Universitat Politècnica de Catalunya

Programa de Doctorat:

Automàtica, Robòtica i Visió

Tesis Doctoral

Advances in Planning and Control for Autonomous Vehicles

Eugenio Alcalá Baselga

Directors: Dr. Vicenç Puig Cayuela i Dr. Joseba Quevedo Casin

January 2020

Abstract

This thesis presents some contributions to the state of the art of state estimation, automatic control and trajectory planning fields applied to autonomous vehicles.

Such contributions have a common aspect throughout the thesis, all of them are model-based techniques. The Linear Parameter Varying (LPV) and Takagi-Sugeno (TS) theory are used to generate control-oriented models by using the non-linear embedding approach. Several vehicle models are proposed depending on the application and estimation-control-planning technique. First, non-linear vehicle formulations are presented. Later, the same models are represented in the LPV form.

In the area of control and estimation, the thesis shows different approaches for different applications: normal and racing driving modes. First, for normal driving, gain-scheduling (GS) LPV state feedback techniques are developed. In the first instance, an LPV-Linear Quadratic Regulator (LQR) design via Linear Matrix Inequality (LMI) formulation is stated for control at low velocities. Later, a cascade scheme including kinematic and dynamic control layers is presented to improve the last design. Here, both controller designs are set up using the LPV-LQR design via LMI formulation and a LPV-Unknown Input Observer (UIO) is presented for estimating vehicle states and exogenous friction force. Second, for racing driving, optimal techniques are explored leading to introduce the Model Predictive Control (MPC) technique as a basis for racing behaviours. In the first instance, the cascade scheme is maintained where the outer control layer is governed by a TS-MPC controller. At this point, an advanced estimation technique is proposed, the TS-Moving Horizon Estimator-UIO (TS-MHE-UIO). It is shown that by using the TS formulation both optimal-based controller and estimator reduce greatly the computational effort in comparison to their non-linear formulation. Then, the idea of designing a unique controller is explored through the LPV-MPC technique. In this case, it is shown the potential of this strategy being able to be executed in real time in small embedded platforms for controlling the vehicle in racing situations. Finally, an online robust MPC is considered that aims at improving the computational load using zonotope theory while preserving high levels of robustness and performance in racing scenarios.

In the area of planning, the thesis focuses on trajectory planning approaches from the optimal point of view. First, the non-linear MPC is formulated as a planner (NL-MPP) in space domain where the goal is the minimization of the total lap time. Later, an innovative real time solution is explored leading to a LPV-MPP. The method follows the structure of the model predictive optimal strategy where the main objective is to maximize the velocity while fulfilling varying constraints such as obstacles. In particular, the aim is on reformulating the non-linear original problem into a pseudo-linear problem by convexifying the objective function and making use of the LPV vehicle formulation.

Resumen

Esta tesis presenta algunos avances en los campos de la estimación de estados, el control automático y la planificación de trayectorias aplicados a vehículos autónomos.

Tales contribuciones comparten un particular aspecto a lo largo de la tesis, todas ellas son técnicas basadas en modelos. La teoría de Variación Lineal de Parámetros (VLP) y Takagi-Sugeno (TS) se utilizan para generar modelos orientados al control mediante el uso de enfoques de inclusión no lineal y de no linealidad sectorial. Se proponen diferentes modelos de vehículos según la aplicación y la técnica de estimación-control-planificación. Primero, se presentan los modelos de vehículos en la formulación no lineal. Más tarde, dichos modelos se reformulan como VLP.

En el área de control y estimación, la tesis muestra diferentes enfoques para diferentes aplicaciones: modos de conducción normal y de carreras. Primero, para la conducción normal, se desarrollan técnicas de retroalimentación de estado VLP de programación de ganancia (PG). En primera instancia, un diseño de Regulador Cuadrático Lineal (RCL) VLP a través de la formulación de Desigualdad de Matriz Lineal (DML) se establece para el control del vehiculo a bajas velocidades. Más tarde, se presenta un esquema en cascada que incluye capas de control cinemático y dinámico para mejorar el último diseño. Aquí, ambos diseños de controlador se realizan utilizando el diseño VLP-LQR a través de la formulación LMI y un Observador de Entrada Desconocida (OED) VLP está preestablecido para estimar los estados del vehículo, así como la fuerza de fricción que actúa sobre el vehículo. Segundo, para la conducción en carreras, se exploran técnicas óptimas que conducen a introducir la técnica de Control de Modelo Predictivo (CMP) como base para los comportamientos de carrera. En primera instancia, el esquema en cascada se mantiene donde la capa de control externa está gobernada por un controlador TS-CMP. En este punto, se presenta una técnica de estimación avanzada, el TS-Moving Horizon Estimator-UIO (TS-MHE-OED). Se demuestra que al usar la formulacion TS, tanto el controlador como el estimador óptimos reducen en gran medida el esfuerzo computacional en comparación con su formualción no lineal. Luego, la idea de diseñar un controlador único se explora a través de la técnica VLP-CMP. En este caso, se muestra el potencial de esta estrategia para poder ejecutarse en tiempo real en pequeñas plataformas integradas para controlar el vehículo en situaciones de carrera. Finalmente, se considera un CMP robusto en línea que tiene como objetivo mejorar la carga computacional utilizando la teoría de zonótopos mientras preserva altos niveles de robustez y rendimiento en escenarios de carreras.

En el área de planificación, la tesis se centra en los enfoques de planificación de trayectorias desde el punto de vista óptimo. Primero, el CMP no lineal se formula como un planificador (NL-MPP) en el dominio espacial donde el objetivo es la minimización del tiempo de vuelta total. Más tarde, se explora una solución innovadora en tiempo real que conduce a un VLP-MPP. El método sigue la estructura de la estrategia óptima de modelo predictivo donde el objetivo principal es maximizar la velocidad mientras se cumplen las limitaciones dinámicas del vehiculo. En particular, el objetivo es reformular el problema original no lineal en un problema pseudo-lineal convexificando la función objetivo y haciendo uso de la formulación del vehículo VLP.

To my parents,
to my brother,
to my grandparents,
to my friends.

Contents

List of Figures

List of Tables

Acronyms

ANFIS	**A**rtificial **N**euro **F**uzzy **I**nterference **S**ystem
CLQOC	**C**onstrained **L**inear **Q**uadratic **O**ptimal **C**ontrol
CNLOC	**C**onstrained **N**on - **L**inear **O**ptimal **C**ontrol
CT	**C**ontinuous-**T**ime
DT	**D**iscrete-**T**ime
ECU	**E**lectronic **C**ontrol **U**nit
FL	**F**uzzy **L**ogic
GPS	**G**lobal **P**ositioning **S**ystem
GS	**G**ain-**S**cheduling
IMU	**I**nertial **M**easurement **U**nit
IO	**I**nput-**O**utput
LIDAR	**Li**ght **D**etection **A**nd **R**anging
LMI	**L**inear **M**atrix **I**nequality
LQ	**L**inear **Q**uadratic
LQR	**L**inear **Q**uadratic **R**egulator
LPV	**L**inear **P**arameter **V**arying
LPV-IO	**L**inear **P**arameter **V**arying - **I**nput **O**utput
LPV-SS	**L**inear **P**arameter **V**arying - **S**tate-**S**pace
LS-SVM	**L**east **S**quares - **S**upport **V**ector **M**achine
LTI	**L**inear **T**ime **I**nvariant
LTV	**L**inear **T**ime **V**arying
MHE	**M**oving **H**orizon **E**stimator
MIMO	**M**ulti-**I**nput **M**ulti-**O**utput
MPC	**M**odel **P**redictive **C**ontrol
MPP	**M**odel **P**redictive **P**lanning

NL-MPC	**N**on **L**inear - **M**odel **P**redictive **C**ontrol
NL-MPP	**N**on **L**inear - **M**odel **P**redictive **P**lanning
NRMSE	**N**ormalized **R**oot **M**ean **S**quare **E**rror
PID	**P**roportional **I**ntegral **D**erivative
RMPC	**R**obust **M**odel **P**redictive **C**ontrol
RMSE	**R**oot **M**ean **S**quare **E**rror
ROS	**R**obotic **O**perating **S**ystem
RPI	**R**obust **P**ositive **I**nvariant
RWD	**R**ear **W**heel **D**riven
SF	**S**tate **F**eedback
SMC	**S**liding **M**ode **C**ontrol
SS	**S**tate-**S**pace
SVM	**S**upport **V**ector **M**achine
TS	**T**akagi-**S**ugeno
UIO	**U**nknown **I**nput **O**bserver
UTM	**U**niversal **T**ransverse **M**ercator
VRV	**V**irtual **R**eference **V**ehicle

Symbols

A	state matrix
A_r	vehicle frontal area
B	input matrix
C	output matrix
C_d	vehicle drag coefficient
C_f	front tire stiffness coefficient
C_r	rear tire stiffness coefficient
CG	center of gravity
E	disturbance input matrix
F_{df}	drag and static friction force
F_{fr}	estimated friction force
F_{yf}	front lateral tire force
F_{yr}	rear lateral tire force
H_p	MPC prediction horizon
I	vehicle yaw inertia
K	state feedback control matrix
L	observer matrix
M	identification problem horizon
N	number of vertex systems/subsystems
P	Lyapunov matrix
Q	state weighting matrix
R	input weighting matrix
R_w	zonotope generator matrix
T_s	sampling time
U	system inputs domain

