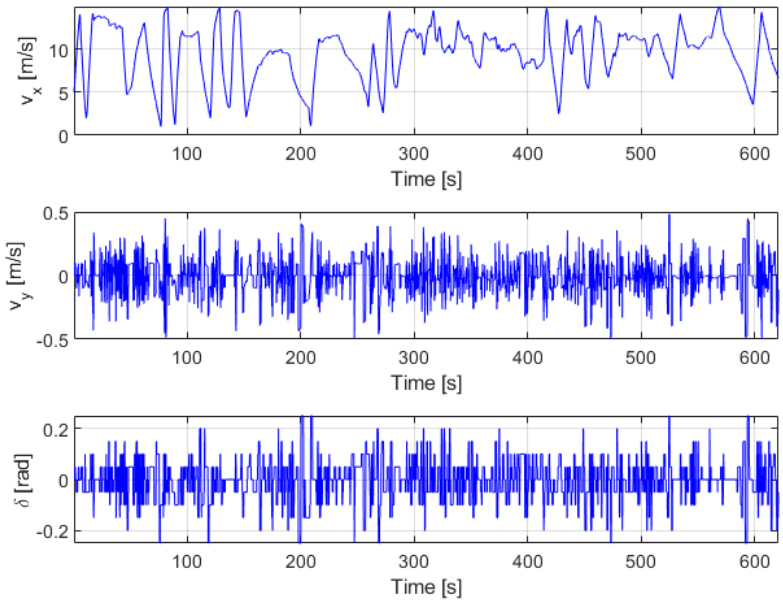


Figure 4.2: Trajectories of the scheduling variable Υ.

As presented in (4.12), the trajectories of the varying parameters can then be computed by evaluating (3.10c) for the generated trajectories of the scheduling variables. As a result, the trajectories of the varying parameters shown in Figure 4.3 are obtained.

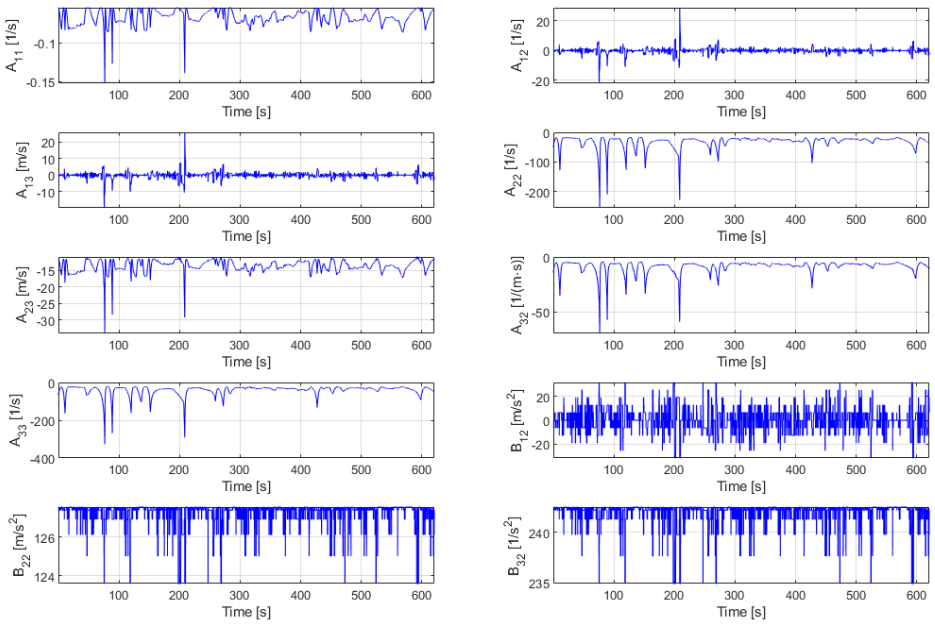


Figure 4.3: Trajectories of the varying parameters Ξ

The next step involves normalizing the trajectory of each varying parameter as indicated in (4.13). The trajectories resulting from this normalization are shown in Figure 4.4.

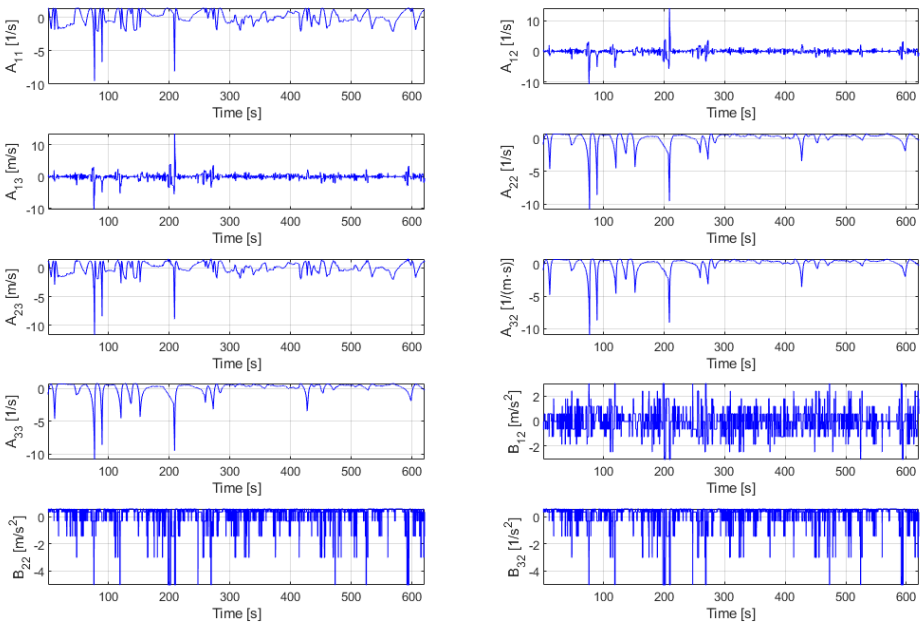


Figure 4.4: Normalized trajectories of the varying parameters Ξ^{norm}.

After applying the singular value decomposition to the normalized trajectory matrix, the singular values shown in (4.24) were obtained.

$$\Sigma = \begin{bmatrix} 2.5804 \cdot 10^2 \\ 2.1580 \cdot 10^2 \\ 2.0231 \cdot 10^2 \\ 1.6681 \cdot 10^2 \\ 6.5056 \cdot 10 \\ 4.6110 \\ 4.2121 \\ 7.0412 \cdot 10^{-9} \\ 6.2057 \cdot 10^{-14} \\ 3.4784 \cdot 10^{-14} \end{bmatrix} \tag{4.24}$$

If only the first five singular values are considered, then the fraction of total variation defined in (4.17), will present the value shown in (4.25).

$$v_m = \frac{\sum_{i=1}^{5} \sigma_i}{\sum_{i=1}^{10} \sigma_i} 100\% = 99.04\% \tag{4.25}$$

This high percentage indicates that around 99% of the information contained in the normalized trajectories of the varying parameters is represented by the first five singular values. Thus, by applying the mapping shown in (4.16), the number of varying parameters can be reduced to five, while ensuring a good approximation of the model.

As a result, the number of vertices of the polytope bounding the varying parameters has been effectively reduced from the $N = 2^{10} = 1024$ vertices yielded by the bounding box method, to the $M = 2^5 = 32$ vertices computed by the PCA parameter set mapping method. These vertices can be computed by applying Equations (4.19), (4.20) and (4.21).

The graph presented in Figure 4.5 serves as an illustrative example of the capacity of the PCA parameter set mapping method of yielding tighter polytopes. Specifically, the three-dimensional polytope bounding the varying parameters A_{11}, A_{22} and A_{33} obtained by applying the bounding box method, was reduced to a two-dimensional polytope which better bounded the varying parameter trajectory shown in red.

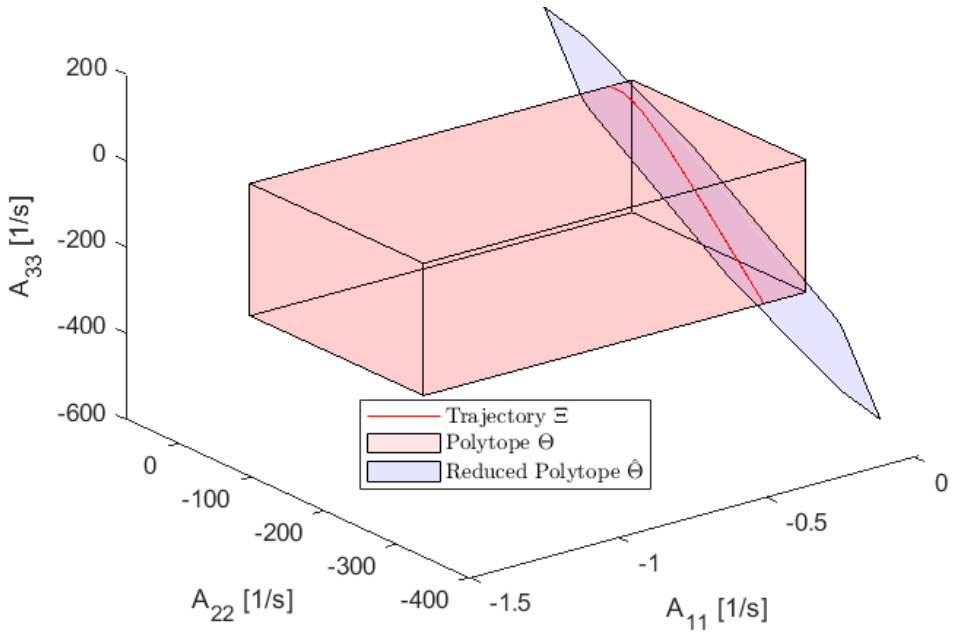


Figure 4.5: Bounding polytopes computed by the bounding box method (red) and the PCA parameter set mapping method (blue).

Finally, the vertices of the polytopic LPV model of the dynamic bicycle system were obtained by applying the affine transformation shown in Equations (3.11c) and (3.11d) to vertices of the reduced polytope $\hat{\Theta}$, as presented in (4.26).

$$\dot{x} = \sum_{i=1}^{32} \mu_i(\theta)(A_i x + B_i u) \tag{4.26a}$$

$$
A_1 = \begin{bmatrix} -0.1400 & -19.5224 & -17.7292 \\ 0 & -249.7994 & -31.8731 \\ 0 & -68.9487 & -321.1669 \end{bmatrix}, \quad B_1 = \begin{bmatrix} 1 & 8.1807 \\ 0 & 129.2453 \\ 0 & 245.6939 \end{bmatrix}, \quad A_2 = \begin{bmatrix} -0.1446 & -7.2003 & -7.4967 \\ 0 & -260.3390 & -32.5475 \\ 0 & -71.0887 & -334.5146 \end{bmatrix}, \quad B_2 = \begin{bmatrix} 1 & 131.8493 \\ 0 & 129.4535 \\ 0 & 246.0898 \end{bmatrix}
$$

$$
A_3 = \begin{bmatrix} -0.1846 & -18.2613 & -16.5938 \\ 0 & -212.7207 & -39.1326 \\ 0 & -57.2953 & -273.1203 \end{bmatrix}, \quad B_3 = \begin{bmatrix} 1 & 3.9603 \\ 0 & 125.0664 \\ 0 & 237.7500 \end{bmatrix}, \quad A_4 = \begin{bmatrix} -0.1892 & -5.9391 & -6.3613 \\ 0 & -223.2603 & -39.8070 \\ 0 & -59.4353 & -286.4680 \end{bmatrix}, \quad B_4 = \begin{bmatrix} 1 & 127.6290 \\ 0 & 125.2747 \\ 0 & 238.1458 \end{bmatrix}
$$

$$
A_5 = \begin{bmatrix} -0.0376 & -24.5611 & -22.8220 \\ 0 & -155.6979 & -9.7749 \\ 0 & -42.2200 & -199.9814 \end{bmatrix}, \quad B_5 = \begin{bmatrix} 1 & 51.7991 \\ 0 & 124.0234 \\ 0 & 235.7672 \end{bmatrix}, \quad A_6 = \begin{bmatrix} -0.0422 & -12.2389 & -12.5895 \\ 0 & -166.2375 & -10.4493 \\ 0 & -44.3601 & -213.3291 \end{bmatrix}, \quad B_6 = \begin{bmatrix} 1 & 175.4677 \\ 0 & 124.2317 \\ 0 & 236.1631 \end{bmatrix}
$$

$$
A_7 = \begin{bmatrix} -0.0823 & -23.3000 & -21.6865 \\ 0 & -118.6193 & -17.0343 \\ 0 & -30.5666 & -151.9348 \end{bmatrix}, \quad B_7 = \begin{bmatrix} 1 & 47.5787 \\ 0 & 119.8446 \\ 0 & 227.8233 \end{bmatrix}, \quad A_8 = \begin{bmatrix} -0.0869 & -10.9778 & -11.4540 \\ 0 & -129.1589 & -17.7087 \\ 0 & -32.7067 & -165.2825 \end{bmatrix}, \quad B_8 = \begin{bmatrix} 1 & 171.2474 \\ 0 & 120.0528 \\ 0 & 228.2191 \end{bmatrix}
$$

$$
A_9 = \begin{bmatrix} -0.0892 & 6.5001 & 7.0041 \\ 0 & -380.3930 & -23.4798 \\ 0 & -103.3946 & -488.6487 \end{bmatrix}, \quad B_9 = \begin{bmatrix} 1 & -117.8307 \\ 0 & 125.4415 \\ 0 & 238.4629 \end{bmatrix}, \quad A_{10} = \begin{bmatrix} -0.0938 & 18.8223 & 17.2366 \\ 0 & -390.9326 & -24.1542 \\ 0 & -105.5346 & -501.9964 \end{bmatrix}, \quad B_{10} = \begin{bmatrix} 1 & 5.8379 \\ 0 & 125.6497 \\ 0 & 238.8587 \end{bmatrix}
$$

$$
A_{11} = \begin{bmatrix} -0.1338 & 7.7613 & 8.1396 \\ 0 & -343.3144 & -30.7393 \\ 0 & -91.7412 & -440.6021 \end{bmatrix}, \quad B_{11} = \begin{bmatrix} 1 & -122.0511 \\ 0 & 121.2626 \\ 0 & 230.5189 \end{bmatrix}, \quad A_{12} = \begin{bmatrix} -0.1384 & 20.0834 & 18.3721 \\ 0 & -353.8540 & -31.4137 \\ 0 & -93.8812 & -453.9498 \end{bmatrix}, \quad B_{12} = \begin{bmatrix} 1 & 1.6175 \\ 0 & 121.4708 \\ 0 & 230.9147 \end{bmatrix}
$$

$$
A_{13} = \begin{bmatrix} 0.0132 & 1.4615 & 1.9113 \\ 0 & -286.2916 & -1.3816 \\ 0 & -76.6659 & -367.4631 \end{bmatrix}, \quad B_{13} = \begin{bmatrix} 1 & -74.2123 \\ 0 & 120.2196 \\ 0 & 228.5362 \end{bmatrix}, \quad A_{14} = \begin{bmatrix} 0.0085 & 13.7836 & 12.1438 \\ 0 & -296.8312 & -2.0560 \\ 0 & -78.8060 & -380.8109 \end{bmatrix}, \quad B_{14} = \begin{bmatrix} 1 & 49.4563 \\ 0 & 120.4278 \\ 0 & 228.9320 \end{bmatrix}
$$

$$
A_{15} = \begin{bmatrix} -0.0315 & 2.7226 & 3.0468 \\ 0 & -249.2130 & -8.6410 \\ 0 & -65.0125 & -319.4165 \end{bmatrix}, \quad B_{15} = \begin{bmatrix} 1 & -78.4327 \\ 0 & 116.0407 \\ 0 & 220.5922 \end{bmatrix}, \quad A_{16} = \begin{bmatrix} -0.0361 & 15.0448 & 13.2793 \\ 0 & -259.7526 & -9.3154 \\ 0 & -67.1526 & -332.7643 \end{bmatrix}, \quad B_{16} = \begin{bmatrix} 1 & 45.2359 \\ 0 & 116.2490 \\ 0 & 220.9881 \end{bmatrix}
$$