W	system disturbance domain
X	system states domain
Y	LMI-LQR problem decision matrix
a	rear wheel acceleration
c_w	center of zonotope
e	error vector
f	non-linear system
g	gravity
k	discrete-time sample
l_f	distance from CG to front axle
l_r	distance from CG to rear axle
m	vehicle mass
n	number of system inputs
p	number of measurable states
r	control reference
s	number of system states
t	continuous-time variable
u	system input / control action vector
u^*	optimal system input / control action from MPC
v	linear velocity at CG
v_w	wind velocity acting on vehicle lateral axis
v_x	longitudinal linear velocity at CG
v_y	lateral linear velocity at CG
w	system disturbance vector
x	time-domain system state vector
$\tilde{x}$	space-domain state vector
$\check{x}$	nominal state vector
$\hat{x}$	estimated state vector
y	system output vector
y_e	vehicle lateral error
α	slip angle at CG
α_f	slip angle at front wheels
α_r	slip angle at rear wheels

δ	steering angle at front wheels
Δu	time variation of input vector
Γ	pseudoinverse of CE
γ	LMI problem attenuation scalar
κ	road curvature
μ_i	i-th coefficient of a polytopic decomposition
μ	vehicle friction cefficient
ω	vehicle angular velocity
Φ	reachability domain
ρ	air density at 25 °C
χ_f	terminal state domain
ε	epsilon variable
ζ	vector of varying parameters
θ	vehicle orientation (yaw angle)
θ_e	vehicle orientation error
Θ	domain of variation of ζ
φ	longitudinal road slope

List of publications

Papers included in the thesis

The following papers are included in the thesis.

Journal papers

- Alcalá, Eugenio, et al. [2018.A]. Alcalá, E., Puig, V., Quevedo, J., Escobet, T., & Comasolivas, R. (2018). Autonomous vehicle control using a kinematic Lyapunov-based technique with LQR-LMI tuning. *Control engineering practice (CEP)*, 73, 1-12.

- Alcalá, Eugenio, et al. [2018.B]. Alcalá, E., Puig, V., Quevedo, J., & Escobet, T. (2018). Gain-scheduling LPV control for autonomous vehicles including friction force estimation and compensation mechanism. *IET Control Theory & Applications*, 12(12), 1683-1693.

- Alcalá, Eugenio, Puig, Vicenç and Quevedo, Joseba. [2019.A]. Alcalá, E., Cayuela, V. P., & Casin, J. Q. (2019). TS-MPC for Autonomous Vehicles including a TS-MHE-UIO estimator. *IEEE Transactions on Vehicular Technology.*

- Alcalá, Eugenio, Puig, Vicenç and Quevedo, Joseba. [2019.B]. Alcalá, E., Puig, V. & Quevedo, J. (2019). LPV-MP Planning for Autonomous Racing Vehicles considering Obstacles. *Robotics and Autonomous Systems*, 103392.

- Alcalá, Eugenio, et al. [2020.A]. Alcalá, E., Puig, V., Quevedo, J., & Rosolia, U. (2020). Autonomous racing using Linear Parameter Varying-Model Predictive Control (LPV-MPC). In *Control Engineering Practice*, 95, 104270.

- Alcalá, Eugenio, et al. [2020.B]. Alcalá, E., Puig, V., Quevedo, J. & Sename, O. (2019). Fast Zonotope-Tube-based LPV-MPC for Autonomous Vehicles. *The International Journal of Control*, 2019 (submitted).

Papers not included in the thesis

The following papers were published/prepared during the study, but are not included in the thesis.

Conference papers

- Alcalá, Eugenio, et al. [2016] Alcalá, E., Sellart, L., Puig, V., Quevedo, J., Saludes, J., Vázquez, D., & López, A. (2016, June). Comparison of two non-linear model-based control strategies for autonomous vehicles. In *2016 24th Mediterranean Conference on Control and Automation (MED)* (pp. 846-851). IEEE.
- Alcalá, Eugenio, et al. [2019.C] Alcalá, E., Puig, V. & Quevedo, J. (2019). LPV-MPC Control for Autonomous Vehicles. In *IFAC-PapersOnLine*, 52(28), 106-113.
- Alcalá, E., Sename, O., Puig, V. & Quevedo, J. (2020). TS-MPC for Autonomous Vehicle using a Learning Approach. In *21st IFAC World Congress*. (submitted).

Chapter 1

Introduction

1.1 Context of the thesis

The results presented in this thesis have been developed at the Research Center for Supervision, Safety and Automatic Control (CS2AC) of the Universitat Politècnica de Catalunya (UPC) in Terrassa, Spain. The research was jointly supervised by Dr. Vicenç Puig and Dr. Joseba Quevedo, and was sponsored by the Agència de Gestió d'Ajuts Universitaris i de Recerca (AGAUR) through an FI grant. The supports are gratefully acknowledged.

1.2 Motivations

The Victoria Transport Policy Institute (VTPI) presents a recent report exploring the impacts of autonomous driving in the near future [Litman, Todd., 2019]. This report highlights a rapid advance in the autonomous vehicle industry but it will not be until 2040 when we begin to appreciate most of the impacts that this technology entails: (1) reduction of accidents by taking humans out of the driving task [Morando et al., 2018, Papadoulis et al., 2019]; (2) inclusion of citizens with low physical mobility by the introduction of door-to-door transportation services; (3) reduction of congestion by route sharing (passengers and goods) and a centralized mobility intelligence; (4) decrease of energy consumption and pollution by relying on electric vehicles with a smarter vehicle control.

To achieve these benefits, many different engineering disciplines have to row in the same direction and research on innovative algorithms. Three software blocks are clearly differentiated in autonomous guidance: perception, localization and motion layers (see Figure 1.1). Motion block is composed by trajectory planning and automatic control techniques whose development has attracted a lot of attention in the automotive field during the last years, as testified by the increasing number of publications dealing with these topics.

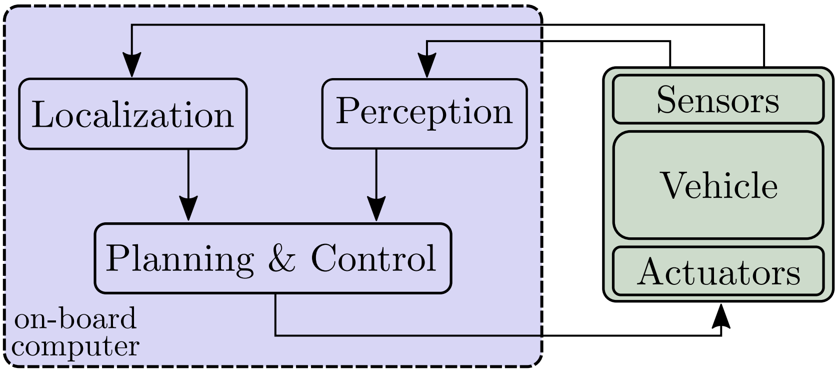


FIGURE 1.1: Overview of the different blocks playing a roll in the autonomous vehicle

The increasing need for safety and reliability has motivated the research on techniques using advanced vehicle representations for controlling both longitudinal and lateral vehicle dynamics at the same time. However, a still more motivating problem due to its high complexity is emerging in recent years. This is the autonomous racing driving [Formula Student., 2019, Roborace., 2019].

From a planning and control point of view, this last is a more challenging problem since the vehicle has to go as fast as possible without exceeding the limits of maximum overall acceleration and thus avoid slipping. Additionally, solving the racing problem requires a huge amount of calculations per second which implies the need of new algorithms able to run on embedded systems at high speed. Consequently, researching in the racing field allow us to develop more reliable algorithms for normal autonomous driving in cities and highways.

On the other hand, the huge interest on reducing the design complexity of non-linear based algorithms as well as reducing their computational load in optimization-based problems has attracted the attention of the research community. In particular, the linear parameter varying (LPV) paradigm has allowed a manner of applying linear techniques to solve non-linear problems offering guarantees of stability and performance.

1.3 Thesis objectives

The objectives of this thesis are the following:

- to state clearly the differences and connections between the current vehicle models for control purposes;
- to study the benefits of decoupling kinematics and dynamics when controlling the vehicle;
- to show how control strategies developed using the LPV representation could solve non-linear control problems with similar performance but reducing the design complexity and computational effort;
- to study the trajectory planning task for racing vehicles and propose an on-line optimal-based strategy able to avoid obstacles;
- to investigate additional techniques to the Extended Kalman filter based on a vehicle model formulated in the LPV framework;
- to study an approach for the design of robust MPC controllers for uncertain LPV systems that can guarantee some desired performance;
- to implement the developed algorithms in real platforms.

1.4 Outline of the thesis

The thesis is organized as follows:

Chapter 2 provides an overview of the most advanced projects in the autonomous vehicle field and present the state of the art of the different parts involved in the motion layer, i.e. planning, control and estimation algorithms.

Vehicle modeling part presents different vehicle model formulations for the model-based planning and control techniques presented in next chapters. It is made up of two chapters:

- **Chapter 3** presents the different non-linear vehicle representations used in this thesis. From a mass-point formulation and considering the bicycle representation, this chapter shows different kinematic, dynamic and error-based models.
- **Chapter 4** addresses the LPV modeling. In particular, the mathematical representations presented in Chapter 3 are reformulated into a LPV form by using the non-linear embedding approach. This allows to obtain a polytopic LPV model for a given non-linear system. This LPV modeling is the basis for the next planning, control and estimation approaches.

Control and estimation part presents the results that constitute a contribution to the state of the art of motion control and state estimation for autonomous vehicles. It is made up of five chapters:

- **Chapter 5** proposes a control approach for autonomous vehicles using a Lyapunov-based technique with a LQR-LMI tuning. In particular, using the kinematic model of the vehicle, a non-linear control strategy based on Lyapunov theory is proposed for solving the control problem of autonomous guidance. To optimally adjust the parameters of the Lyapunov controller, the closed-loop system is reformulated using an LPV formulation. Then, an optimization algorithm that solves the LQR-LMI problem is used to determine the controller parameters. Furthermore, the tuning process is complemented by adding a pole placement constraint that guarantees that the maximum achievable performance of the kinematic loop could be achieved by the dynamic loop. The obtained controller jointly with a trajectory generation module are in charge of the autonomous vehicle guidance. Finally, the chapter illustrates the performance of the autonomous guidance system in a virtual reality environment and in a real scenario achieving the proposed goal: to move autonomously from a starting point to a final point in a comfortable way.
- **Chapter 6** presents a solution for the integrated longitudinal and lateral control problem of urban autonomous vehicles. It is based on a GS-LPV control approach combined with the use of an UIO for estimating the vehicle states and friction force. Two GS-LPV controllers are used in cascade configuration that use the kinematic and dynamic vehicle models and the friction and observed states provided by the UIO. The LPV–UIO is designed in an optimal manner by solving a set of LMIs. On

the other hand, the design of the kinematic and dynamic controllers lead to solve separately two LPV–LQR problems formulated also in LMI form. The UIO allows to improve the control response in disturbance affected scenarios by estimating and compensating the friction force. The proposed scheme has been integrated with a trajectory generation module and tested in a simulated scenario. A comparative study is also presented considering the cases that the friction force estimation is used or not to show its usefulness.

- **Chapter 7** presents a novel approach to solve the autonomous guidance problem for racing vehicles. This approach is based on the use of a cascade control where the external loop solves the position control using a novel TS-MPC approach and the internal loop is in charge of the dynamic control of the vehicle using a TS-LQR technique designed via LMIs. Both control techniques use a TS formulation of the kinematic and dynamic models of the vehicle. In addition, a novel TS-MHE-UIO is presented. This method estimates the dynamic states of the vehicle optimally as well as the force of friction acting on the vehicle that is used to reduce the control efforts. The innovative contribution of the TS-MPC and TS-MHE-UIO techniques is that using the TS model formulation of the vehicle allows us to solve the non-linear problem as a linear optimization problem, reducing computation times by 10-20 times. To demonstrate the potential of the TS-MPC, a comparison between three methods of solving the kinematic control problem is performed: using the NL-MPC with compensated friction force, using the TS-MPC approach with compensated friction force and using TS-MPC without compensated friction force is presented.

- **Chapter 8** considers the LPV theory to represent the non-linear dynamics of the testing vehicle and implement an LPV-MPC approach that can be computed online in embedded platforms with reduced computational cost under racing behaviour. In the proposed strategy, the unknown future vehicle states for performing the prediction is a problem since the LPV model has to be instantiated. This issue is solved by using the predicted data from the last control optimization. The optimal time problem is solved by an optimal offline trajectory planner that calculates the best trajectory under the constraints of the circuit. An identification of the system model based on optimization is also carried out. The planning and control scheme is validated in simulation and experimentally in a real platform where the effectiveness of the proposed LPV-MPC is demonstrated.

- **Chapter** 9 presents an effective online robust model predictive control solution for autonomous vehicles that aims at improving the computational load while preserving high levels of robustness and performance in racing scenarios. A scheme composed by a nominal controller (tube-based LPV-MPC) and a faster corrective controller (LPV-H_∞) is proposed. The robustness is introduced by means of computing a constrained tube based on the maximum disturbance-uncertainty considered. Then, for computing the robust tube, a polytopic local controller is designed as a H_∞ controller able to reject external disturbances and a finite number of reachable sets are computed online using zonotope theory taking into account the system dynamics and the local controller. Finally, the proposed strategy is tested and the performance is compared against a current state of the art tube based MPC. It is demonstrated the effectiveness of this approach in a disturbed racing scenario being able to reject strong exogenous disturbances and fulfilling imposed constraints at a reduced computational cost.

Trajectory planning part presents a new trajectory planning algorithm that constitutes a contribution to the current state of the art. It is made up of one chapter:

- **Chapter** 10 presents the space-domain and the time-domain representations as well as their optimal formulations for trajectory planning. Then, it presents the effective online planning solution (LPV-MPP) for autonomous vehicles where the focus is on improving the computational load while preserving high levels of performance in racing scenarios. The method follows the structure of the model predictive optimal strategy where the main objective is to maximize the velocity while smoothing the dynamic behaviour and fulfilling varying constraints. In particular, the aim is on reformulating the non-linear original problem into a pseudo-linear problem by convexifying the objective function and making use of the LPV vehicle formulation. In addition, the ability of avoiding obstacles is introduced in a simple way and with reduced computational cost. Finally, the approach is tested and compared the performance of the proposed strategy against its non-linear approach through simulations. Moreover, the performance of the planning approach is tested in a racing scenario. First, in a free obstacles track and consecutive in a scenario considering static obstacles. The simulation results show the effectiveness of the proposed strategy by reducing the algorithm elapsed time while finding appropriate trajectories under several input/state constraints.

Finally, the thesis is concluded by:

- **Chapter 11**, which summarizes the main conclusions and suggests possible lines of future research;
- **Appendix A** presents the development of the reference-based kinematic model for control.

Chapter 2

Background and State of the Art

2.1 Vehicle control and state estimation

The automatic control, also known as motion control in the self-driving field, consists on following a given reference provided by the trajectory planner by taking action over a set of actuators of the vehicle such as steering motor, electric motor and brake pump. This problem is generally defined by three aspects:

- vehicle behaviour to be controlled/estimated (lateral, longitudinal or both)
- vehicle model complexity (kinematic, linear dynamic, non-linear simplified dynamic or non-linear dynamic)
- control/observer strategy

Up to now, different control problems have been treated such as the longitudinal control, the lateral control and the mixed one, that includes both cases. The goal in the longitudinal control task is to maintain the linear velocity of the vehicle around a given velocity set point. It is also known as cruise control. At this point, the driver is released of the accelerating and braking tasks, being the autonomous system the responsible. This case is included in the level 1 of automation defined by the Society of Automotive Engineers (SAE) [SAE., 2016]. The lateral control is in charge of controlling the yaw movement of the vehicle by acting over the angle of the front wheels. Unlike longitudinal control, the human driver only controls the acceleration and brake, being the automatic controller in charge of turning. The last control problem is the mixed one. In this case, the vehicle governs the complete 2D motion, i.e., full control of the accelerating, braking, and steering tasks and rises to the levels 2-5 of automation.

In turn, most of these control problems in the literature rely on a vehicle model. This problems are often called *model-based control* problems and *control-oriented model* is the name given to the model representation. A model is a set of differential equations that defines a particular behaviour such as kinematics, longitudinal dynamics, lateral dynamics, etc. In the context of autonomous guidance of autonomous vehicles, the objective of the vehicle model is to provide a relationship between physical inputs and dynamic variables of the vehicle, i.e. accelerations, velocities, etc. Vehicle models can be classified, in an increasing order of complexity, into two categories:

- Kinematic formulations, also known as mass point models, can represent the vehicle motion in a range of conditions which does not involve dynamics, assume null skidding and consider lateral force to be so small that can be neglected. They are widely used due to its low parameter dependency [Rajamani, Rajesh., 2011]. Notice that different complexity versions for the kinematic model can be found in the automotive field literature such as the mass point model and the bicycle kinematic model.

- Dynamic representations are generally composed by two subsystems most of time in the vehicle field literature. Tire modeling represents the interaction between the wheel and the road. In this model, the complexity comes in the form of nonlinear relations between slip angles-ratios and lateral-longitudinal tire forces. Vehicle modeling states the relation between lateral-longitudinal tire forces and body frame accelerations. Simplified dynamic models are often used due to their satisfactory results such as bicycle dynamic representations. However, currently, it is increasingly common to observe authors using more complex dynamic models as the 4-wheel dynamic model considering roll, pitch and vertical motions.