$$
A_{17} = \begin{bmatrix} -0.1455 & -6.3045 & -5.2586 \\ 0 & -59.0102 & -30.3175 \\ 0 & -17.8629 & -76.2852 \end{bmatrix}, \quad B_{17} = \begin{bmatrix} 1 & -46.6717 \\ 0 & 132.9620 \\ 0 & 252.7594 \end{bmatrix}, \quad A_{18} = \begin{bmatrix} -0.1501 & 6.0177 & 4.9739 \\ 0 & -69.5498 & -30.9919 \\ 0 & -20.0030 & -89.6330 \end{bmatrix}, \quad B_{18} = \begin{bmatrix} 1 & 76.9969 \\ 0 & 133.1703 \\ 0 & 253.1553 \end{bmatrix}
$$

$$
A_{19} = \begin{bmatrix} -0.1901 & -5.0433 & -4.1232 \\ 0 & -21.9316 & -37.5769 \\ 0 & -6.2095 & -28.2386 \end{bmatrix}, \quad B_{19} = \begin{bmatrix} 1 & -50.8921 \\ 0 & 128.7832 \\ 0 & 244.8154 \end{bmatrix}, \quad A_{20} = \begin{bmatrix} -0.1947 & 7.2788 & 6.1093 \\ 0 & -32.4712 & -38.2513 \\ 0 & -8.3496 & -41.5863 \end{bmatrix}, \quad B_{20} = \begin{bmatrix} 1 & 72.7766 \\ 0 & 128.9914 \\ 0 & 245.2113 \end{bmatrix}
$$

$$
A_{21} = \begin{bmatrix} -0.0431 & -11.3431 & -10.3514 \\ 0 & 35.0912 & -8.2193 \\ 0 & 8.8658 & 44.9003 \end{bmatrix}, \quad B_{21} = \begin{bmatrix} 1 & -3.0533 \\ 0 & 127.7402 \\ 0 & 242.8327 \end{bmatrix}, \quad A_{22} = \begin{bmatrix} -0.0477 & 0.9791 & -0.1189 \\ 0 & 24.5516 & -8.8937 \\ 0 & 6.7257 & 31.5526 \end{bmatrix}, \quad B_{22} = \begin{bmatrix} 1 & 120.6153 \\ 0 & 127.9484 \\ 0 & 243.2286 \end{bmatrix}
$$

$$
A_{23} = \begin{bmatrix} -0.0878 & -10.0820 & -9.2159 \\ 0 & 72.1699 & -15.4787 \\ 0 & 20.5192 & 92.9469 \end{bmatrix}, \quad B_{23} = \begin{bmatrix} 1 & -7.2737 \\ 0 & 123.5613 \\ 0 & 234.8887 \end{bmatrix}, \quad A_{24} = \begin{bmatrix} -0.0924 & 2.2402 & 1.0166 \\ 0 & 61.6303 & -16.1531 \\ 0 & 18.3791 & 79.5992 \end{bmatrix}, \quad B_{24} = \begin{bmatrix} 1 & 116.3949 \\ 0 & 123.7696 \\ 0 & 235.2846 \end{bmatrix}
$$

$$
A_{25} = \begin{bmatrix} -0.0947 & 19.7181 & 19.4747 \\ 0 & -189.6039 & -21.9242 \\ 0 & -52.3088 & -243.7670 \end{bmatrix}, \quad B_{25} = \begin{bmatrix} 1 & -172.6832 \\ 0 & 129.1582 \\ 0 & 245.5283 \end{bmatrix}, \quad A_{26} = \begin{bmatrix} -0.0993 & 32.0403 & 29.7072 \\ 0 & -200.1435 & -22.5986 \\ 0 & -54.4489 & -257.1147 \end{bmatrix}, \quad B_{26} = \begin{bmatrix} 1 & -49.0145 \\ 0 & 129.3664 \\ 0 & 245.9242 \end{bmatrix}
$$

$$
A_{27} = \begin{bmatrix} -0.1393 & 20.9792 & 20.6102 \\ 0 & -152.5252 & -29.1837 \\ 0 & -40.6554 & -195.7204 \end{bmatrix}, \quad B_{27} = \begin{bmatrix} 1 & -176.9035 \\ 0 & 124.9793 \\ 0 & 237.5844 \end{bmatrix}, \quad A_{28} = \begin{bmatrix} -0.1439 & 33.3014 & 30.8427 \\ 0 & -163.0648 & -29.8581 \\ 0 & -42.7955 & -209.0681 \end{bmatrix}, \quad B_{28} = \begin{bmatrix} 1 & -53.2349 \\ 0 & 125.1876 \\ 0 & 237.9802 \end{bmatrix}
$$

$$
A_{29} = \begin{bmatrix} 0.0077 & 14.6794 & 14.3819 \\ 0 & -95.5024 & 0.1740 \\ 0 & -25.5801 & -122.5815 \end{bmatrix}, \quad B_{29} = \begin{bmatrix} 1 & -129.0648 \\ 0 & 123.9363 \\ 0 & 235.6017 \end{bmatrix}, \quad A_{30} = \begin{bmatrix} 0.0030 & 27.0016 & 24.6144 \\ 0 & -106.0420 & -0.5004 \\ 0 & -27.7202 & -135.9292 \end{bmatrix}, \quad B_{30} = \begin{bmatrix} 1 & -5.3961 \\ 0 & 124.1446 \\ 0 & 235.9975 \end{bmatrix}
$$

$$
A_{31} = \begin{bmatrix} -0.0370 & 15.9406 & 15.5174 \\ 0 & -58.4238 & -7.0854 \\ 0 & -13.9267 & -74.5348 \end{bmatrix}, \quad B_{31} = \begin{bmatrix} 1 & -133.2851 \\ 0 & 119.7575 \\ 0 & 227.6577 \end{bmatrix}, \quad A_{32} = \begin{bmatrix} -0.0416 & 28.2627 & 25.7499 \\ 0 & -68.9634 & -7.7598 \\ 0 & -16.0668 & -87.8826 \end{bmatrix}, \quad B_{32} = \begin{bmatrix} 1 & -9.6165 \\ 0 & 119.9657 \\ 0 & 228.0535 \end{bmatrix}
$$

(4.26b)

Chapter 5

Robust LPV Corrective Controller

The complete dynamic bicycle model presented in (2.26) can be expressed in DT as a non-linear difference equation similar to the one shown in (5.1).

$$x[k+1] = f\left(x[k], u[k]\right) + d[k] \tag{5.1}$$

Where $f\left(x[k], u[k]\right)$ would be the non-linear difference equation obtain from the discretization of the simplified dynamic bicycle model presented in (2.28) and $d[k]$ would represent a vector containing all the unmodelled dynamics of the real vehicle and/or exogenous disturbances acting over it, such as the wind forces or the angle of inclination of the road.

Chapter 3 presented the LPV model of the simplified dynamic bicycle model, such that the non-linear model presented in (5.1), could be expressed in LPV form as shown in (5.2).

$$x[k+1] = A(\zeta)x[k] + B(\zeta)u[k] + d[k] \tag{5.2}$$

However, since the disturbance vector $d[k]$ cannot be predicted based on the assumptions that have been made, the MPC controller designed to safely coordinate the vehicles must work with the nominal system presented in (5.3), which does not take into account any of the disturbances.

$$\tilde{x}[k+1] = A(\zeta)\tilde{x}[k] + B(\zeta)\tilde{u}[k] \tag{5.3}$$

As a result, a mismatch error between the states of the real vehicle system (5.2) and the states of the disturbance-free nominal system (5.3) will occur. This mismatch error can be computed as shown in (5.4).

$$e[k] = x[k] - \tilde{x}[k] \tag{5.4}$$

In order to minimize this mismatch error, a local corrective controller must be implemented, such as the control law proposed in (5.5).

$$u[k] = \tilde{u} + K(\zeta)e[k] \tag{5.5}$$

Where $K(\zeta)$ is a gain-schedule polytopic LPV controller, as presented in (5.6).

$$K(\zeta) = \sum_{i=1}^{N} \mu_i(\theta) K_i \tag{5.6}$$

This LPV controller should be designed such that the dynamic behaviour of the closed-loop mismatch error system, presented in (5.7), is asymptotically stabilize.

$$e[k+1] = x[k+1] - \tilde{x}[k+1] = \big(A(\zeta) + B(\zeta)K(\zeta)\big)\, e[k] + d[k] \tag{5.7}$$

To better illustrate the control scheme, a diagram of the local corrective controller is presented in Figure 5.1, in which the connection between the different systems is shown.

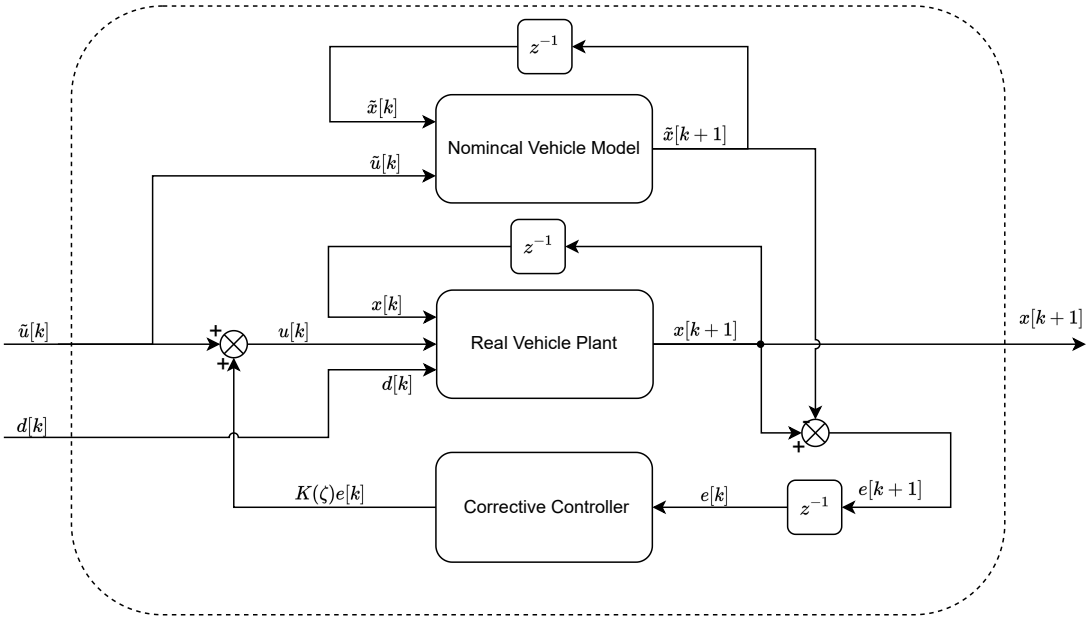


Figure 5.1: Diagram of the stabilized vehicle using a robust LPV controller.

The proposed LPV controller will be synthesized by applying H_∞-optimal control, as it has been proven to be more efficient when dealing with systems subject to external disturbances at rejecting system variations than linear quadratic regulator (LQR)-optimal control strategies [19].

5.1 Theoretical Background on $\mathbf{H}_\infty$-Optimal Control Theory

This section will be focused on presenting the LMI used to synthesize H_∞-optimal controllers for linear time invariant (LTI) and LPV DT systems.

5.1.1 $\mathbf{H}_\infty$-Optimal Control Design for LTI Systems

Optimization problems solved by the H_∞-optimal control theory are expressed using the generalized plant P, also known as the standard control problem, as many LTI systems can be expressed in this form. A diagram of a generalized plant is presented in Figure 5.2.

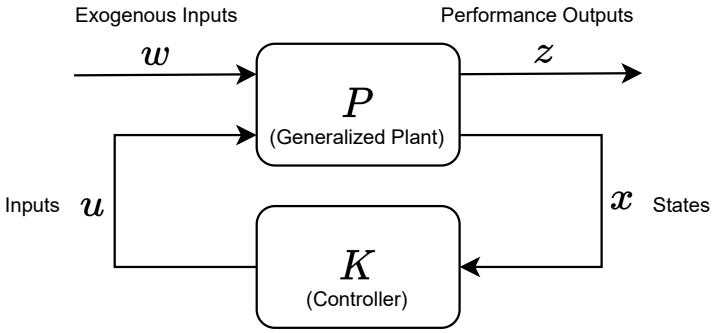


Figure 5.2: General control configuration used for H_∞-optimal control.

The discrete-time generalized plant P can be described by the SS plant shown in (5.8). Where x are the system's states, u are the system's inputs, w are the exogenous inputs and z are the performance outputs.

$$\begin{cases} x[k+1] = A^d x[k] + B_u^d u[k] + B_w^d w[k] & \text{(5.8a)} \\ z[k] = Cx[k] + D_u u[k] + D_w w[k] & \text{(5.8b)} \end{cases}$$

The goal of H_∞-optimal control is to design a feedback controller K that minimizes the H_∞-norm of the closed-loop transfer matrix from w to z, as presented in (5.9).

$$\begin{aligned} \min_{K} \quad & \gamma \\ \text{s.t.} \quad & \|G_{w\to z}\|_\infty < \gamma \end{aligned} \tag{5.9}$$

Where the closed-loop transfer matrix from w to z can be computed by first substituting $u[k] = Kx[k]$ in (5.8), as presented in (5.10).

$$\begin{cases} x[k+1] = \left(A^d + B_u^d K\right) x[k] + B_w^d w[k] & \text{(5.10a)} \\ z[k] = (C + D_u K)\, x[k] + D_w w[k] & \text{(5.10b)} \end{cases}$$

And then converting the SS representation shown in (5.10) into its equivalent DT transfer function representation, shown in (5.11).

$$G_{w\to z} = (C + D_u K)\left(zI - \left(A^d + B_u^d K\right)\right)^{-1} B_w^d + D_w \tag{5.11}$$

Thus, the DT H_∞-optimal feedback controller K can be synthesized by solving the LMI optimization problem shown in (5.8) [26].

$$\begin{aligned} \min_{X,F,\gamma} \quad & \gamma \\ \text{s.t.} \quad & X > 0 \\ & \begin{bmatrix} X & A^dX + B^dF & B_w^d & 0 \\ * & X & 0 & XC^\intercal + F^\intercal D_u^\intercal \\ * & * & \gamma I & D_w^\intercal \\ * & * & * & \gamma I \end{bmatrix} > 0 \\ & X \in \mathbb{S}^{n_x} \\ & \gamma \in \mathbb{R} \\ & F \in \mathbb{R}^{n_x \times n_u} \end{aligned} \tag{5.12}$$

Where $P = X^{-1}$ is the Lyapunov matrix and F is a matrix used to recover the DT H_∞-optimal gain as presented in (5.13).

$$K = FX^{-1} \tag{5.13}$$

5.1.2 H_∞-Optimal Control Design for LPV Systems

The previous chapter presented a LMI optimization problem capable of synthesizing a DT H_∞-optimal feedback controller for LTI systems, such as (5.8).