In Liniger, Alexander. [2018], the author performs a comparison between the bicycle and the four wheels model showing a hardly any difference between the two stating that the main difference is the longitudinal load change which may be also negligible due to the low center of gravity of racing vehicles.

The third aspect defining the autonomous guidance problem is the choice of the control strategy. This selection is often being nested with the choice of the vehicle model, i.e., a linear model will require a linear technique while a non-linear one will need a non-linear strategy. In the recent literature, several control strategies have been successfully applied to guide vehicles showing different advantages and drawbacks.

Some of the most relevant control strategies in the autonomous driving field are: Proportional-Integral-Derivative (PID), H infinity (H_∞), Fuzzy Logic (FL) control, Sliding Mode Control (SMC), Lyapunov-based control, Gain-Scheduling Linear Parameter Varying (GS-LPV), Gain-Scheduling Takagi-Sugeno (GS-TS), Linear Quadratic Regulator (LQR) and Model Predictive Control (MPC).

After describing the three general aspects defining the autonomous guidance, a classification of control strategies according to the controller behaviour and vehicle model complexity can be made. This labeling will serve in order to illustrate what kind of solutions are being more used for solving the autonomous guidance problem. Table 2.1 shows such a classification with the corresponding references.

Depending on the type of control problem that we want to solve, all of them could be a good option, however from a full autonomous point of view the mixed control problem should be considered.

The approach of using a kinematic model for achieving a mixed control strategy is a good idea for starting since offers nice results. Lyapunov theory has become a standard method for analyzing stability of non-linear systems [Dixon, Warren E., et al., 2001], but also for obtaining model-based strategies for controlling in this case longitudinal and lateral behaviours jointly. In addition, robust techniques like H_∞, SMC or MPC have been also proposed for solving the kinematic mixed problem.

However, controlling a vehicle is more complex than governing a kinematic model. For mobile robots that move at very low speeds and their mass is not so large, kinematic models may even be the best option. But, controlling a real vehicle, whose mass is larger and moves at higher speeds, is a more challenging problem. This is why having a good knowledge of vehicle dynamics is especially interesting and therefore dynamic models are required.

	Kinematic Model		
Control Strategy	Long	Lateral	Mixed
PID/PID Adaptive		Li, Hui-min, et al. [2016]	
H_∞			Nawash, Nuha. [2005]
FL		Li, Hui-min, et al. [2016], Wang, Xinyu, et al. [2015]	
SMC/SMC Adaptive			Alcalá, Eugenio, et al. [2016]
Lyapunov			Indiveri, Giovanni. [1999], Alcalá, Eugenio, et al. [2016], Blažič, Sašo. [2010], Dixon, Warren E., et al. [2001]
GS-LPV			
GS-TS			Blažič, Sašo. [2010]
LQR			
MPC	Li, Shengbo, et al. [2010]		Olsson, Christian. [2015], González, Ramón, et al. [2011]
NL-MPC	Geiger, Andreas, et al. [2012]		Farrokhsiar, Morteza et al. [2013]
LPV-MPC			

	Dynamic Model		
Control Strategy	Long	Lateral	Mixed
PID/PID Adaptive	Nie, Linzhen, et al. [2018]	Marino, Riccardo, et al. [2011], Zhao, Pan, et al. [2012]	
H_∞			
FL		Soualmi, B., et al. [2014], Zhang, Changzhu, et al. [2018]	Guo, Jinghua, et al. [2018]
SMC/SMC Adaptive	Soualmi, B., et al. [2014]	Tagne, Gilles, et al. [2013], Nam, Kanghyun, et al. [2015], Hu, Chuan, et al. [2019]	
Lyapunov	Attia, Rachid, et al. [2014]		
GS-LPV		Guo, Jinghua, et al. [2019]	Németh, Balázs, et al. [2016]
GS-TS		Soualmi, B., et al. [2014], Nguyen, Anh-Tu et al. [2016]	
LQR		Kang, Juyong, et al. [2008]	Gonzales, J., et al. [2016]
MPC		Guo, Hongyan, et al. [2019]	Olsson, Christian. [2015]
NL-MPC	Zheng, Yang, et al. [2016]	Keviczky, Tamás, et al. [2006], Gray, Andrew, et al. [2013], Attia, Rachid, et al. [2014], Besselmann, Thomas. [2010],	Gao, Yiqi, et al. [2014], Carvalho, Ashwin, et al. [2015], Liniger, A., et al. [2015] Verschueren, Robin, et al. [2016]
LPV-MPC		Besselmann, Thomas. [2010]	

Table 2.1: Classification of control techniques according to the type of model and the type of control problem considered.

In Table 2.1, it can be observed how many of the control strategies solve the problem of lateral control using the dynamic model. FL control allows the vehicle to perform in a similar way than the human brain, obtaining for lateral control satisfactory results being a simple control algorithm. The case of SMC results to obtain right results by obtaining a control law in an off-line way. However, reaching such a control law when using complex models could be a tedious task. The MPC approach is a suitable control technique that has shown interesting results for autonomous vehicle steering task. Unlike the SMC technique, it performs on an on-line way as an optimal constrained algorithm.

GS-LPV and GS-TS approaches address the control of non-linear systems using a family of linear controllers where each one guarantee stability and performance for a different operating point of the system.

However, solving the dynamic lateral control problem is not enough to fix the full autonomous vehicle problem. Thus, only by addressing the dynamic mixed control problem the purpose will be fulfilled. It can be appreciated in the dynamic mixed problem column of Table 2.1 that LPV and MPC are the only techniques applied to solve the complete autonomous vehicle problem. It may be due to the fact that the development of other techniques like Lyapunov-based or SMC may result in a more complex task.

2.2 LPV and TS systems

The LPV systems were first introduced by J. S. Shamma to distinguish these from the LTI and LTV systems [Shamma, Jeff S., 1988, 2012]. Specifically, LPV systems are defined as finite-dimensional linear time-varying plants whose state space matrices are fixed functions of some vector of varying and measurable parameters [Sename, Olivier, Peter Gaspar, and József Bokor., 2013]. LPV systems have proven to be suitable for controlling non-linear systems by incorporating non-linearities in the varying parameters, which will depend on some endogenous signals such as states, inputs or outputs.

On the other hand, the TS systems were introduced by Takagi and Sugeno in Takagi, Tomohiro, and Michio Sugeno. [1985] before the LPV systems. These provide an effective way to represent highly non-linear systems in terms of fuzzy sets and fuzzy reasoning applied to a set of linear sub-models [Cao, Yong-Yan, and Paul M. Frank., 2001].

Throughout the thesis, some control and estimation problems are approached from a TS perspective. However, given the equivalence with the LPV modeling paradigm as the author discusses in Rotondo, Damiano, et al. [2015], only the LPV methodology will be explained and referenced.

In this section, some basic concepts about modeling of LPV systems are defined. Next chapters focus mostly on control, planing and estimation techniques based on models formulated as LPV in state-space (SS). However, there exist other alternatives such as LPV input-output (LPV-IO) modeling from experimental data [Ali, Mukhtar, Hossam Abbas, and Herbert Werner., 2010, Mejari, Manas Dilip., 2018].

We dissect such modeling tasks by encompassing them in two kinds of methods: analytical and experimental methodologies. On the one hand, the analytical methods consist in obtaining an LPV formulation from a non-linear physical model of the system generally represented in state-space (LPV-SS). In this thesis, the method used for generation of LPV systems is the non-linear embedding approach [Kwiatkowski, Andreas, Marie-Theres Boll, and Herbert Werner., 2006]. However, other approaches have been addressed for the formulation of LPV models such as in Rotondo, Damiano, et al. [2015], where the concept of sector non-linearity is used.

On the other hand, the experimental methods are based on obtaining LPV models with a particular structure from input-output data (LPV-IO) of the real system [Sename, Olivier, Peter Gaspar, and József Bokor., 2013, Toth, Roland., 2010, Bachnas, A. A., et al., 2014, Mejari, Manas Dilip., 2018]. The identification problem consists in estimating recursively through a parameter adaptation algorithm the unknown parameters and the model order from the measurements of the inputs and outputs.

In addition, some recent works show that combining input and output data with the state-space representation leads to an analytical-based LPV model obtained by means of an experimental methodology. In Rizvi, Syed Zeeshan, et al. [2018], a state-space LPV identification is presented using a least-squares support vector machine (LS-SVM) algorithm. Taking advantage of analytical and experimental methodologies and following a similar learning spirit, an LPV-SS representation identified by means of using least-squares and back propagation algorithms can be obtained using Artificial Neuro Fuzzy Interference Systems (ANFIS) approach [Jang, J-SR., 1993].

An LPV system is defined as

$$\begin{aligned}\dot{x} &= A(\zeta)x + B(\zeta)u \\ y &= C(\zeta)x\end{aligned}$$

where the vector of scheduling variables $\zeta \in \mathbb{R}^{n_\zeta}$ is assumed to be measured or estimated in real-time being n_ζ the number of scheduling variables. $x \in \mathbb{R}^s$, $u \in \mathbb{R}^n$ and $y \in \mathbb{R}^p$ are the state, input, and output vectors, respectively, and $A(\zeta) \in \mathbb{R}^{s\times s}$, $B(\zeta) \in \mathbb{R}^{s\times n}$ and $C(\zeta) \in \mathbb{R}^{p\times s}$ are varying matrices of appropriate dimensions.