However, this optimization problem is not directly applicable for LPV systems, such as the one presented in (5.14), since the LMI constraint should have to be evaluated under all the possible numerical values of $A^d(\zeta)$ and $B^d(\zeta)$.

$$\begin{cases} x[k+1] = A^d(\zeta)x[k] + B_u^d(\zeta)u[k] + B_w^d w[k] & \text{(5.14a)} \\ z[k] = Cx[k] + D_u u[k] + D_w w[k] & \text{(5.14b)} \end{cases}$$

Thus, in order to compute a LPV DT H_∞-optimal feedback controller $K(\zeta)$, the LMI constraint must be evaluated in each vertex of the DT polytopic LPV model, as shown in (5.15).

$$
\begin{aligned}
\min_{X,F_i,\gamma} \quad & \gamma \\
\text{s.t.} \quad & X > 0 \\
& \begin{bmatrix} X & A_i^d X + B_i^d F_i & B_w^d & 0 \\ * & X & 0 & XC^\mathsf{T} + F_i^\mathsf{T} D_u^\mathsf{T} \\ * & * & \gamma I & D_w^\mathsf{T} \\ * & * & * & \gamma I \end{bmatrix} > 0, \quad \forall i = 1, \dots, N \\
& X \in \mathbb{S}^{n_x} \\
& F_i \in \mathbb{R}^{n_x \times n_u}, \quad \forall i = 1, \dots, N \\
& \gamma \in \mathbb{R}
\end{aligned}
\tag{5.15}
$$

Where the matrices (A_i^d, B_i^d) correspond to the discretized matrices (A_i, B_i) of the DT polytopic LPV model, $P = X^{-1}$ is the common Lyapunov matrix of the LPV system and F_i is a matrix used to recover each vertex of the gain-schedule polytopic LPV controller, as presented in (5.16).

$$K_i = F_i X^{-1} \tag{5.16}$$

5.2 $\mathbf{H}_\infty$-LPV Corrective Controller for the Bicycle Dynamic Model

To compute an optimal H_∞-LPV corrective controller, the disturbance and the performance vectors, z and d, must be appropriately defined in order to achieve the desired corrective behaviour.

A common strategy to define the performance vector is to compute them as the weighted control input and weighted state error, as presented in (5.17).

$$z_1[k] = W_{v_x} \left(v_{x_{ref}}[k] - v_x[k] \right) \tag{5.17a}$$
$$z_2[k] = W_{v_y} \left(v_{y_{ref}}[k] - v_y[k] \right) \tag{5.17b}$$
$$z_3[k] = W_\omega \left(\omega_{ref}[k] - \omega[k] \right) \tag{5.17c}$$
$$z_4[k] = W_a a[k] \tag{5.17d}$$
$$z_5[k] = W_\delta \delta[k] \tag{5.17e}$$

Where $v_{x_{ref}}$, $v_{y_{ref}}$ and ω_{ref} are the velocity references, and $F_i \in \mathbb{R}^+$ are the weights used to normalize each performance output.

Regarding the disturbance vector $d[k]$, it can be defined as a three variable vector affecting each velocity independently, as presented in (5.18).

$$d[k] = \left[d_{v_x}[k], d_{v_y}[k], d_\omega[k] \right]^\mathsf{T} \tag{5.18}$$

From these two definitions, the exogenous inputs $w[k]$ can be defined as in (5.19).

$$w[k] = \left[d_{v_x}[k], d_{v_y}[k], d_{\omega}[k], v_{x_{ref}}[k], v_{y_{ref}}[k], \omega_{ref}[k]\right]^{\mathsf{T}} \tag{5.19}$$

Such that the matrices B_w^d, C, D_u and D_w can be derived, as shown in (5.20).

$$\begin{aligned} B_w^d &= \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix}, \quad C = \begin{bmatrix} -W_{v_x} & 0 & 0 \\ 0 & -W_{v_y} & 0 \\ 0 & 0 & -W_{\omega} \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \\ D_u &= \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ W_a & 0 \\ 0 & W_{\delta} \end{bmatrix}, \qquad D_w = \begin{bmatrix} W_{v_x} & 0 & 0 & 0 & 0 & 0 \\ 0 & W_{v_y} & 0 & 0 & 0 & 0 \\ 0 & 0 & W_{\omega} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \end{aligned} \tag{5.20}$$

Finally, having defined the previous vectors and matrices, the optimization problem presented in (5.15) was optimally solved using the matrices (A_i, B_i) of the polytopic LPV model, shown in (4.26), and the weights presented in (5.21). The solver used for the optimization problem was Mosek [27], which was configured using the YALMIP framework [28].

$$W_{v_x} = \frac{0.4363}{v_{x_{max}}}, \; W_{v_y} = \frac{0.2285}{v_{y_{max}}}, \; W_{\omega} = \frac{0.1454}{\omega_{max}}, \; W_a = \frac{0.1891}{a_{max}} \; W_{\delta} = \frac{0.0007}{\delta_{max}} \tag{5.21}$$

The resulting attenuation of the H_∞-norm of the closed-loop transfer matrix $G_{w \to z}$ is presented in (5.22).

$$\gamma = 1815.298 \tag{5.22}$$

The common Lyapunov matrix is presented in (5.23).

$$P = \begin{bmatrix} 0.3025 & -0.0459 & -0.0037 \\ -0.0459 & 6.0813 & -4.8026 \\ -0.0037 & -4.8026 & 4.1140 \end{bmatrix} > 0 \tag{5.23}$$

Whereas the vertices of the polytopic H_∞-LPV corrective controller are presented in (5.24).

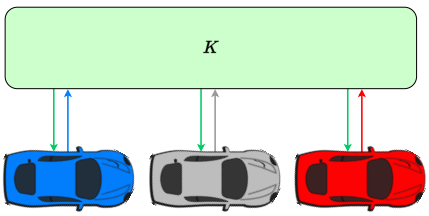
$$K(\zeta) = \sum_{i=1}^{32} \mu_i(\theta) K_i \tag{5.24a}$$

$$
\begin{aligned}
K_1 &= \begin{bmatrix} -59.8026 & 8.1119 & 2.2422 \\ 0.0268 & 1.0382 & -0.0939 \end{bmatrix}, & K_2 &= \begin{bmatrix} -56.2344 & -16.3128 & 0.4590 \\ -0.1661 & 0.9194 & 0.0172 \end{bmatrix} \\
K_3 &= \begin{bmatrix} -59.7404 & 10.5713 & 0.1289 \\ 0.0339 & 1.2991 & -0.5905 \end{bmatrix}, & K_4 &= \begin{bmatrix} -56.2725 & -19.6580 & 9.2467 \\ -0.1708 & 1.1669 & -0.4485 \end{bmatrix} \\
K_5 &= \begin{bmatrix} -59.6847 & -0.3965 & 9.8619 \\ -0.0512 & 1.7042 & -0.9790 \end{bmatrix}, & K_6 &= \begin{bmatrix} -52.7904 & -35.9001 & 24.6460 \\ -0.2365 & 1.4534 & -0.7557 \end{bmatrix} \\
K_7 &= \begin{bmatrix} -59.7194 & 0.1283 & 11.5797 \\ -0.0486 & 2.0039 & -1.5295 \end{bmatrix}, & K_8 &= \begin{bmatrix} -52.6492 & -41.3503 & 37.2701 \\ -0.2462 & 1.7154 & -1.2353 \end{bmatrix} \\
K_9 &= \begin{bmatrix} -55.4040 & 5.2595 & 27.9716 \\ 0.2152 & 0.2181 & 1.3162 \end{bmatrix}, & K_{10} &= \begin{bmatrix} -60.4384 & -6.3211 & 1.3055 \\ 0.0352 & 0.1406 & 1.5762 \end{bmatrix} \\
K_{11} &= \begin{bmatrix} -53.0890 & 8.6245 & 21.8098 \\ 0.2352 & 0.4172 & 0.9078 \end{bmatrix}, & K_{12} &= \begin{bmatrix} -60.2333 & -4.2215 & -0.0334 \\ 0.0438 & 0.3783 & 1.1244 \end{bmatrix} \\
K_{13} &= \begin{bmatrix} -56.5246 & 14.8602 & 9.9017 \\ 0.1709 & 0.8127 & 0.5832 \end{bmatrix}, & K_{14} &= \begin{bmatrix} -60.1847 & -5.9420 & -5.1009 \\ -0.0373 & 0.7406 & 0.7223 \end{bmatrix} \\
K_{15} &= \begin{bmatrix} -56.0764 & 20.9050 & 0.9351 \\ 0.1884 & 1.0620 & 0.1004 \end{bmatrix}, & K_{16} &= \begin{bmatrix} -60.6647 & -5.0566 & -3.4702 \\ -0.0629 & 1.0626 & 0.2234 \end{bmatrix} \\
K_{17} &= \begin{bmatrix} -58.4475 & 36.5288 & -26.8768 \\ 0.1052 & 2.1357 & -2.0860 \end{bmatrix}, & K_{18} &= \begin{bmatrix} -59.1245 & -14.9753 & 18.6779 \\ -0.0809 & 2.1092 & -2.0014 \end{bmatrix} \\
K_{19} &= \begin{bmatrix} -58.2520 & 42.4815 & -35.7753 \\ 0.1168 & 2.4078 & -2.6105 \end{bmatrix}, & K_{20} &= \begin{bmatrix} -59.1733 & -15.4340 & 22.0784 \\ -0.0807 & 2.3989 & -2.5420 \end{bmatrix} \\
K_{21} &= \begin{bmatrix} -62.0529 & 17.7156 & -4.1667 \\ 0.0435 & 2.8673 & -3.0759 \end{bmatrix}, & K_{22} &= \begin{bmatrix} -57.2793 & -40.7242 & 53.3373 \\ -0.1543 & 2.6892 & -2.8160 \end{bmatrix} \\
K_{23} &= \begin{bmatrix} -59.7802 & 27.2625 & -14.4515 \\ 0.0634 & 3.2423 & -3.7286 \end{bmatrix}, & K_{24} &= \begin{bmatrix} -60.5589 & -49.6574 & 66.2920 \\ -0.1270 & 3.0484 & -3.4256 \end{bmatrix} \\
K_{25} &= \begin{bmatrix} -49.7579 & 43.5823 & -24.3954 \\ 0.2627 & 1.1868 & -0.5413 \end{bmatrix}, & K_{26} &= \begin{bmatrix} -58.3122 & 14.5575 & -13.2440 \\ 0.1172 & 1.3206 & -0.5377 \end{bmatrix} \\
K_{27} &= \begin{bmatrix} -48.8968 & 52.6749 & -40.3918 \\ 0.2800 & 1.3874 & -0.9509 \end{bmatrix}, & K_{28} &= \begin{bmatrix} -58.0574 & 19.8082 & -20.8282 \\ 0.1306 & 1.5791 & -1.0301 \end{bmatrix} \\
K_{29} &= \begin{bmatrix} -53.0135 & 56.1909 & -42.4744 \\ 0.2348 & 1.8441 & -1.3979 \end{bmatrix}, & K_{30} &= \begin{bmatrix} -60.6768 & 13.4131 & -12.9358 \\ 0.0424 & 2.0240 & -1.4737 \end{bmatrix} \\
K_{31} &= \begin{bmatrix} -52.3325 & 65.2017 & -57.5873 \\ 0.2537 & 2.0866 & -1.8746 \end{bmatrix}, & K_{32} &= \begin{bmatrix} -60.8157 & 17.7883 & -17.9848 \\ 0.0514 & 2.3294 & -2.0366 \end{bmatrix}
\end{aligned}
\tag{5.24b}
$$

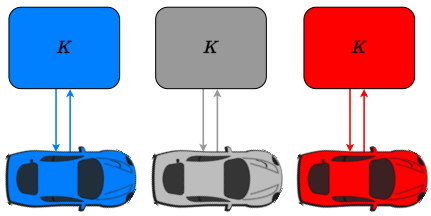
Chapter 6

Autonomous Vehicle Coordination

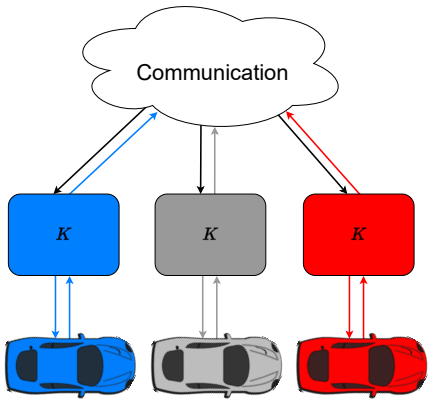
The coordinated control of autonomous vehicles can be divided into three architectures [29]: centralized control, decentralized control, and distributed control, as shown in Figure 6.1.



(a) Centralized control.



(b) Decentralized Control.



(c) Distributed Control.

Figure 6.1: Different coordinated control architectures.

The characteristics, as well as the advantages and disadvantages of each architecture are presented below:

- *Centralized control*. In this approach, a central control unit gathers all the information from the autonomous vehicles, makes decisions using this in-

formation, and then control each autonomous vehicle. An advantage of this architecture is that the controller can compute a globally optimal solution since all the decision making is made at the same location and time. However, a disadvantage of this architecture is that the required computational power and communication bandwidth increases with the number of autonomous vehicles. Another drawback is that they are vulnerable to a single point of failure (SPOF) since a fault in the decision making or in the communication could result in a catastrophic failure of the whole system.

- *Decentralized control*. In this approach, each autonomous vehicles makes its own decisions based on the information provided by the local sensors. As a result, it can only achieve locally optimal solutions. However, the advantage of this architecture is that it is scalable, modular and tolerant to individual failures, as it lacks any SPOF.

- *Distributed control*. In this architecture, a local controller is implemented in each autonomous vehicle (or groups of autonomous vehicles), but there exists information exchange between the different control units. This approach does not need a central control unit like in the centralized architecture, at the expense of requiring a far more complex communication and decision process structure. This architecture has been shown to be more promising in autonomous vehicles that present limited computational power and short-range communication capacities [30].

In this project, a distributed coordination control system is proposed, by which an autonomous vehicle is capable of maneuvering through a road while avoiding other vehicles. This is achieved by using the information broadcasted by the other vehicles in order to compute the lateral bounds of a trajectory that avoids any possible collision.

6.1 Lateral Bounds Computation Using Distributed Vehicle Coordination

The vectors of lateral bounds, $\underline{y}_e^{vect}$ and $\overline{y}_e^{vect}$, of the controlled vehicle during a prediction horizon of H_p steps are computed based on the following information, received from the surrounding vehicles:

- s^{vect}: A vector containing the predicted s coordinates during a prediction horizon of H_p steps.

- y_e^{vect}: A vector containing the predicted lateral error y_e with respect to the centre of the road during a prediction horizon of H_p steps.

Using this information, the controlled vehicle can predict future collisions and avoid them by modifying its future lateral bounds, as proposed in Algorithm 1. This algorithm takes respectively as input the vector of predicted s coordinates of the controlled vehicle, s_0^{vect}, and the obstacle vehicle, s_1^{vect}; the vector of pre-

dicted lateral error of the obstacle vehicle, $y_{e_1}^{vect}$; the length and width of the vehicle, $L_{vehicle}$ and $W_{vehicle}$; the width of the road, W_{road}, and the number of steps of the prediction horizon, H_p. As output, it computes the vectors of lateral bounds, $\underline{y}_e^{vect}$ and $\overline{y}_e^{vect}$.

The algorithm works by looping though the prediction horizon in inverse order, i.e. form t_{k+H_p} to t_{k+1}, while searching for possible collisions. A collision at step t_k is indicated by a distance between the predicted $s_0^{vect}[k]$ and $s_1^{vect}[k]$ smaller than a predefined collision threshold, which could be defined as the length of the largest vehicle plus a security margin. If a collision is predicted, then the lateral bounds are reduced in order to avoid the collision by either turning to the right or to the left lane. Finally, the auxiliary variables Δ^+ and Δ^- are used to gradually reduce the lateral bounds between t_1 and t_k, instead of a discontinuous jump at t_k.

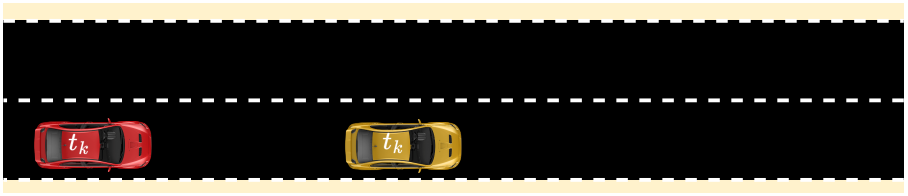
Algorithm 1 Lateral bounds computation with one obstacle vehicle.