There exist different analysis and synthesis approaches studied to date in the literature (linear fractional transformation approach, grid-based techniques, etc [Hoffmann, Christian, and Herbert Werner., 2014]). However, one of the most popular and the one addressed in this thesis is the polytopic approach [Hoffmann, Christian, and Herbert Werner., 2014].

Then, a system is called *polytopic LPV system* when it can be represented by matrices $A(\zeta)$, $B(\zeta)$ and $C(\zeta)$, where the scheduling vector ζ ranges over a fixed polytope $\Theta = \{\zeta \in \mathbb{R}^{n_\zeta} : H_\zeta \zeta \leq b_\zeta\}$, resulting in the following representation

$$\begin{aligned}\dot{x} &= \sum_{i=1}^{N} \mu_i\left(\zeta\right)\left(A_i x + B_i u\right) \\ y &= \sum_{i=1}^{N} \mu_i\left(\zeta\right) C_i x \ ,\end{aligned} \tag{2.1}$$

where the system matrices A_i, B_i and C_i define the so-called *vertex systems* and $\mu(\cdot)$ is known as the vertex membership function given by

$$\mu_i(\zeta) = \prod_{j=1}^{n_\zeta} \xi_{ij}(\eta_0^j, \eta_1^j), \quad \forall i = 1, ..., N \ , \tag{2.2}$$

$$\begin{aligned}\eta_0^j &= \frac{\overline{\zeta_j} - \zeta_j(k)}{\overline{\zeta_j} - \underline{\zeta_j}} \\ \eta_1^j &= 1 - \eta_0^j \ ,\end{aligned} \tag{2.3}$$

where $\xi_{ij}(\eta_0^j, \eta_1^j)$ is the function that performs the N possible combinations as the following example shows for the two scheduling variables case

$$\begin{aligned} \mu_1 &= \eta_0^2 \times \eta_0^1 \ , \ \mu_2 = \eta_0^2 \times \eta_1^1 \ , \\ \mu_3 &= \eta_1^2 \times \eta_0^1 \ , \ \mu_4 = \eta_1^2 \times \eta_1^1 \ . \end{aligned} \tag{2.4}$$

In addition, next conditions must be satisfied

$$\sum_{i=1}^{N} \mu_i(\zeta) = 1, \ \ \mu_i(\zeta) \geq 0, \ \forall \zeta \in \Theta \ . \tag{2.5}$$

2.3 Vehicle planning

Every autonomous vehicle application requires of an accurate planning of the route in order to perform the desired journey. This can be either an offline or an online task, but in both cases the planner has to provide a set of references for the automatic control module, which will try to reproduce the expected behaviour acting on the real system, i.e., the vehicle.

In this thesis, three classifications are carried out for planning algorithms. These cover the time dependency, the driving mode and the solving method.

Temporary dependence is one of the most interesting classifications [Paden, B., et al., 2016]. Planning methods can be divided into two large groups: trajectory planning and path planning. Trajectory planning is given when there exists a temporary dependent position evolution, i.e. the planning strategy computes temporal-based functions (positions and velocities). Otherwise, it is referred to path planning when the route planned is not time dependent, i.e. only lateral position and orientation are considered.

Another classification is depending on the behavioral mode: normal driving and racing driving. Racing mode refers to a planning based on taking the vehicle to its dynamic limits while normal planning refers to highways and city trajectory planning under a normal driving mode.

2.3.1 Planning for normal diving

In the literature [Katrakazas, C., et al., 2015, Paden, B., et al., 2016, González, David, et al., 2015], planning algorithms are clustered in four families according to the methodology: graph-search-based methods, curves-based methods, sampling-based techniques and optimal-based approaches.

- **Graph-search-based planning.** These strategies discretize the configuration space of the vehicle and represent it in the form of a graph. Then, using particular algorithms such as Dijkstra or A* find the shortest path in the graph. The solution does not take into account dynamic aspects of the vehicle and high precision in the solution requires a high computational cost being not recommended for real-time applications. Generally, these methods are more oriented to mobile robotics and belong to the path planning group.

- **Curves-based planning.** These algorithms initially require way-points that describe a global road map. Then they try to find the curve that best approximates the way-points by ensuring comfort, continuous vehicle accelerations and some other parameters in order to compute the trajectory. Clothoid, polynomial, Bézier and spline curve-based methods are probably the most common in the autonomous guidance field and, generally, belong to the trajectory planning group. The main advantage of this planning group is its low computational cost. In Elbanhawi, Mohamed, Milan Simic, and Reza N. Jazar. [2014], a trajectory planning approach based on Bézier splines is presented. In Talamino, J. P., et al. [2019], authors propose different trajectory planning approaches based on quintic splines.

- **Sampling-based planning.** This family of techniques appear to give solution to the expensive search in wide spaces where other algorithms are not capable. The approach consists on randomly sampling the configuration space looking for connectivity. These techniques have gained a lot of interest in the autonomous vehicle field during the last years. The most used strategies are the Probabilistic Road-map Method (PRM) and the Rapidly-exploring Random Tree (RRT). The RRT algorithm [LaValle, Steven M., 1998] was proposed by La Valle as an efficient method for finding feasible trajectories for high-dimensional non-holonomic systems. The rapid exploration is achieved by taking a random sample from the free configuration space and extending the tree in the direction of the random sample. This strategy has been very used in mobile robots tasks and now it is being introduced in autonomous vehicles field. Due to the lack of smoothness of conventional RRT method, different approaches have been developed to find feasible but also smooth trajectories. An interesting strategy is to use an RRT that is expanded by considering not only the vehicle dynamic model, but also the controller [Kuwata, Yoshiaki, et al., 2009]. It runs a forward simulation by computing control actions in order to predict where the vehicle will be. Despite of the good results of RRT and its approaches it has been demonstrated to converge to a non-optimal solution. Due to the lack of optimality, a variant that includes the solution to such a drawback is presented in Karaman, Sertac, and Emilio Frazzoli. [2010], which was called RRT*. In Karaman, Sertac, and Emilio Frazzoli. [2011], an analysis of the behaviour of Probabilistic Road Maps method (PRM) and RRT is made. The main contribution of this work is the introduction of PRM* and RRT* which have demonstrated to be optimal although at a higher computational cost. Recently, Fast Marching Tree (FMT*) [Janson, Lucas, and Marco Pavone., 2016] has been proposed as an asymptotically optimal and faster alternative to PRM* and RRT*. Authors in Schmerling, Edward, Lucas Janson, and Marco Pavone. [2015] propose differential versions of PRM* and FMT* and prove asymptotic optimality of the algorithms for control affine dynamical systems, a class that includes models of non-slipping wheeled vehicles.

- **Optimal-based planning.** The optimal methods consist of minimizing or maximizing a cost function subject to different constraints such as vehicle dynamics, bounded vehicle variables, road limits and even obstacles.

  Despite referring to these methods in González, David, et al. [2015], authors do not name the application of predictive-based optimal technique to solve the optimization problem. At the same time, they talk about certain disadvantages such as high time consumption in each optimization as well as the dependence of global waypoints. In Katrakazas, C., et al. [2015], authors do not even talk about predictive optimal planning since they allude to real-time motion planning algorithms.

  In Liu, C., et al. [2017], the authors propose a path planning strategy based on non-linear Model Predictive Control (MPC) for the case of driving in highways. A unicycle kinematic model is used for predicting future vehicle states and the approach ensures safety by being able to generate free collision trajectories. In Plessen, Mogens Graf, et al. [2017], a trajectory planning strategy is presented using linear programming tools. This approach combines kinematic-based modeling with obstacle avoidance to provide a solution to highway planning problem. In a similar way, a trajectory planning strategy based on solving a non-linear optimization is presented in Hegedüs, F., et al. [2017], where a lane change methodology to be employed in highways is proposed.

In the literature, we can also find some recent works where the problem of planning in cities or places with a complex environment, i.e. moving obstacles, is solved. This scenario is one of the most challenging because the ability to deal with a variety of mobile obstacles is one of the main skills included in the planner. In Ahmadi Mousavi, M., et al. [2018], a path planning method is introduced which is based on solving a LTV MPC using a kinematic formulation of the vehicle. Besides, this is able to deal with multiple obstacles by means of a convex formulation.

2.3.2 Planning for racing driving

Autonomous vehicle racing is a variant of the field of autonomous driving that is attracting many researchers in recent years given the challenge that supposes [Formula Student., 2019, Roborace., 2019]. Such a problem involves a complex interaction with the environment due to the fast vehicle dynamic variation, i.e. high linear and angular accelerations, what implies directly a short reaction time in certain situations.

Starting from the premise that the four families of planners presented in Section 2.3.1 have interesting strategies and have achieved very good results recently, in this thesis we will focus attention on optimization based strategies for various reasons:

- obtaining optimal solutions (local or global)
- possibility of taking into account vehicle dynamics
- ability to introduce obstacles as constraints
- restriction of dynamic variables and their time derivative

The way of guiding a vehicle as quickly as possible requires a planning algorithm able to get certain road information (e.g. track curvature in the next meters) from the environment perception layer (online information) or a previously mapped track (offline information). The resultant trajectory must not only remain inside the track limits but also fulfill the dynamic constraints such as velocity, acceleration and slip angle limits. While respecting these constraints, the trajectory should also minimize the time to reach the end of the track as well as the difference between front and rear slip angles to avoid understeering and oversteering situations. Additionally, the trajectory should be updated online as the vehicle progresses along the track with the aim of avoiding unexpected static and/or dynamic obstacles.

The resolution of this problem is not trivial at all and there are few studies under this topic, being the main motivation for its research. The objective of this type of planners is to find the optimal trajectory while maximizing the speed or minimizing the lap time. One of the main conditions of racing planning is to accurately consider the dynamics of the vehicle in the algorithm calculations. In this way, one common strategy is to handle the vehicle acceleration vector as well as the front and rear wheels slip angles under an optimization problem to find their optimal values.

In Caporale, Danio, et al. [2018], the problem of trajectory planning for a racing application is solved by means of two stages. The first solves a convex optimization that consists in minimizing the length and curvature of the path using a geometric representation. This optimization results in the optimal trajectory. Then, obtaining the curvature of such an optimal trajectory and using an equation that relates the curvature, speed and the maximum lateral force tires can sustain, then the optimum velocity profile is obtained.

In Alrifaee, B., et al. [2018], a real-time MPC for racing trajectory planning is presented. The key point of this planning approach resides in reformulating the initial non-convex problem into a linearly constrained convex quadratic optimization problem (QP) that can be solved in real-time. A point mass kinematic model is used for predicting future states while constraining vehicle accelerations.

In Verschueren, Robin, et al. [2014], the authors present a racing planning strategy based on non-linear MPC using a kinematic representation of the vehicle. Years later, the same authors enhance this version by introducing a dynamic vehicle model between other improvements [Verschueren, Robin, et al., 2016].

According to Liniger, A., et al. [2015], the authors propose a racing path planning strategy based on linear MPC strategy. This optimization procedure uses a linearized dynamic model of the vehicle which allows the real-time implementation. The paper presents the results with and without obstacles demonstrating in both cases a high performance.

The weaknesses that all the previous works have in common, and that this thesis addresses, are the use of models of reduced complexity (kinematic-based models basically) and/or the use of a path-based planner formulation.

At this point, a classification within the techniques based on optimization can be made. Then, a summary of optimal-based approaches for normal and racing planning is presented in Table 2.2. This sorting depends mainly on three variables: time-dependent trajectory, type of vehicle modeling and behavioral driving mode (scenario).

TABLE 2.2: Optimal planning works depending on the type of vehicle prediction model and the scenario. TP and PP refer to trajectory and path planning, respectively

Scenes \ Models	**L. Kin.**	**NL. Kin**
Highways		
Highways & obs		Liu, C., et al. [2017]$_{PP}$
City & obs	Ahmadi Mousavi, M., et al. [2018]$_{PP}$ Plessen, Mogens Graf, et al. [2017]$_{TP}$	
Racing	Alrifaee, B., et al. [2018]$_{TP}$ Caporale, Danio, et al. [2018]$_{TP}$	Verschueren, Robin, et al. [2014]$_{TP}$
Racing & obs		
Scenes \ Models	**L. Dyn.**	**NL. Dyn.**
Highways		Hegedüs, F., et al. [2017]$_{TP}$
Highways & obs		
City & obs		
Racing	Liniger, A., et al. [2015]$_{PP}$	Verschueren, Robin, et al. [2016]$_{TP}$
Racing & obs	Liniger, A., et al. [2015]$_{PP}$	

Part I

Vehicle modeling

Chapter 3

Control-oriented modeling

3.1 Introduction

Nowadays, some of the most advanced control, planning and estimation techniques are based on differential mathematical models that describe some mechanical properties of the road vehicle. These operational properties are the result of dynamic interactions of the different components conforming the vehicle structure, i.e. chassis, wheels, tires, etc. Playing a major role the pneumatic tire.

Choosing an appropriate model is sometimes a relevant and critical task when controlling a system. In particular, the vehicle modeling choice will depend on the current scenario-vehicle and the control-planning-estimation technique that is going to be used. For example, if the proposed scenario consists on controlling a vehicle at low speeds applications then, the mass-point kinematic models have demonstrated to achieve good enough results in the past [Alcalá, Eugenio, et al., 2018.A, Broggi, Alberto, et al., 2012, Tzafestas, Spyros G., 2013].

However, situations where the task consists on controlling, planning and/or estimating variables for an autonomous racing car require a vehicle model that represents accurately the real dynamic behaviour [Caporale, Danio, et al., 2018, Brunner, Maximilian, et al., 2017, Verschueren, Robin, et al., 2016, Liniger, A., et al., 2015].

In this chapter, vehicle modeling is discussed while gradually increasing its complexity, thereby allowing the presentation of a set of vehicle models in terms of a wide range of application. Fig. 3.1 shows a clear overview of the different physics-based formulations used in this thesis for autonomous driving strategies. Red block represents the kinematic formulations. Green rectangle depicts dynamic model representations. Blue block refers to extended model representations as a combination of red and green blocks.

The wide range of applications goes from driving at low speeds to racing environments. Special attention will be given in next sections to discuss some properties of the models from a mathematical and control point of view.

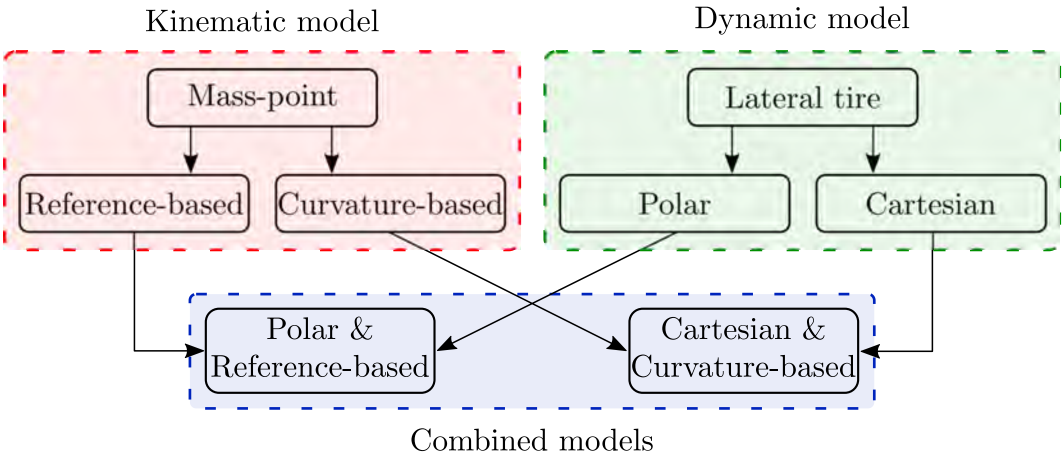


FIGURE 3.1: Tree diagram of different physics-based models used for control, planning and estimation purposes

For simplicity, the non-linear models presented below are continuous-time (CT) systems of the form

$$\dot{x}(t) = g\left(x(t), u(t)\right) \ . \tag{3.1}$$

Note that, the majority of the physical systems of interest for control purposes are expressed in CT. If the objective is an implementation in an embedded system, probably the design strategies will require a discrete-time (DT) modeling. Such models can be obtained from CT models using discretization techniques, such as Euler or more sophisticated approaches [Kazantzis, Nikolaos, et al., 2005].

3.2 Kinematic model

The known as kinematic model provides a simple mathematical description of the vehicle movement. The equations that govern the system are a set of trigonometric relationships that describe its motion. There are several variants when representing the kinematic model mathematically. In this thesis, the "mass-point" representation has been considered as the kinematic model. However, other portrayals such as the "bicycle" representation are adopted in the literature for kinematic models [Kong, Jason, et al., 2015, Rajamani, Rajesh., 2011]. Under the assumption of driving at low speed (less than 5 $\frac{m}{s}$), the vehicle can assume null skidding and consider lateral forces to be so small that can be neglected. Basically, this model represents the transformation of a vehicle velocity vector represented in Polar coordinates into a vehicle Cartesian frame located at the center of gravity (CG) of the vehicle. The mass-point kinematic equations are introduced below:

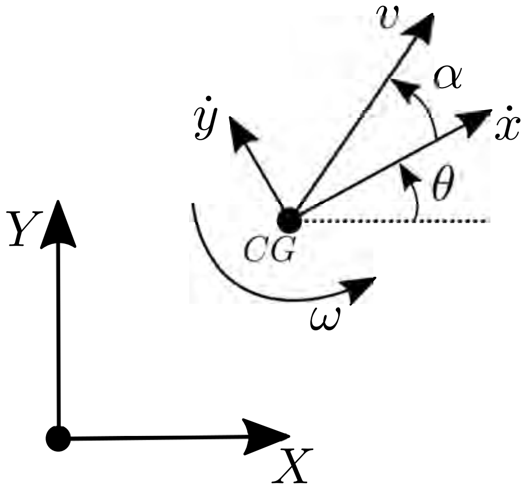


FIGURE 3.2: Mass-point kinematic representation

$$\begin{cases} \dot{x} = v\cos(\theta + \alpha) \\ \dot{y} = v\sin(\theta + \alpha) \\ \dot{\theta} = \omega \end{cases}, \tag{3.2}$$

where $\dot{x}$, $\dot{y}$ and $\dot{\theta}$ represent the longitudinal, lateral and angular velocities, respectively, in the body frame (see Figure 3.2). The velocity vector at the CG is denoted by v which performs an α angle with the longitudinal velocity vector ($\dot{x}$) called vehicle's CG slip angle. Besides, angle θ represents the orientation of the vehicle. X and Y describe the vehicle's position in the inertial coordinate frame. The kinematic model motion is performed by controlling the magnitudes of the velocity vector (v) and the angular velocity (ω).