Input: $s_0^{vect}, s_1^{vect}, y_{e_1}^{vect}, L_{vehicle}, W_{vehicle}, W_{road}, H_p$
Output: $\underline{y}_e^{vect}, \overline{y}_e^{vect}$

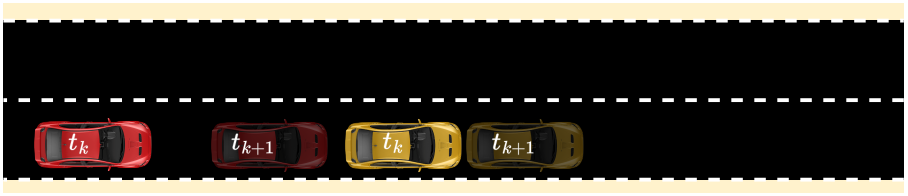
$\Delta^+ \leftarrow 0$
$\Delta^- \leftarrow 0$
$threshold \leftarrow L_{vehicle}$
$margin \leftarrow W_{vehicle}/2$
$Collision \leftarrow$ `False`
for $i = H_p, H_p - 1, \ldots, 2, 1$ **do**
  if distance from $s_0^{vect}(i)$ to $s_1^{vect}(i) < threshold$ **then** ▷ Collision?
    $Collision \leftarrow$ `True`
    if $y_{e_1}^{vect}(i) > 0$ **then** ▷ Obstacle on left lane?
      $\overline{y}_e^{vect}(i) \leftarrow \frac{W_{road}}{2} - \left(y_{e_1}^{vect}(i) + margin\right)$
      $\Delta^+ \leftarrow \left(\frac{W_{road}}{2} - \overline{y}_e^{vect}(i)\right)/(i-1)$
    else ▷ Obstacle on right lane?
      $\underline{y}_e^{vect}(i) \leftarrow -\left(\frac{W_{road}}{2} - \left(y_{e_1}^{vect}(i) + margin\right)\right)$
      $\Delta^- \leftarrow \left(\frac{W_{road}}{2} - \overline{y}_e^{vect}(i)\right)/(i-1)$
    end if
  else if $Collision =$ `True` **then** ▷ Previous collision?
    $\overline{y}_e^{vect}(i) \leftarrow \overline{y}_e^{vect}(i) + \Delta^+$
    $\underline{y}_e^{vect}(i) \leftarrow \underline{y}_e^{vect}(i) - \Delta^-$
  else ▷ No collision
    $\overline{y}_e^{vect}(i) \leftarrow \frac{W_{road}}{2}$
    $\underline{y}_e^{vect}(i) \leftarrow -\frac{W_{road}}{2}$
  end if
end for

Figure 6.2 presents a series of scenarios that illustrate how the lateral bound of the red vehicle are computed using Algorithm 1. The first scenario shown in Figure 6.2a presents the initial position of the controlled vehicle (in red) and the

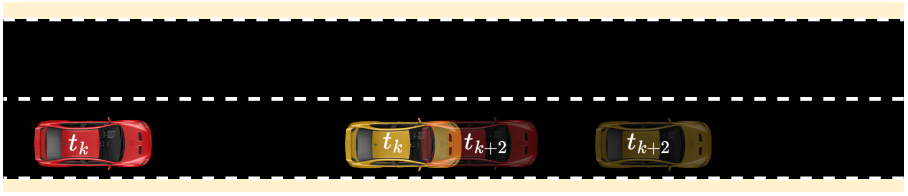
obstacle vehicle (in yellow) at t_k. The second and third scenarios shown in Figures 6.2b and 6.2c present the predicted position of each vehicle at t_{k+1} and t_{k+2} in which no collisions are detected. Finally, the fourth scenario shown in Figure 6.2d predicts a collision between the vehicles at t_{k+3}. As such, the vector of lateral bounds is computed so that the red vehicle avoids the yellow vehicle, as presented in the final scenario shown in Figure 6.2e. Also note that the lateral bound gradually decrease thanks to the use of the auxiliary variables Δ^+ and Δ^-.



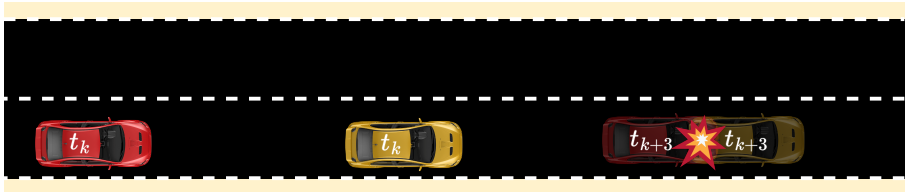
(a) Initial position of each vehicle at t_k.



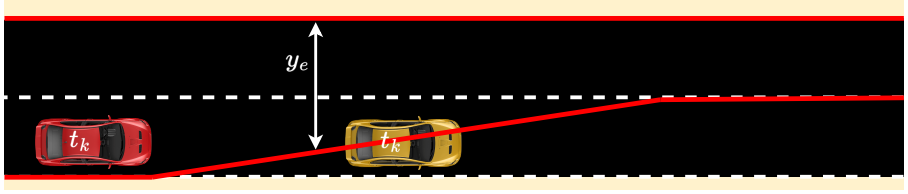
(b) Predicted position of each vehicle at t_{k+1} (without collision).



(c) Predicted position of each vehicle at t_{k+2} (without collision).



(d) Predicted collision on right lane at t_{k+3}.



(e) Lateral bounds based on the predicted collision at t_{k+3}.

Figure 6.2: Computation of the lateral bounds based on vehicle coordination.

Unfortunately, the lateral bounds computed using Algorithm 1 are only valid when only one obstacle vehicle is present. For scenarios with multiple obstacle

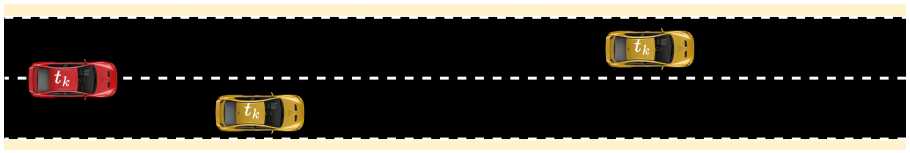
vehicles occupying the road, the procedure presented in Algorithm 2 has been designed. This new algorithm computes the individual bounds generated by each obstacle vehicle using Algorithm 1, and then loops through each step of the prediction horizon to select the most restrictive constraint.

Algorithm 2 Lateral bounds computation with N obstacle vehicle.

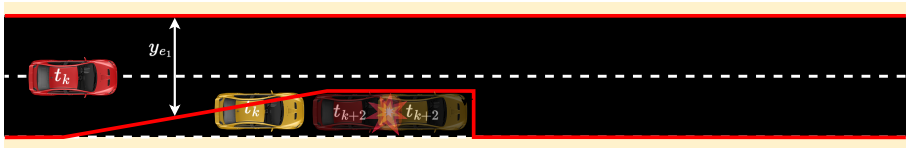
Input: $s_0^{vect}, [s_1^{vect}, \ldots, s_N^{vect}], [y_{e_1}^{vect}, \ldots, y_{e_N}^{vect}], L_{vehicle}, W_{vehicle}, W_{road}, H_p$
Output: $\underline{y}_e^{vect}, \overline{y}_e^{vect}$

1: **for** $i = 1, \ldots, N$ **do** ▷ Compute bounds for each obstacle
2: $\quad [\underline{y}_{e_i}^{vect}, \overline{y}_{e_i}^{vect}] \leftarrow f(s_0^{vect}, s_i^{vect}, y_{e_i}^{vect}, L_{vehicle}, W_{vehicle}, W_{road}, H_p)$ ▷ Algorithm 1
3: **end for**
4: **for** $j = 1, \ldots, H_p$ **do** ▷ Select most restrictive constraints
5: $\quad \overline{y}_e^{vect}(j) \leftarrow \min_i \overline{y}_{e_i}^{vect}(j)$
6: $\quad \underline{y}_e^{vect}(j) \leftarrow \max_i \underline{y}_{e_i}^{vect}(j)$
7: **end for**

Finally, Figure 6.3 presents a series of scenarios that illustrate how the lateral bound of the red vehicle are computed using Algorithm 2. The first scenario shown in Figure 6.3a presents the initial position of the controlled vehicle (in red) and the obstacle vehicles (in yellow) at t_k. The second and third scenario shown in Figures 6.3b and 6.3c present the individual lateral bounds generated by the predicted collision with each obstacle vehicle at t_{k+2} and t_{k+5}, respectively. Finally, the final scenario shown in Figure 6.3d presents the resulting lateral bounds obtained from selecting the most restrictive bounds.



(a) Initial position of each vehicle at t_k.



(b) Predicted collision on right lane at t_{k+2}.

(c) Predicted collision on left lane at t_{k+5}.

(d) Resulting lateral bounds.

Figure 6.3: Computation of the lateral bounds with multiple vehicle coordination.

Chapter 7

Robust Tube-Based MPC Controller for Autonomous Driving

In this chapter, a MPC controller is designed to maneuver an autonomous vehicle through a road, whilst avoiding other moving vehicles. The proposed controller uses the LPV models presented in Chapter 3, to reduce the computational burden of the optimization problem by transforming the non-linear optimization problem into a quadratic one.

The path generated by the controller avoids collision by constraining the lateral displacement of the vehicle to the bounds computed by the vehicle coordination algorithm presented in Chapter 6.

Meanwhile, the robust feasibility of the problem is ensured by implementing a tube-based model predictive control (T-MPC), in which the state and input constraints are adapted using the reachable sets (also known as tubes) of the mismatch error, presented in Chapter 5, between the real dynamics of the vehicle and the nominal model. The reachable sets are represented by zonotopes, which are a special type of polytope that present efficient computational properties, that improve the controller's speed and accuracy. Further information about zonotopes can be found in Appendix A.

7.1 Reachable Set Computation Using Zonotopes

The main idea behind reachable sets is to bound, at every step time t_k, the maximum achievable values of a DT system, as the one presented in (7.1), given a set of initial states $\psi[0] \in \Psi_0$.

$$\psi[k+1] = A^d\psi[k] + B^d u[k] + B_w^d w[k] \tag{7.1}$$

Such that each state is bounded by the reachable set: $\psi[k] \in \Psi_k$, for $k = 1, \ldots, H_p$, being H_p the prediction horizon of the reachable set. To compute each bounding set Ψ_{k+1}, the one-step reachable set from Ψ_k is defined as in (7.2).

$$\Psi_{k+1} = \left\{ A^d \psi[k] + B^d u[k] + B_w^d w[k] \;\middle|\; \psi[k] \in \Psi_k, u[k] \in U, w[k] \in W \right\} \tag{7.2}$$

Where Ψ_k is the reachable set at t_k, and U and W are sets bounding the inputs and disturbances. If the sets Ψ_k, U and W are expressed using zonotopes, then the one-step reachable set can be efficiently computed using the zonotopic operation shown in (7.3) [19].

$$\Psi_{k+1} = \left(A^d \otimes \Psi_k \right) \oplus \left(B^d \otimes U \right) \oplus \left(B_w^d \otimes W \right) \tag{7.3}$$

7.1.1 Reachable Sets of the Mismatch Error

In this section, the reachable sets of the mismatch error between the real and nominal states of the COM will be computed using the Equations presented in Section 7.1.

First, the DT system shown in (7.1) must be defined for the dynamics of the mismatch error. These dynamics had been previously defined in Chapter 5, from which the mismatch error of the COM can be computed using (7.4).

$$e[k+1] = \begin{bmatrix} e_{v_x}[k+1] \\ e_{v_y}[k+1] \\ e_{\omega}[k+1] \\ e_{y_e}[k+1] \\ e_{\theta_e}[k+1] \\ e_s[k+1] \end{bmatrix} = A_{CL}(\zeta[k])e[k] + w[k] \tag{7.4}$$

Where $w[k]$ are the disturbances affecting the system and A_{CL} is the state matrix of the close-loop system presented in (7.5).

$$A_{CL}(\zeta[k]) = A(\zeta[k]) + B(\zeta[k])K^{\star}(\zeta[k]) \tag{7.5}$$

In which $A(\zeta[k])$ and $B(\zeta[k])$ are the LPV matrices obtained from the COM presented in (3.11), and $K^{\star}(\zeta[k])$ is the state feedback gain shown in (7.6).

$$K^{\star}(\zeta) = \left[K(\zeta[k]), 0, 0, 0 \right] \tag{7.6}$$

Where $K(\zeta[k])$ is the polytopic H_∞-LPV corrective controller presented in (5.24). Note that the state feedback gain is only multiplying the dynamic states of the bicycle model, which implies that the mismatch error over the prediction horizon will be smaller for the dynamic states than for the kinematic ones.

If the disturbances can be bounded by a zonotope, $w[k] \in W$, then the zonotopic one-step reachable set defined in (7.3), can be computed for the mismatch error system presented in (7.4), as shown in (7.7).

$$E_{k+1} = A_{CL}(\zeta_k) \otimes E_k \oplus W, \quad \forall k = 0, \dots, H_p - 1 \tag{7.7}$$

Where the initial reachable set is set to zero, $E_0 = \langle 0, 0 \rangle$, since the initial mismatch error is null. This is due to the measurement of the real states at t_0, which thereby reset the values of the nominal states: $\tilde{x}_0 = x_0, \Rightarrow e_0 = 0$.

Note that to compute the reachable sets, the scheduling variables $\zeta = [v_x, v_y, y_e, \theta_e, \kappa, \delta]$ must be estimated over the prediction horizon of H_p steps, in order to estimate the matrix $A_{CL}(\zeta_k)$ and the feedback gain $K(\zeta_k)$. This prediction can be performed using the estimated states and inputs of the MPC optimization problem.

Finally, these reachable sets will used to compute a window of robust state and input constraints that will be used in the MPC controller, as presented in the following section.

7.2 Robust Constraint Computation Using Reachable Sets

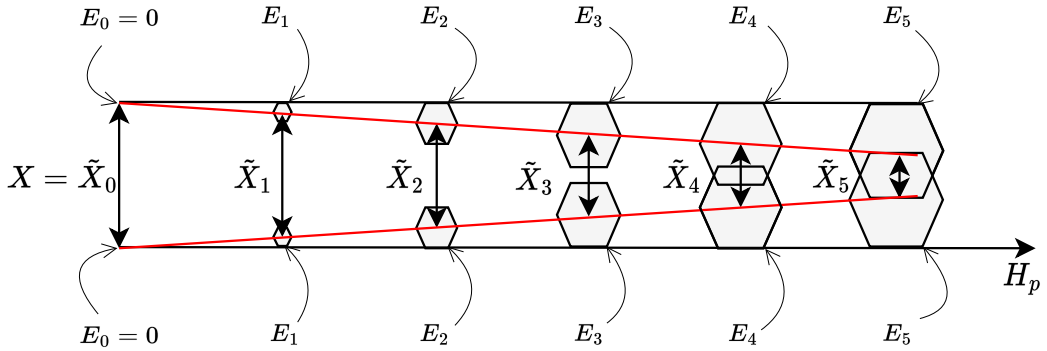
In this section, the constraints of the states and inputs of the controlled vehicle will be tightened to ensure the robust feasibility of the MPC controller, as presented in [19]. To do so, the new input and state constraints sets will be updated at every control iteration, as presented in (7.8).

$$\tilde{X}_{k+i} = X_{k+i} \ominus E_{k+i}, \qquad \forall i = 1, \dots, H_p \tag{7.8a}$$

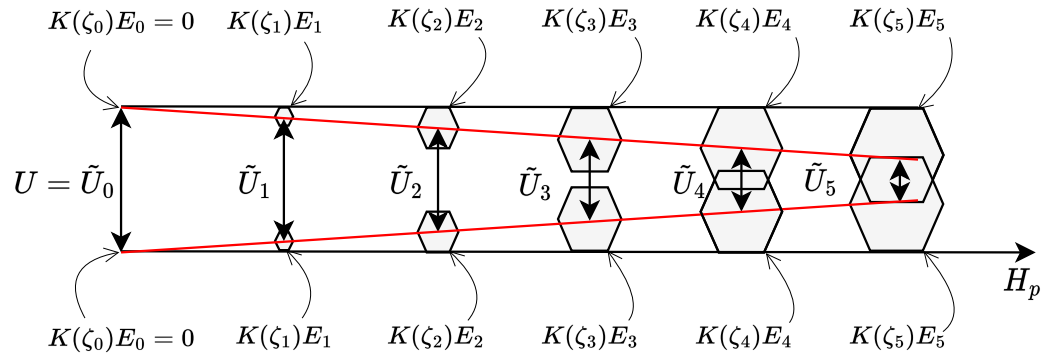
$$\tilde{U}_{k+i} = U \ominus K_{\zeta_{k+i}} E_{k+i}, \qquad \forall i = 0, \dots, H_p - 1 \tag{7.8b}$$

Where U and X_{k+i} are the original state and input constraints and $\tilde{X}_{k+i}$ and $\tilde{U}_{k+i}$ are the tightened constraints that ensure robust feasibility. Note that the original set of state constraints X_{k+i} is time dependent due to the time dependency of the lateral displacement of the vehicle $y_{e_{k+i}}$, which ensured a collision-free path, as presented in Chapter 6.

An illustrative example is presented in Figure 7.1, which show how the new robust constraint sets ensure robust feasibility under the presence of mismatch error caused by disturbances.



(a) New state constraints using the reachable set of the mismatch error.



(b) New input constraints using the reachable set of the mismatch error.

Figure 7.1: Computation of the new robust state and input constraints.

Specifically, if the nominal states and inputs belong to the tightened sets, $\tilde{x}_k \in \tilde{X}_k$ and $\tilde{u}_k \in \tilde{U}_k$, then the real states and inputs are ensured to be held within the original constraints, as presented in (7.9).

$$x_k = \tilde{x}_k + e_k \quad \in \quad X_k = \tilde{X}_k \oplus E_k \tag{7.9a}$$

$$u_k = \tilde{u}_k + K(\zeta_k)e_k \quad \in \quad U_k = \tilde{U}_k \oplus K(\zeta_k)E_k \tag{7.9b}$$

However, the drawback of this method is that as the length of the prediction horizon increases, the mismatch error sets E_k will increase, and as such, there exists the possibility that the new constraint sets become empty, i.e. $\tilde{X}_i = \emptyset$ and $\tilde{U}_i = \emptyset$, resulting in an unfeasible optimization problem.

7.3 MPC Formulation

The MPC controller is formulated as a quadratic optimization problem that is solved at each sampling time t_k, in order to obtain the optimal sequence of nominal

control actions $\tilde{u}_{k+i}$ and the estimated nominal states $\tilde{x}_{k+i}$ during the prediction horizon of H_p steps. This quadratic optimization problem is presented in (7.10).

$$\min_{\Delta u_k, x_k} \quad \sum_{i=0}^{H_p-1} \left((r_{k+i+1} - \tilde{x}_{k+i+1})^\mathsf{T} Q (r_{k+i+1} - \tilde{x}_{k+i+1}) + \Delta \tilde{u}_{k+i}^\mathsf{T} R \Delta \tilde{u}_{k+i} \right) \tag{7.10a}$$

$$\text{s.t.} \quad \tilde{x}_{k+i+1} = A^d_{\zeta_{k+i}} \tilde{x}_{k+i} + B^d_{\zeta_{k+i}} \tilde{u}_{k+i} \tag{7.10b}$$

$$\tilde{u}_{k+i} = \tilde{u}_{k+i-1} + \Delta \tilde{u}_{k+i} \tag{7.10c}$$

$$\Delta \tilde{u}_{k+i} \in \left[\underline{\Delta \tilde{u}}_{k+i},\ \overline{\Delta \tilde{u}}_{k+i} \right] \tag{7.10d}$$

$$\tilde{x}_{k+i+1} \in \tilde{X}_{k+i+1} = X_{k+i} \ominus E_{k+i} \tag{7.10e}$$

$$\tilde{u}_{k+i} \in \tilde{U}_{k+i} = U \ominus K_{\zeta_{k+i}} E_{k+i} \tag{7.10f}$$

$$\tilde{x}_k = x_k \tag{7.10g}$$

$$\tilde{u}_{k-1} = u_{k-1} \tag{7.10h}$$

Where the quadratic cost function shown in (7.10a), presents two conflicting objectives: the penalization of both the reference error defined as $e_k = r_k - \tilde{x}_k$, and the incremental control action $\Delta \tilde{u}_k$. A low reference error will demand high variation in the control action and vice versa. Thus, the tuning matrices for the reference error and the variation of the control inputs, $Q \in \mathbb{S}^{n_x} \geq 0$ and $R \in \mathbb{S}^{n_u} \geq 0$ must be adequately tuned to achieve the desired dynamical behaviour.

Regarding the constraints, the LPV DT system presented in (7.10b) corresponds to the discretized version of COM presented in (3.11). The parameter $\Delta \tilde{u}_{k+i} = [\Delta a, \Delta \delta]$ shown in (7.10c), represent the incremental control action performed between sampling periods. Equation (7.10d) shows the constraint imposed on this parameter, which ensures a progressive increment on the control actions instead of abrupt changes. The constraints shown in (7.10e) and (7.10f) bound the nominal state and input such that robust feasibility of the controller is ensured, as presented in Section 7.2. Finally, the parameters x_k and u_{k-1} shown in (7.10g) and (7.10h), represent the initial state and input, which are assumed to be known by either direct measurement or estimation.

7.4 Summary of the Proposed Controller

The complete control scheme that safely maneuvers an autonomous vehicle through a vehicle-filled road is presented in the diagram shown in Figure 7.2. This diagram also shows the sampling period of each block. The proposed controller is composed of the following blocks:

- *Real vehicle plant*. This continuous time block represents the dynamics of the autonomous vehicles, which could be obtained from a real life vehicle, or by using a SOM, like the one presented in (2.27).

- *Vehicle coordination*. This block is sampled at $30\ Hz$ and is responsible for the coordination with the surrounding vehicles in order to determine the lateral bounds of a collision-free path, as presented in Chapter 6.
- *Robust MPC Controller*. This block, which is also sampled at $30\ Hz$, is responsible for the computation of a nominal control action that safely maneuvers the autonomous vehicle through a collision-free road, as presented in this Chapter.
- *H_∞-corrective controller*. This block is in charge of modifying the value of the nominal control action to robustly reject the disturbances affecting the autonomous vehicle. Consequently, it is sampled at a higher frequency than the other blocks $(300\ Hz)$.

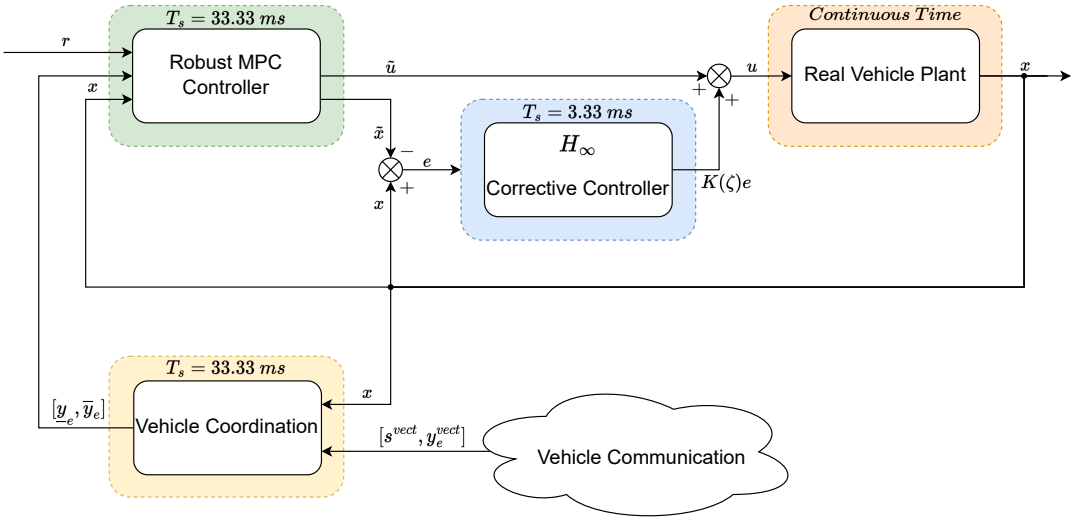


Figure 7.2: Diagram of the control scheme used to safely maneuver the autonomous vehicle in a vehicle-filled road.

Chapter 8

Tests and Results

In this Chapter, a series of tests will be performed to validate the proposed control scheme. These tests will consist on a series of simulations performed using Matlab [31] on a HP OMEN 16.1 (Intel Core i7-11800H @ 2.30GHz) Laptop, in which the assumptions presented in the following section have been made.

8.1 Tests Specifications

The road used in the proposed tests has been simulated using the *Circuit de Barcelona-Catalunya* track, which was presented in Figure 4.1. In each test simulation, a portion of $90\ m$ of this track is chosen depending on the desired curvature requirements.

As for the vehicle, the driverless UPC racing car presented in [19], has been considered. The model parameters of this vehicle were presented in Table 2.1, while the bounds on the dynamic states were shown in Table 2.2. The bounds of the kinematic states were derived from the chosen track and can be seen in Table 8.1. Note that the bounds shown for the lateral displacement of the vehicle y_e only represents the lateral limits, i.e. the width of the road, and not the bounds obtained from the vehicle coordination algorithm.

Variable	Lower Bound	Upper bound
y_e	$-5\ m$	$5\ m$
θ_e	$-\pi\ rad$	$\pi\ rad$
s	$0m$	$4650.5\ m$

Table 8.1: Bounds of the kinematic states derived from the *Circuit de Barcelona-Catalunya* track.

During the simulation, two sources of disturbances will be considered. These disturbances are caused by the dynamics of the SOM, which are unmodelled in the COM. These unmodelled dynamics were presented in Section 2.3.2, and cor-

respond to the grade of the road and the longitudinal and lateral forces caused by the wind.

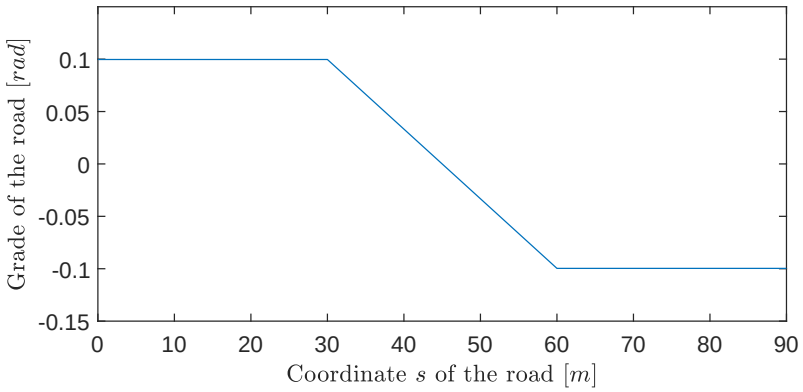


Figure 8.1: Grade (slope) of the road during the test simulations.

The considered grade for the track is presented in Figure 8.1, in which the maximum angle reached is $\pm 0.1\ rad$, which is equivalent to a steep road with a 10% inclination. The disturbances caused by the road's grade are modelled using (2.17d).

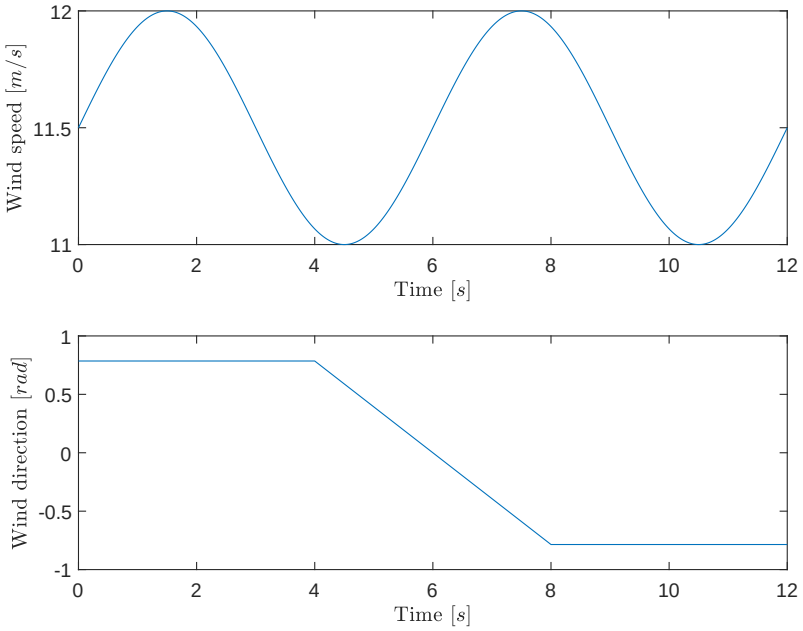


Figure 8.2: Wind velocity and direction during the test simulations.

The disturbances caused by the wind are simulated using the graph presented Figure 8.2, in which the speed and the direction of the wind is shown. The maximum speed of $12\ m/s$ corresponds to a strong breeze of level 6 on the Beaufort scale. The disturbances caused by the wind's force are modelled using Equations (2.17c), (2.21) and (2.25).

From the assumed grade of the road and the wind, the maximum disturbances affecting the dynamics of the vehicle during a time interval of $T_s = 33.33\ ms$ can be computed as shown in (8.1), in which a 25% margin has been added to bound any other possible disturbance.

$$w_{v_x}^{max}[k] = 125\%\ T_s \left[mg \sin \phi_{road}^{max} + \frac{1}{2}\rho C_{d_{long}} A_{r_{long}} \left(v_w^{max}\right)^2 \right] = 0.0257\ \frac{m}{s} \tag{8.1a}$$

$$w_{v_y}^{max}[k] = 125\%\ \frac{1}{2}\rho C_{d_{lat}} A_{r_{lat}} \left(v_w^{max}\right)^2 \frac{1}{m} = 0.0085\ \frac{m}{s} \tag{8.1b}$$

$$w_{\omega}^{max}[k] = 125\%\ \frac{1}{2}\rho C_{d_{lat}} A_{r_{lat}} \left(v_w^{max}\right)^2 \frac{l_w}{I_z} = 0.0024\ \frac{rad}{s} \tag{8.1c}$$

Where ϕ_{road}^{max} is the maximum considered grade and v_w^{max} is the maximum wind speed. From these values, the zonotope that bounds the dynamic disturbances $w[k] \in W$ can be computed, as shown in (8.2), which will be used to compute the reachable sets of the MPC controller.

$$W = \left\langle \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \begin{bmatrix} w_{v_x}^{max}/2 & 0 & 0 \\ 0 & w_{v_y}^{max}/2 & 0 \\ 0 & 0 & w_{\omega}^{max}/2 \end{bmatrix} \right\rangle \tag{8.2}$$

Finally, the MPC optimization problem has been solved using the Mosek solver [27], which was configured using the YALMIP framework [28]. The optimal length of the prediction horizon has been determined to be $H_p = 15$, which allowed a prediction of up to $0.5\ s$ ahead. As for the tuning matrices Q and R, they have been defined as shown in (8.3), where each weight has been selected after a thorough process of trial and error performed throughout multiple simulations.

$$Q = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & \frac{0.0064}{(1m/s)^2} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{0.1919}{(1\ m)^2} & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{0.0007}{(\pi/3\ rad)^2} & 0 \\ 0 & 0 & 0 & 0 & 0 & \frac{0.6396}{(1\ m)^2} \end{bmatrix} \geq 0 \tag{8.3a}$$

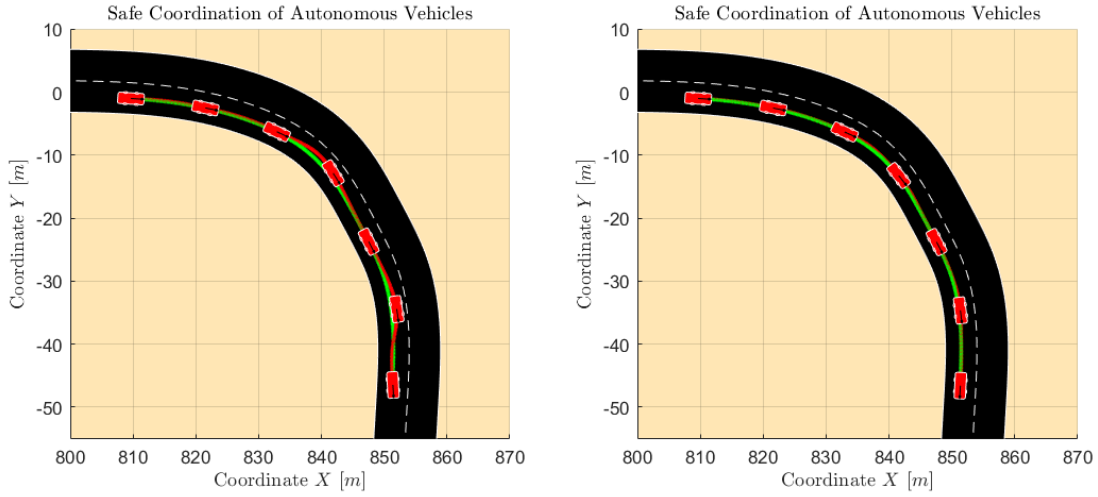
$$R = \begin{bmatrix} \frac{0.1599}{(0.5\ m/s^2)^2} & 0 \\ 0 & \frac{0.0016}{(0.05\ rad)^2} \end{bmatrix} \geq 0 \tag{8.3b}$$

Note that each weight of the tuning matrices has been normalized by dividing by the square of the maximum acceptable value according to Bryson's rule [32].