Its high simplicity and low dependence on parameters makes this representation very useful for low longitudinal and angular velocity applications. Note that, from the control perspective, the mass-point kinematic model is singular at $v = 0$ and therefore not controllable at this point.

3.2.1 Reference-based kinematic model

Considering the uncontrollability characteristic of the kinematic model (3.2), this section focuses on the development of a representation controllable at $v = 0$. Introducing the idea of virtual reference vehicle (VRV), and using the real car for comparison, allows to generate useful error variables in position and orientation. Under this idea, a reference-based model is presented. This model variation is defined as the difference between real vehicle position and orientation (x, y and θ) and desired position and orientation (x_d, y_d and θ_d). However, this set of errors are expressed with respect to the inertial global frame XY (see Figure 3.2). For control purposes is convenient to express the errors in the vehicle frame such that the lateral error is always measured in the lateral axis of the vehicle. Thus, a rotation over the road orthogonal axis is considered as

$$\begin{bmatrix} x_e \\ y_e \\ \theta_e \end{bmatrix} = \begin{bmatrix} \cos(\theta) & \sin(\theta) & 0 \\ -\sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_d - x \\ y_d - y \\ \theta_d - \theta \end{bmatrix} , \tag{3.3}$$

where subindexes d and e represent desired and error values, respectively. To develop the reference-based model, it is necessary to impose the non-holonomic constraint of the form

$$\dot{x}\sin(\theta) = \dot{y}\cos(\theta) \ . \tag{3.4}$$

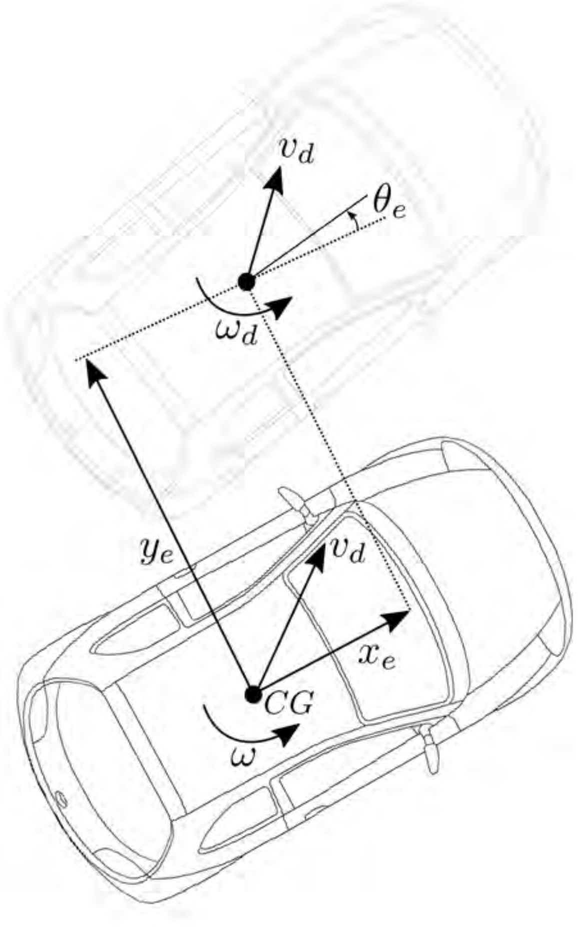


FIGURE 3.3: Reference-based kinematic representation. VRV vehicle, real vehicle below

Hence, computing the time derivative of (3.3) and using (3.2), (3.4) and some trigonometric identity, we obtain the following open-loop error system

$$\begin{aligned} \dot{x}_e &= \omega y_e + v_d \cos\theta_e - v \\ \dot{y}_e &= -\omega x_e + v_d \sin\theta_e \qquad , \\ \dot{\theta}_e &= \omega_d - \omega \end{aligned} \tag{3.5}$$

where v_d refers to the desired linear velocity, ω_d to the desired angular velocity, x_e, y_e and θ_e are the longitudinal, lateral and heading errors, respectively, represented in the vehicle frame. Details about the development of (3.5) are presented in Appendix A.

3.2.2 Curvature-based kinematic model

For control purposes, a kinematic model represented in the curvilinear coordinate system is obtained as the error of heading angle and lateral position with respect to the center line of the track [Verschueren, Robin, et al., 2014]. Observing Figure 3.4, obtaining two different definitions for vector v_{x_c} and matching them, the following constraint can be stated

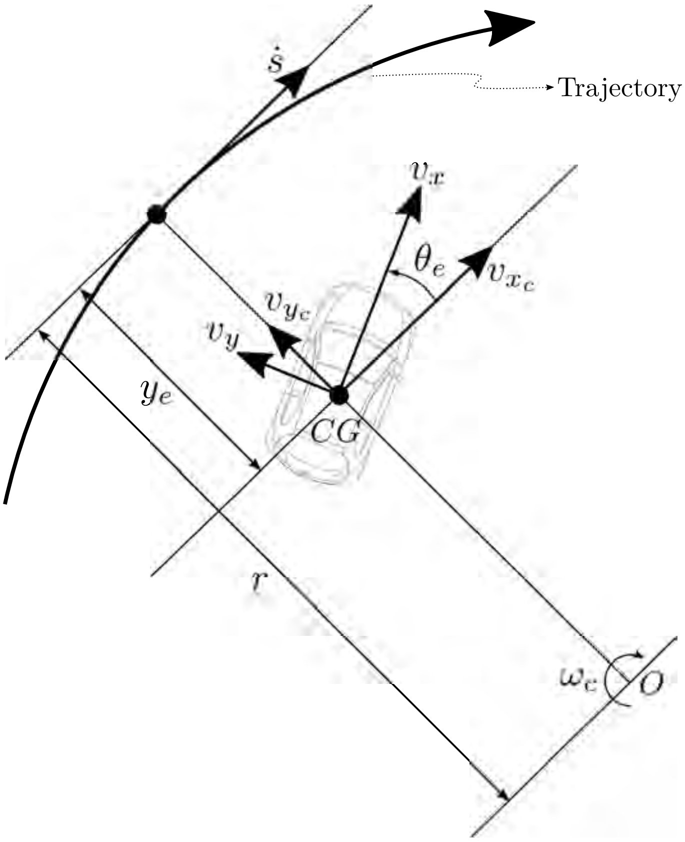


FIGURE 3.4: Curvature-based kinematic representation

$$(r - y_e)\omega_c = v_x \cos\theta_e - v_y \sin\theta_e \ , \tag{3.6}$$

where v_x and v_y are the longitudinal and lateral linear speeds, respectively, ω_c is the desired rotation speed with rotation center O and r is the radius of the curve. Consequently, defining the road curvature κ as the inverse of the radius r, the rate of change of the desired heading angle is expressed as

$$\omega_c = \frac{v_x \cos\theta_e - v_y \sin\theta_e}{1 - y_e\kappa}\kappa \ . \tag{3.7}$$

Then, the orientation error dynamics can be defined as

$$\dot{\theta}_e = \omega - \frac{v_x \cos\theta_e - v_y \sin\theta_e}{1 - y_e\kappa}\kappa \ . \tag{3.8}$$

Using (3.7), the time derivative of the projected travelled distance along the trajectory ($\dot{s}$) is defined as follows

$$\dot{s} = r\omega_c = \frac{v_x \cos\theta_e - v_y \sin\theta_e}{1 - y_e\kappa} \ . \tag{3.9}$$

Regarding the lateral error, note that, v_{y_c} corresponds with the time derivative of y_e.

Finally, the curvature-based model is given as

$$\begin{aligned} \dot{y}_e &= v_x \sin\theta_e + v_y \cos\theta_e \\ \dot{\theta}_e &= \omega - \frac{v_x \cos\theta_e - v_y \sin\theta_e}{1 - y_e\kappa}\kappa \ . \\ \dot{s} &= \frac{v_x \cos\theta_e - v_y \sin\theta_e}{1 - y_e\kappa} \end{aligned} \tag{3.10}$$

Note that, κ is a function of s and assumed to be known. Like the reference-based model, the curvature-based representation is also controllable from a control design point of view.