The following sections present the test simulations that have made to validate the control scheme. Specifically, these simulations evaluate the performance of the corrective controller against disturbances, and the single and multiple overtaking maneuver achieved by vehicle coordination.

8.2 Corrective Controller Results

In this section, the effectiveness of the corrective controller presented in Chapter 5 has been validated. To do so, two simulations have been performed, one with the corrective controller installed and the other one without, in which an autonomous vehicle has to follow a reference (represented in green) while rejecting the disturbances. Regarding the references, these were obtained by performing the same simulation but without any disturbances affecting the dynamics of the system, i.e. with a grade of $0°$ and wind speed of $0\ m/s$.



(a) Path without the corrective controller. (b) Path using the corrective controller.

Figure 8.3: Paths followed by the controlled vehicle both with and without the corrective controller. The red vehicle represents the position of the vehicle every 2 seconds, the green line represents the reference path while the red line represents the ground truth path.

The trajectories resulting from these simulations are presented in Figure 8.3, where the path followed by the vehicle with the corrective controller tracks the desired reference better than the vehicle without the corrective controller. It is also worth noting that due to this corrective controller, the vehicle maneuvers with a smooth behaviour with less oscillation than in the other vehicle.

These results can be better analysed by viewing the evolution of the states of each vehicle in Figure 8.4 . Clearly, the states of the vehicle with the corrective controller are closer to the reference, especially in the states y_e and θ_e.

This improvement in the reference tracking obtained by the disturbance rejection properties of the corrective controller is of utmost importance, since an ill-tracked path could derive in a vehicle going out of track or even a collision.

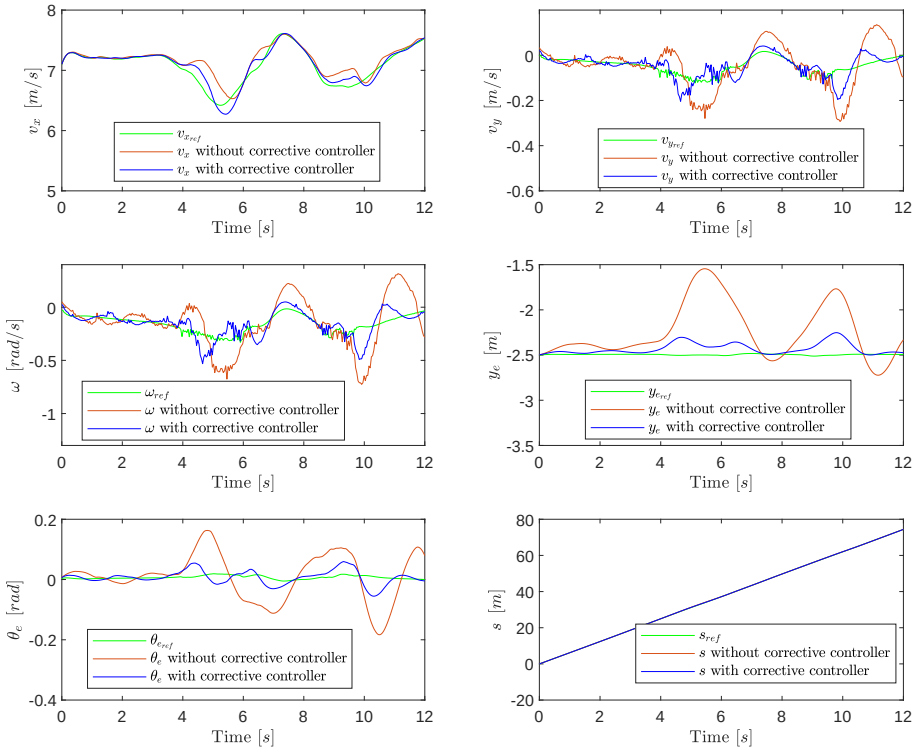


Figure 8.4: Evolution of the states of the autonomous vehicle with and without the corrective controller.

The applied control actions of each vehicle are presented in the graph shown in Figure 8.5. Since the proposed corrective controller is operating at $300\ Hz$, the control actions computed by this controller present a higher oscillation, in order to reject the disturbances affecting the states.

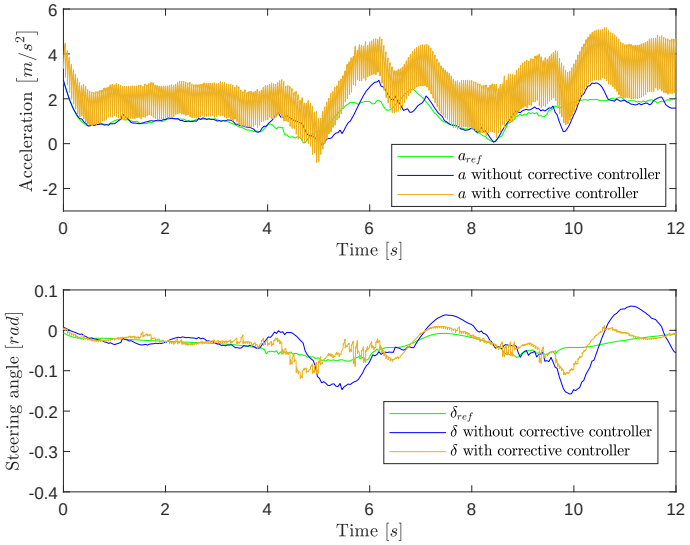


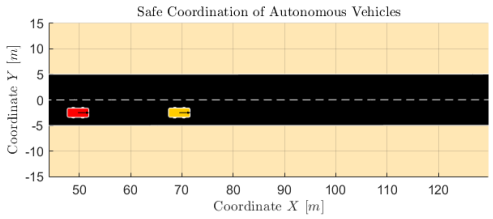
Figure 8.5: Applied control actions to the autonomous vehicle with and without the corrective controller.

8.3 Overtaking Maneuver Results

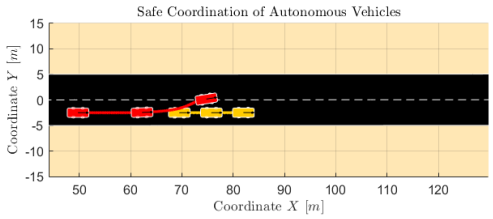
In this section, the vehicle coordination algorithms presented in Chapter 6 will be tested. Specifically, two simulations will be performed, in which an autonomous vehicle is to overtake a single and multiple vehicle obstacles, respectively.

8.3.1 Single Overtaking Maneuver

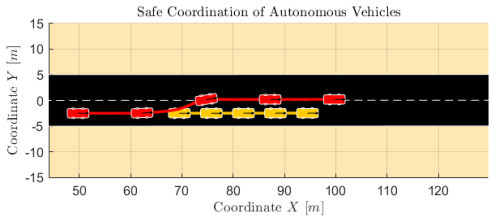
In this first test, an autonomous vehicle has to overtake a single vehicle using the vehicle coordination procedure presented in Algorithm 1. The results from this test are presented in Figure 8.6. Specifically, four graphs are shown which present the overtaking sequence from the initial position of each vehicle to the final one. The complexity of the overtaking maneuver has been reduced by considering a road without curvature, in order to evaluate the proposed control scheme in a simple scenario.



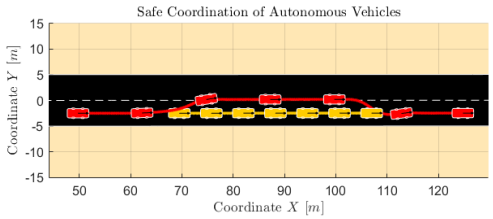
(a) Initial position of each vehicle.



(b) Beginning of the overtaking maneuver.



(c) Finalization of the overtaking maneuver.



(d) Final position of each vehicle.

Figure 8.6: Single overtaking maneuver sequence performed by the red vehicle using the proposed control scheme. The red and yellow vehicles represent the position of the controlled vehicle and the obstacle vehicle every two seconds, while red and yellow lines represent the ground truth path followed by each vehicle.

The effectiveness of the proposed algorithm is made evident in this simulation, as the controlled vehicle is capable of overtaking the obstacle vehicle vehicle, while rejecting the disturbances presented in Figure 8.1 and 8.2.

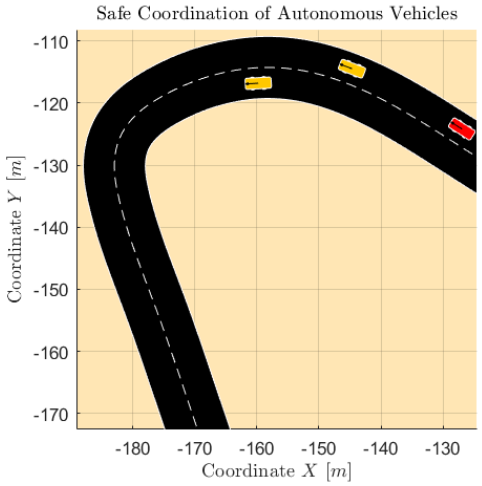
Before showing the evolution of the states and the control actions, a more complex overtaking maneuver will be presented in the next subsection.

8.3.2 Multiple Overtaking Maneuver

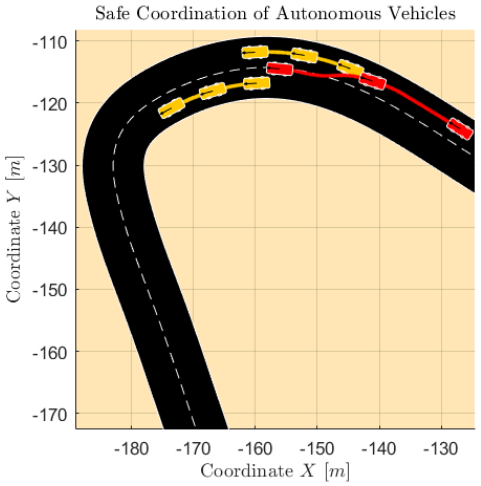
In this second test, an autonomous vehicle has to overtake two obstacle vehicles, which are driving in different lanes, by using the multiple vehicle coordination

procedure presented in Algorithm 2. The results from this test are presented in Figure 8.7 and in the video uploaded in the following link: `https://youtu.be/QxyDjeZTl9U`.

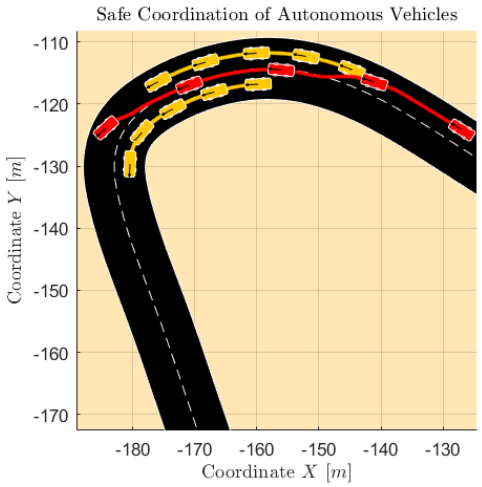
As in the previous simulation, four graphs are shown which present the overtaking sequence from the initial position of each vehicle to the final one. The complexity of the overtaking maneuver has been increased by considering a road with a close curvature, in order to evaluate the proposed control scheme in a demanding scenario.



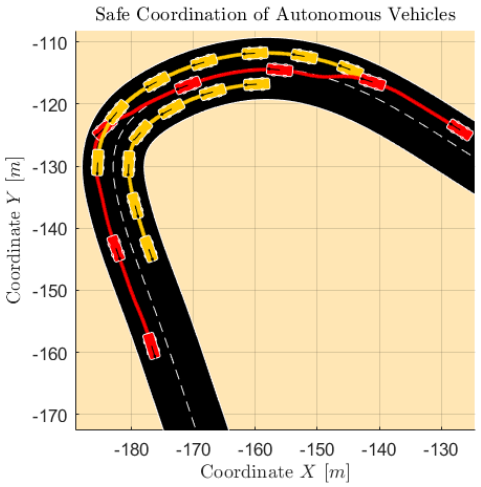
(a) Initial position of each vehicle.



(b) Beginning of the overtaking maneuver.



(c) Finalization of the overtaking maneuver.



(d) Final position of each vehicle.

Figure 8.7: Multiple overtaking maneuver sequence performed by the red vehicle using the proposed control scheme. The red and yellow vehicles represent the position of the controlled vehicle and the obstacle vehicle every two seconds, while red and yellow lines represent the ground truth path followed by each vehicle.

This overtaking sequence does not only demonstrate the effectiveness of the proposed control scheme under the disturbances presented in Figure 8.1 and 8.2,

but in the presence of a complex dynamical scenario, such as the closed curvature considered.

These results can be better analysed by viewing the evolution of the states in the graphs presented in Figure 8.8. In each graphs, four lines have been plotted, which correspond to the nominal states predicted by the MPC controller, $\tilde{x}$, the real states affected by the disturbances, x, and the upper and lower bound of each state, $\underline{x}$ and $\overline{x}$.

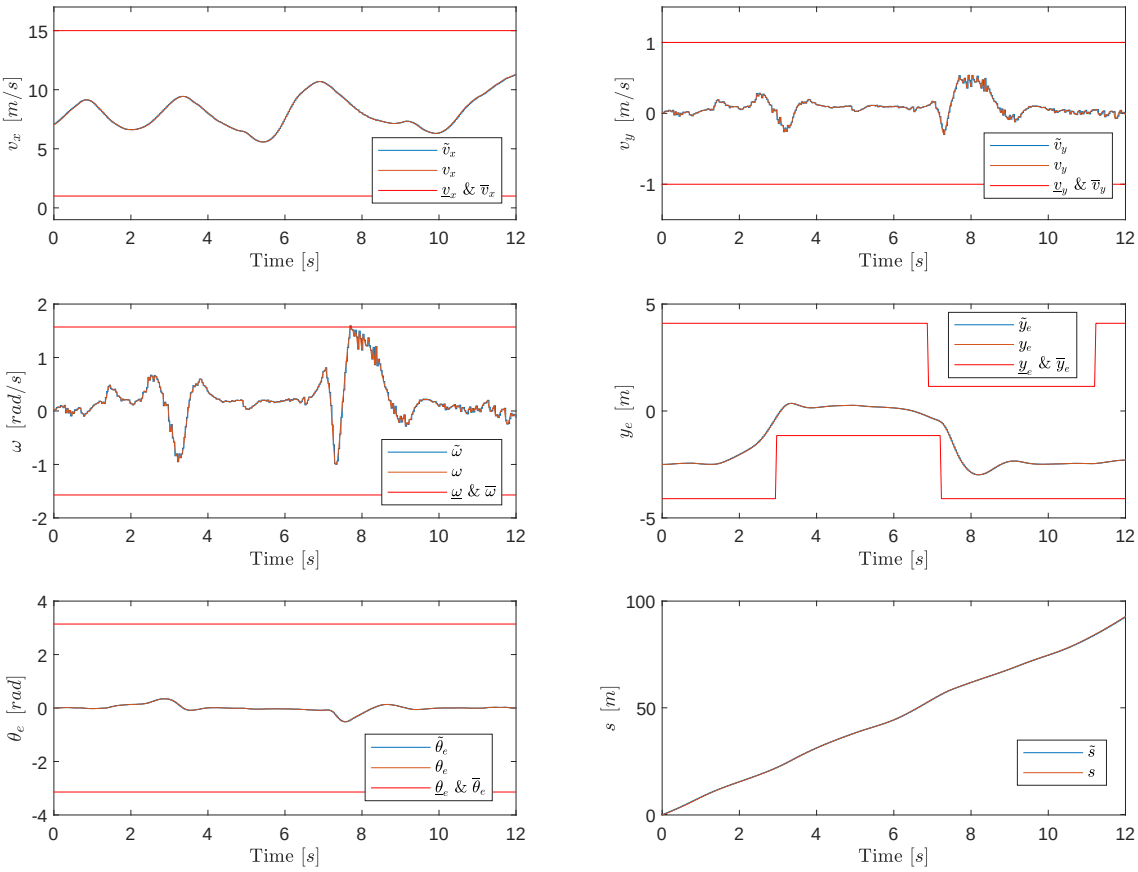


Figure 8.8: Nominal and real states of the controlled vehicle during the multiple overtaking maneuver.

As expected, the nominal states are held within the constraints, as ensured by the MPC optimization problem. Similarly, the real states are also held within the constraints thanks to the computation of the reachable sets, presented in Section 7.1, which were used to compute a robust set of constraints that ensured the feasibility of the problem. Also, the error between the nominal and the real states is almost inappreciable, thanks to the efforts of the corrective controller.

It is also worth noticing the bounds of the lateral displacement of the vehicle, $\underline{y}_e$ and $\overline{y}_e$, that avoid the collisions with the surrounding obstacles. These were computed using the vehicle coordination procedure presented in Algorithm 2.

The nominal and real control actions applied to the controlled vehicle during

the multiple overtaking maneuver, as well as the control action bounds, are presented in the graph shown in Figure 8.9.

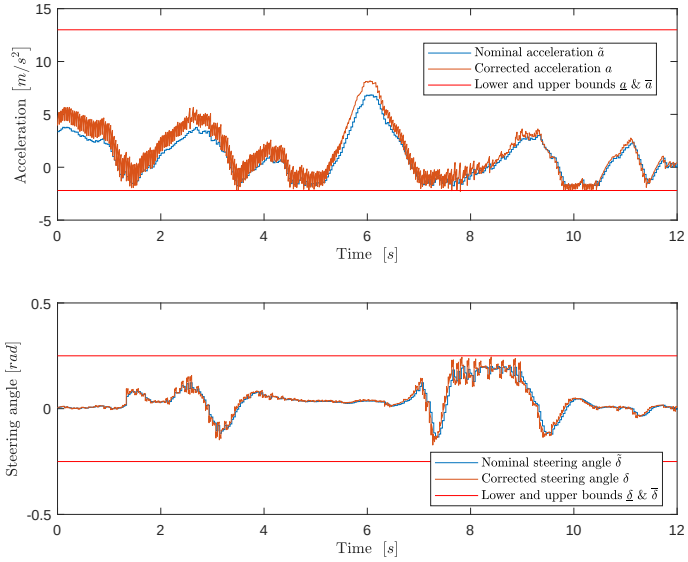


Figure 8.9: Nominal and real inputs of the vehicle during the simulation.

Similarly to the nominal and real states presented in Figure 8.8, neither the nominal, nor the real control actions violate the considered constraints, due to the mentioned robust set of constraints that ensured the feasibility of the problem.

8.4 Execution Time Results

The execution time of the Mosek solver during the multiple overtaking simulation presented in Section 8.3.2, is shown in Figure 8.10.

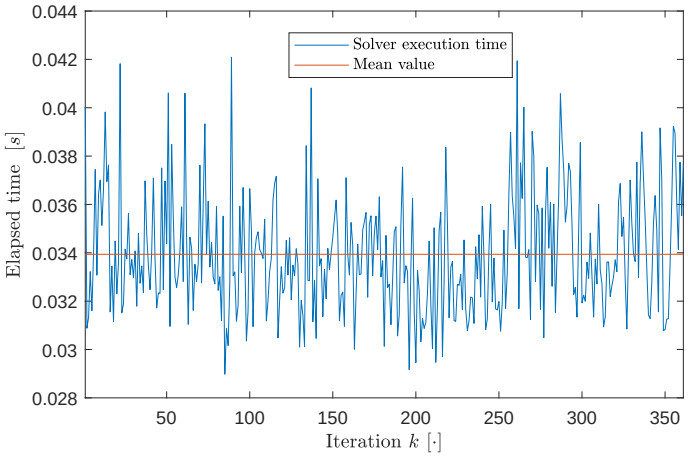


Figure 8.10: Solver execution time per iteration during the simulation.

Unfortunately, the mean value of approximately $0.034\ s$ is two high for it to be

solved in real time, since, in the proposed control scheme, the MPC should be executed at $30\ Hz$.

Nonetheless, the proposed quadratic optimization problem obtained by using the LPV model of the vehicle presents a huge reduction in the execution time in comparison with the non-linear optimization problem obtained from considering the non-linear vehicle model.

For example, a single execution of the non-linear MPC using the fmincon solver [33] would require an execution time of approximately $30\ s$, which is almost 900 times the execution time required in the proposed LPV MPC.

Chapter 9

Social, Economic and Environmental Impact

The project's sustainability study can be divided into the social, economic and environmental impact.

9.1 Social Impact

Autonomous vehicles will relieve passengers from the burden of driving, planning routes, or even paying attention to the road, thus improving the driving experience by providing extra time that can be exploited in other activities [10].

The extensive usage of autonomous vehicles together with the development of V2V communication protocols will reduce most of the crashes attributed to human factors, thus avoiding multiple injuries and saving countless of lives [7]. while intelligent route optimization techniques will reduce traffic congestion, which will notably reduce the total number of vehicles circulating on the road, thus reducing air pollution [8].

New mobility solutions will be implemented thanks to the spread of coordinated autonomous vehicles, such as car sharing, in which a fleet of autonomous vehicles will be available for the public [34]; personalized public transport, where autonomous buses can dynamically modify their route based on their passengers necessities [35]; or smart parking solutions, in which autonomous vehicles can cooperate between themselves to satisfy their parking needs [36].

9.2 Economic Impact

A direct economic impact can be directly derived from the usage of coordinated autonomous vehicles, as their precise driving and intelligent route optimization can reduce fuel consumption and increase the durability of the mechanical parts of the vehicle [37].

The aforementioned mobility solutions can also promote the development of new industries which can economically exploit these solutions.

9.3 Environmental Impact

As mentioned, the intelligent route optimization together with the precise driving of autonomous vehicles can reduce the carbon footprint of daily transportation.

Regarding the development of the project, since it was integrally developed using a laptop, the analysis of the environmental impact will only take into account the power consumption of the laptop.

According to the manufacturer, the HP OMEN 16.1 (Intel Core i7-11800H @ 2.30GHz) laptop has a power supply of 230W [38]. If a total of 840 hours of work are considered in the development of the project, then the total power consumption can be estimated in 193.2kWh.

Thus, by using the *greenhouse gas equivalencies calculator* developed by the U.S. Environmental Protection Agency [39], the estimated carbon footprint of the project is 83.6 kilograms of Carbon Dioxide (CO_2) equivalent.

Chapter 10

Project Budget

The budget associate with this project can be divided into costs related to software licenses, personnel expenses and the amortization of the used equipment.

The personnel expenses have been computed considering a total of 840 hours of work, which have been distributed throughout the 7 months dedicated to the development of the project. The engineer wage has been estimated to be of 15 € per hour. Thus, the personnel cost can be computed as shown in (10.2)

$$\text{Personnel expenses} = 840 \text{ hours } \cdot 15€/\text{hour} = 12,600€ \tag{10.1}$$

The equipment used consists in a laptop with an estimated value of 1,600€. Thus, by considering the maximum annual percentage of amortization for electronic equipment of 20 %, as stated by the Spanish Law 27/2014, of November 27, on Corporation Tax [40], the amortization cost can be computed as shown in (10.2). Where the ratio of $7/12$ has been added to consider the project's estimated duration of 7 months.

$$\text{Amortization cost} = 20\% \cdot 1,600€ \cdot \frac{7}{12} = 186.67€ \tag{10.2}$$

A summary of the aforementioned costs, considering a value-added tax of 21%, are presented in Table 10.1.

Concept	Quantity	Cost (€)
Matlab license	1	2,541
Matlab Optimization Toolbox	1	1,452
Engineer wage	1	12,600
Laptop Amortization	1	186.67
Total Cost		**16,779.67**

Table 10.1: Budget associated to the project.

In conclusion, the total cost invested in developing the project was approximately 16,779.67€.

Part III

Conclusions

Chapter 11

Conclusions and Future Work

This thesis presented a robust control scheme designed for the safe coordination of autonomous vehicles. Specifically, a robust controller was designed to autonomously maneuver an autonomous vehicle through a road, while avoiding collisions with other vehicles. The collision avoidance was achieved by proposing a coordination algorithm between the vehicles, while the robustness was ensured by using a robust tube-based MPC, together with a corrective controller in charge of rejecting disturbances.

To provide a basis for future research, the main conclusions of the thesis, as well as the proposed future work are presented in this chapter.

11.1 Conclusions

The main conclusions reached in each chapter are summarized below:

- **Chapter 2** was devoted to the mathematical modelling of vehicles. Different models were presented, highlighting the kinematic curvature model and the dynamic bicycle model, which proved to be suitable in the proposed control scheme. Furthermore, a SOM was proposed, from which a COM was derived. The later model presented a series of unmodelled dynamics that functioned as disturbances.
- **Chapter 3** presented the LPV versions of the vehicle models. This proved to be essential in the development of the project as it allowed working with the linear control theory and transformed the MPC controller from a non-linear to a quadratic optimization problem, which can be solved more efficiently, thus reducing the execution time.
- **Chapter 4** presented the polytopic LPV model of the dynamic vehicle model. Specifically, the PCA parameter set mapping algorithm was successfully applied, in order to reduce the order of the polytopic model, while maintaining a valid approximation of the system.
- **Chapter 5** presented the design of a robust LPV corrective controller capa-

ble of rejecting the disturbances affecting the system. This controller was synthesized by applying the H_∞-optimal control theory. Specifically, the controller was computed by efficiently solving a LMI optimization problem evaluated on the vertices of the polytopic LPV model.

- **Chapter 6** presented an algorithm by which autonomous vehicles could coordinate with each other, in order to compute the bounds of collision-free paths.
- **Chapter 7** was devoted to the robust tube-based MPC and its formulation. Specifically, the robustness was ensured by computing a series of zonotopic reachable sets (also known as tubes), which were used in the generation of a set of tightened constraints that ensured the robust feasibility of the control scheme.
- **Chapter 8** presented a series of test performed to validate the safe coordination of autonomous vehicles. The results from these test proved that the proposed control scheme was effective in maneuvering the autonomous vehicle through a vehicle-filled road, while rejecting disturbances; and that the use of LPV models greatly reduce the solver's execution time, while still being a faithful approximation of the vehicle.
- **Chapter 9** provided a sustainability study of the project, with special emphasis on its social, economic and environmental impact. Additionally, the carbon footprint of the developed project was computed.
- **Chapter 10** presented the budget associated to the project, considering the material costs, personnel expenses and the amortization of the used equipment.

11.2 Future Work

This section summarizes the open issues that could be addressed and/or improved in future work.

- The vehicle coordination algorithm could be further improved to cope with more complex scenarios, such as intersection management, optimal platooning, ramp merging or traffic flow optimization.
- The COM could be obtained through a parameter estimation process, in order to evaluate the robustness of the proposed controller against parametric uncertainty.
- The disturbance rejection of the proposed H_∞-optimal corrective controller could be compared with corrective controllers synthesized by applying H_2-optimal or mixed H_2-H_∞-optimal control theory.
- The robustness of the proposed tube-based MPC controller could be compared with other robust MPC architectures, such as the Min-Max MPC [41].
- The execution time of the MPC optimization problem could be lowered to

achieved the real-time requirements by considering other commercials solvers such as Gurobi [42], CPLEX [43] or XPRESS [44].

- The proposed method has only been validated in simulation. Thus, a future line of work could be developed by testing the proposed control scheme in an advanced simulators such as CARLA [45] or a real-life vehicle.

Part IV

Appendices

Appendix A

Zonotope Theory

A zonotope $Z \in \mathbb{R}^n$ is a special case of polytope with centre $c \in \mathbb{R}^n$ and generator matrix $G = [g_1, \ldots, g_p] \in \mathbb{R}^{n \times p}$, as defined in (A.1). The order of a zonotope is defined as $\rho = \frac{p}{n}$ and represents a dimensionless measure of the representation size [46].

$$Z \coloneqq \langle c,\ G \rangle = \left\{ c + \sum_{i=1}^{p} \beta_i g_i \;\middle|\; \beta_i \in [-1,\ 1] \right\} \tag{A.1}$$

Zonotopes are a useful tool for representing sets in high-dimensional spaces in a compact way. Another advantage is that the computation of linear maps and the Minkowski sum are especially efficient, as presented below.

Linear Map of a Zonotope

The linear map of a zonotope $M \otimes Z$ with $Z \in \mathbb{R}^n$ and $M \in \mathbb{R}^{q \times n}$ is defined as in (A.2).

$$M \otimes Z = \langle M \cdot c,\ M \cdot G \rangle \tag{A.2}$$

A graphical example of this operation is presented in Figure A.1, in which a zonotope is linearly mapped by the rotation matrix presented in (A.3).

$$M = R(30°) = \begin{bmatrix} \cos 30° & -\sin 30° \\ \sin 30° & \cos 30° \end{bmatrix} \tag{A.3}$$

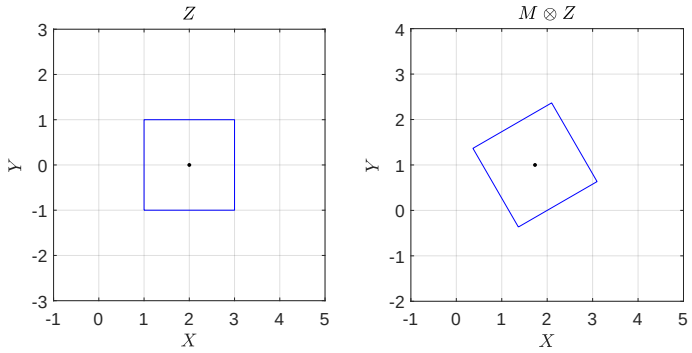


Figure A.1: Linear map of a zonotope.

Minkowski Sum of two Zonotopes

In geometry, the Minkowski sum of two sets A and B is defined by adding each vector in A to each vector in B, as presented in (A.4).

$$A \oplus B = \{a + b \mid a \in A,\ b \in B\} \tag{A.4}$$

The Minkowski Sum of two zonotopes $Z_1 \oplus Z_2$ with $Z_1, Z_2 \in \mathbb{R}^n$ is defined as in (A.5).

$$Z_1 \oplus Z_2 = \langle c_1 + c_2,\ [G_1, G_2] \rangle \tag{A.5}$$

A graphical example of this operation is presented in Figure A.2.

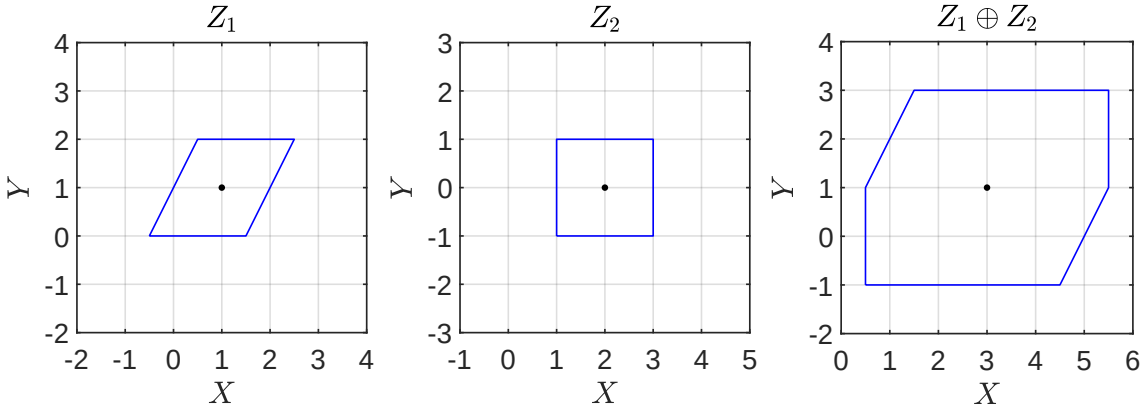


Figure A.2: Minkowski sum of two zonotopes.

Minkowski difference of two Zonotopes

The Minkowski difference of two sets A and B is not defined as $A \oplus (-B)$, but rather by using the complement operation, as presented in (A.6).

$$A \ominus B = \left(A^c \oplus (-B)\right)^c \tag{A.6}$$

Since the complement of zonotope cannot be efficiently computed, the Minkowski difference of two zonotopes $Z_1 \oplus Z_2$ with $Z_1, Z_2 \in \mathbb{R}^n$ is alternatively defined as in (A.7).

$$Z_1 \ominus Z_2 = \{Z \in \mathbb{R}^n \mid Z \oplus Z_2 \subseteq Z_1\} \tag{A.7}$$

Such that $(Z_1 \ominus Z_2) \oplus Z_2 \subseteq Z_1$. When $\tilde{Z}_1 = Z \oplus Z_2$, where Z is an arbitrary zonotope, such that $Z_2 \subseteq \tilde{Z}_1$, the Minkowski difference satisfies $\tilde{Z}_1 \ominus Z_2 \neq \emptyset$ and thus, $\left(\tilde{Z}_1 \ominus Z_2\right) \oplus Z_2 = \tilde{Z}_1$. A graphical example of this operation is presented in Figure A.3.

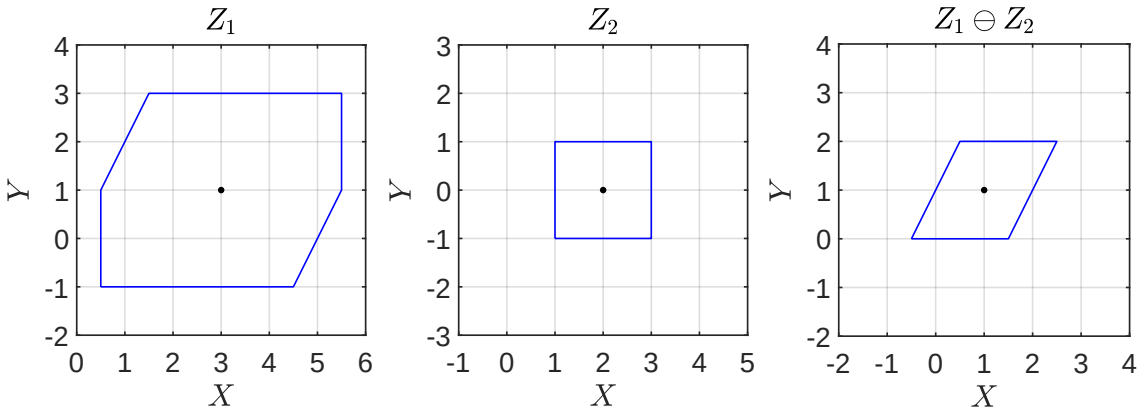


Figure A.3: Minkowski difference of two zonotopes.

Unfortunately, the computation of the zonotopic Minkowski difference is not as efficient as the linear map or the Minkowski sum presented in (A.2) and (A.5). As such, more efficient approximations of this operation have been proposed, e.g. [47]. Nonetheless, these approximations are still far less efficient than the zonotopic linear map or the Minkowski sum operations. In time-critical controllers, the zonotopic Minkowski difference can be further approximated, at the expense of reducing its accuracy, by implementing the interval difference presented in (A.8), in which the Interval Hull of each zonotope has been previously computed.

$$[\underline{a}, \overline{a}] - [\underline{b}, \overline{b}] = [\underline{a} - \underline{b}, \overline{a} - \overline{b}] \tag{A.8}$$

Interpretation of Zonotopes

Zonotopes can be interpreted in three ways depending on the desired application [46]:

- A zonotope can be interpreted as an affine transformation defined by the projection of a p-dimensional unit hypercube $C = [-1, 1]^p$ onto the n-dimensional space by the generator matrix G, followed by the translation to the center c: $Z = c \oplus G \otimes C$.

- A zonotope can be interpreted as the Minkowski sum of line segments $l_i = \beta_i g_i$ where $\beta_i \in [-1,\ 1]$.
- A zonotope can be interpreted as a polytope whose j-faces are centrally symmetric.

Bibliography

[1] World Health Organization. Road traffic injuries. https://www.who.int/news-room/fact-sheets/detail/road-traffic-injuries#:~:text=Approximately%201.3%20million%20people%20die,pedestrians%2C%20cyclists%2C%20and%20motorcyclists. Accessed: 2022-09-16.

[2] Alireza Pakgohar, Reza Sigari Tabrizi, Mohadeseh Khalili, and Alireza Esmaeili. The role of human factor in incidence and severity of road crashes based on the cart and lr regression: a data mining approach. *Procedia Computer Science*, 3:764–769, 2011. World Conference on Information Technology.

[3] World Health Organization. Ambient (outdoor) air pollution. https://www.who.int/news-room/fact-sheets/detail/ambient-(outdoor)-air-quality-and-health. Accessed: 2022-09-16.

[4] U.S. Environmental Protection Agency. Inventory of u.s. greenhouse gas emissions and sinks. https://www.epa.gov/ghgemissions/inventory-us-greenhouse-gas-emissions-and-sinks. Accessed: 2022-09-16.

[5] INRIX. Global traffic scorecard. 2019. https://static.poder360.com.br/2019/02/INRIX_2018_Global_Traffic_Scorecard_Report__final_.pdf. Accessed: 2022-09-16.

[6] The New Now. Vehículos autónomos, estado de la cuestión. https://www.thenewnow.es/negocio/vehiculos-autonomos-estado-la-cuestion/. Accessed: 2022-09-16.

[7] Keshav Bimbraw. Autonomous cars: Past, present and future - a review of the developments in the last century, the present scenario and the expected future of autonomous vehicle technology. *ICINCO 2015 - 12th International Conference on Informatics in Control, Automation and Robotics, Proceedings*, 1:191–198, 01 2015.

[8] Nikhil Menon, Natalia Barbour, Yu Zhang, Abdul Rawoof Pinjari, and Fred Mannering. Shared autonomous vehicles and their potential impacts on household vehicle ownership: An exploratory empirical assessment. *International Journal of Sustainable Transportation*, 13(2):111–122, 2019.

[9] Prateek Bansal and Kara M Kockelman. Forecasting americans' long-term adoption of connected and autonomous vehicle technologies. *Transportation*

Research Part A: Policy and Practice, 95:49–63, 2017.

[10] Stefano Mariani, Giacomo Cabri, and Franco Zambonelli. Coordination of autonomous vehicles: taxonomy and survey. *ACM Computing Surveys (CSUR)*, 54(1):1–33, 2021.

[11] Eugenio Alcalá, Vicenç Puig, Joseba Quevedo, Teresa Escobet, and Ramon Comasolivas. Autonomous vehicle control using a kinematic lyapunov-based technique with lqr-lmi tuning. *Control engineering practice*, 73:1–12, 2018.

[12] Danio Caporale, Adriano Fagiolini, Lucia Pallottino, Alessandro Settimi, Andrea Biondo, Francesco Amerotti, Federico Massa, Stefano De Caro, Andrea Corti, and Luca Venturini. A planning and control system for self-driving racing vehicles. In *2018 IEEE 4th International Forum on Research and Technology for Society and Industry (RTSI)*, pages 1–6, 2018.

[13] Rajesh Rajamani. *Vehicle dynamics and control*. Springer Science & Business Media, 2011.

[14] Rolf Isermann. *Automotive control: modeling and control of vehicles*. Springer, 2022.

[15] Jason Kong, Mark Pfeiffer, Georg Schildbach, and Francesco Borrelli. Kinematic and dynamic vehicle models for autonomous driving control design. In *Conference: 2015 IEEE Intelligent Vehicles Symposium (IV)*, pages 1094–1099, 06 2015.

[16] Robin Verschueren, Stijn De Bruyne, Mario Zanon, Janick V Frasch, and Moritz Diehl. Towards time-optimal race car driving using nonlinear mpc in real-time. In *53rd IEEE conference on decision and control*, pages 2505–2510. IEEE, 2014.

[17] Hans B Pacejka and Robin S Sharp. Shear force development by pneumatic tyres in steady state conditions: a review of modelling aspects. *Vehicle system dynamics*, 20(3-4):121–175, 1991.

[18] Hans Pacejka. *Tire and vehicle dynamics*. Elsevier, 2005.

[19] Eugenio Alcalá, Vicenç Puig, Joseba Quevedo, and Olivier Sename. Fast zonotope-tube-based lpv-mpc for autonomous vehicles. *IET Control Theory & Applications*, 14, 12 2020.

[20] Ron Larson and Bruce H Edwards. *Calculus of a single variable: Early transcendental functions*. Cengage Learning, 2014.

[21] Eugenio Alcalá Baselga. *Advances in planning and control for autonomous vehicles*. PhD thesis, Universitat Politècnica de Catalunya, 2020.

[22] Pierre Apkarian, Pascal Gahinet, and Greg Becker. Self-scheduled h infinity control of linear parameter-varying systems: a design example. *Automatica*, 31(9):1251–1261, 1995.

[23] Damiano Rotondo, Fatiha Nejjari, and Vicenç Puig. Quasi-lpv modeling,

identification and control of a twin rotor mimo system. *Control Engineering Practice*, 21(6):829–846, 2013.

[24] Andreas Kwiatkowski and Herbert Werner. Pca-based parameter set mappings for lpv models with fewer parameters and less overbounding. *IEEE Transactions on Control Systems Technology*, 16(4):781–788, 2008.

[25] TUM Institute of Automotive Technology. Racetrack database. `https://github.com/TUMFTM/racetrack-database`, 2021.

[26] Ryan James Caverly and James Richard Forbes. Lmi properties and applications in systems, stability, and control theory. *arXiv preprint arXiv:1903.08599*, 2019.

[27] MOSEK ApS. *The MOSEK optimization toolbox for MATLAB manual. Version 10.0.*, 2022.

[28] J. Löfberg. Yalmip : A toolbox for modeling and optimization in matlab. In *In Proceedings of the CACSD Conference*, Taipei, Taiwan, 2004.

[29] Zhouhua Peng, Jun Wang, Dan Wang, and Qing-Long Han. An overview of recent advances in coordinated control of multiple autonomous surface vehicles. *IEEE Transactions on Industrial Informatics*, 17(2):732–745, 2021.

[30] Richard M Murray. Recent research in cooperative control of multivehicle systems. *Journal of Dynamic Systems, Measurement, and Control*, 129(5):571–583, May 2007.

[31] MATLAB. *version 9.11.0 (R2021b)*. The MathWorks Inc., Natick, MA, USA, 2021.

[32] Joao P Hespanha. *Linear systems theory*. Princeton university press, 2018.

[33] Matlab optimization toolbox, 2021. The MathWorks, Natick, MA, USA.

[34] Mingyang Hao and Toshiyuki Yamamoto. Shared autonomous vehicles: A review considering car sharing and autonomous vehicles. *Asian Transport Studies*, 5(1):47–63, 2018.

[35] Mitja Stiglic, Niels Agatz, Martin Savelsbergh, and Mirko Gradisar. Enhancing urban mobility: Integrating ride-sharing and public transit. *Computers & Operations Research*, 90:12–21, 2018.

[36] Trista Lin, Hervé Rivano, and Frédéric Le Mouël. A survey of smart parking solutions. *IEEE Transactions on Intelligent Transportation Systems*, 18(12):3229–3253, 2017.

[37] M. Nadeem Ahangar, Qasim Z. Ahmed, Fahd A. Khan, and Maryam Hafeez. A survey of autonomous vehicles: Enabling communication technologies and challenges. *Sensors*, 21(3), 2021.

[38] HP. *OMEN by HP Laptop 16-b0009na Datasheet*, 12 2021. `https://www8.hp.com/h20195/V2/GetPDF.aspx/c07637462`.

[39] U.S. Environmental Protection Agency. Greenhouse gas equivalencies calculator. `https://www.epa.gov/energy/`

greenhouse-gas-equivalencies-calculator. Accessed: 2022-09-10.

[40] BOE. Ley 27/2014, de 27 de noviembre, del impuesto sobre sociedades. https://www.boe.es/eli/es/l/2014/11/27/27/con.

[41] Johan Löfberg. Min-max approaches to robust model predictive control. *Linköping Studies in Science and Technology. Dissertations. No*, 01 2003.

[42] Gurobi Optimization, LLC. Gurobi Optimizer Reference Manual, 2022.

[43] Kenneth Holmström, Anders O Göran, and Marcus M Edvall. User's guide for tomlab/cplex v12. 1. *Tomlab Optim. Retrieved*, 1:2017, 2009.

[44] Dash Optimization. Xpress-optimizer reference manual. *Dash Optimization Ltd., Englewood Cliffs, NJ*, 2007.

[45] Alexey Dosovitskiy, German Ros, Felipe Codevilla, Antonio Lopez, and Vladlen Koltun. Carla: An open urban driving simulator. In *Conference on robot learning*, pages 1–16. PMLR, 2017.

[46] P McMullen. Gm ziegler lectures on polytopes (graduate texts in mathematics, vol. 152, springer-verlag, berlin-heidelberg-new york-london-paris-tokyo-hong kong 1995), ix+ 370 pp., softcover: 3 540 94365 x,£ 21, hardcover: 3 540 94329 3,£ 47. *Proceedings of the Edinburgh Mathematical Society*, 39(1):189–190, 1996.

[47] Matthias Althoff. On computing the minkowski difference of zonotopes. *arXiv preprint arXiv:1512.02794*, 2015.