Note also that, this model does not include the longitudinal error (x_e) as in (3.5) due to a particular interest of control in this work.

3.2.3 From vehicle body to inertial coordinates

Both reference-based (3.5) and curvature-based (3.10) model variables are based on vehicle body frame. Transforming this representation to the inertial (or global) coordinates frame (XY in Figure 3.2) may be of interest. The following expressions allow such a conversion

$$\begin{aligned} X &= x_d + x_e \cos\theta - y_e \sin\theta \\ Y &= y_d + y_e \cos\theta + x_e \sin\theta \ . \end{aligned} \tag{3.11}$$

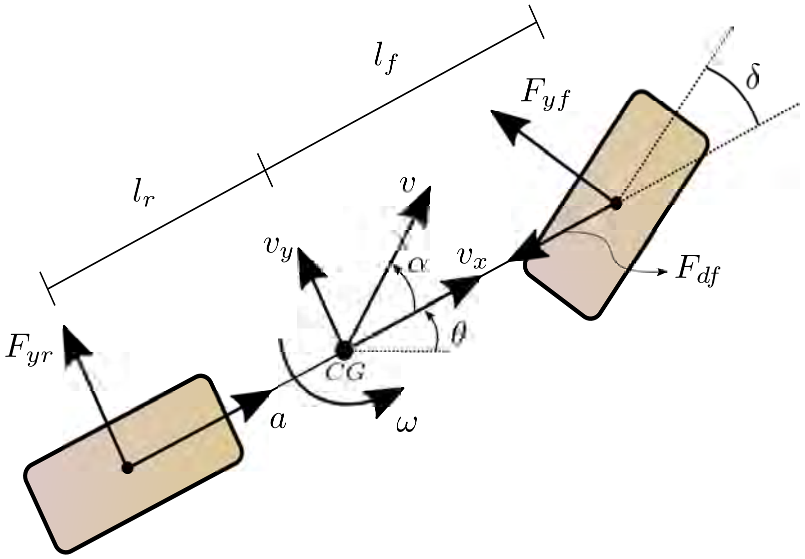


FIGURE 3.5: Bicycle dynamic model representation

3.3 Dynamic model

Unlike kinematic model, the dynamic representation is characterized by using forces for performing movement. In this thesis, a bicycle representation of the vehicle is considered when using a dynamic model (see Figure 3.5).

There are several differences between a mass-point kinematic model and a dynamic bicycle model. The dynamic model is more appropriate for working at higher speeds due to the non-assumption of negative slip angles on the front and rear wheels. The slip angle is defined as the difference between the orientation of the speed vector of a wheel and the orientation of the wheel. In addition, vehicle dynamic modeling generally includes the tire model which is of special interest in the study of vehicle dynamics and becomes relevant when studying the behaviour of a vehicle.

At this point, we distinguish among two ways of representing the dynamic model: Cartesian and Polar representations (see Figure 3.1).

Note that, throughout this thesis, the vehicle motion is performed by applying an acceleration vector in the axis of rear wheel (a) and actuating on the steering angle (δ).

3.3.1 Lateral tire model

The lateral tire model represents the lateral dynamics of the tire which is of special interest in the automotive field since this is the only element of the vehicle in contact with the road surface. The lateral force in a tire is a function of the slip angle. The expression of the slip angles on the front and rear wheels represented in the Cartesian frame is respectively

$$\begin{aligned} \alpha_f &= \delta - \tan^{-1}\left(\frac{v_y}{v_x} - \frac{l_f\omega}{v_x}\right) \\ \alpha_r &= -\tan^{-1}\left(\frac{v_y}{v_x} + \frac{l_r\omega}{v_x}\right) \end{aligned}, \tag{3.12}$$

where l_f, l_r are the distances from CG to front and rear tires, respectively and δ is the steering angle of the front wheel.

There are various representations for this model. In this thesis, we present two: a linear formulation for small slip angles situations and the simplified Magic Formula [Pacejka, Hans., 2005] representing the real tire force curve (see Figure 3.6).

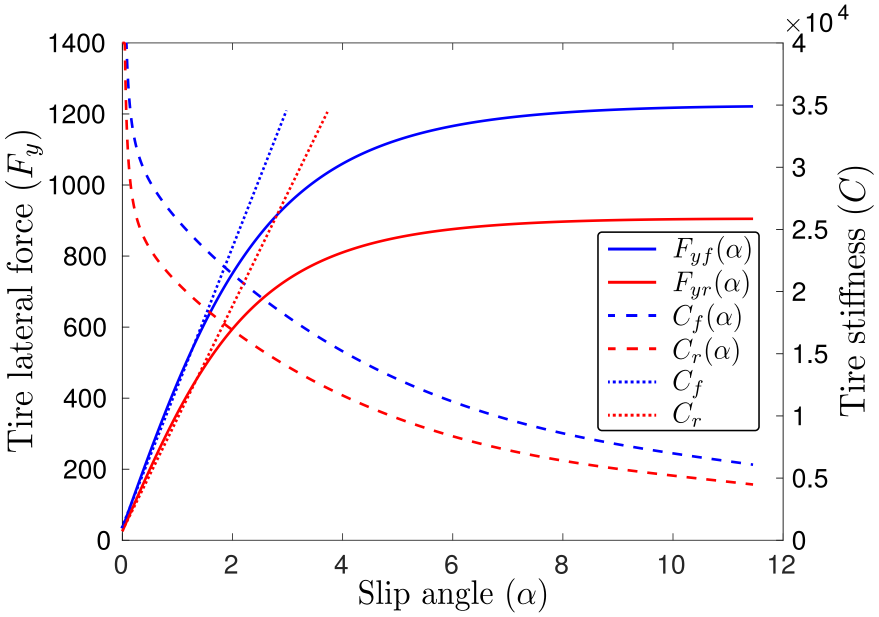


FIGURE 3.6: Tires lateral force and tires stiffness coefficients as functions of the slip angle. Force in Newtons and angle in degrees

- **Lateral tire model at small slip angles**.

  Several experiments show that for small slip angles, the lateral tire model can be considered as a linear function where the lateral force is proportional to the slip angle

$$\begin{aligned} F_{yf} &= C_f \alpha_f \\ F_{yr} &= C_r \alpha_r \end{aligned} , \tag{3.13}$$

where C_f, C_r are the tire cornering stiffness coefficients of the front and rear tires (dotted lines in Figure 3.6), respectively and F_{yf}, F_{yr} are the front and rear lateral tire forces, respectively.

- **Simplified Magic Formula tire model**.

  The linear model can be interesting when the slip angle variable remains small. However, in situations where this does not occur and such variable becomes larger a more sophisticated model is required. The simplified Magic Formula tire model provides a non-linear approach to calculate the resulting lateral force from a wide range of slip angles (solid curves in Figure 3.6). Then, the lateral force is calculated as follows

$$\begin{aligned} F_{yf} &= F_{zf} d \sin\left(c \tan^{-1}(b\alpha_f)\right) \\ F_{yr} &= F_{zr} d \sin\left(c \tan^{-1}(b\alpha_r)\right) \end{aligned} , \tag{3.14}$$

  where F_{zf}, F_{zr} are the vertical load on the front and rear wheels, respectively, d represents the peak value of the curve, c is the shape factor and b is the stiffness factor. To learn how these parameters are adjusted see Chapter 4 of Pacejka, Hans. [2005].

3.3.2 Cartesian dynamic model

Consider the inertial accelerations of the vehicle as

$$\begin{aligned} a_x &= \dot{v}_x - \omega v_y \\ a_y &= \dot{v}_y + \omega v_x \end{aligned} \tag{3.15}$$

at the CG in the direction of x and y axes. Applying Newton's second law and considering the respective trigonometric functions due to the planar motion, the Cartesian dynamic model is given by

$$\begin{aligned}\dot{v}_x &= a - \frac{F_{yf}\sin\delta}{m} - \frac{F_{df}}{m} + \omega v_y\\ \dot{v}_y &= \frac{F_{yf}\cos\delta}{m} + \frac{F_{yr}}{m} - \omega v_x \qquad ,\\ \dot{\omega} &= \frac{F_{yf} l_f \cos\delta - F_{yr} l_r}{I}\end{aligned} \tag{3.16}$$

where F_{yf} and F_{yr} are computed using either (3.13) or (3.14) and the sum of friction and drag forces is formulated as

$$F_{df} = \frac{1}{2} C_d \rho A_r (v_x)^2 + \mu m g \quad , \tag{3.17}$$

where C_d is the aerodynamic drag coefficient, ρ is the mass density of air, A_r is the vehicle frontal area, m is the vehicle mass, g is the gravity and μ represents the static friction coefficient.

3.3.3 Polar dynamic model

The following formulation of the dynamical model is the Polar representation. Unlike Cartesian representation, this presents a more complex formulation. However, Polar representation matches better with the kinematic reference-based model (3.5) since they have related states and inputs, e.g. the velocity (v) is a kinematic control action which is also an state of the dynamic vehicle model. The streamlined Polar dynamical model of the road vehicle can be written as

$$\begin{aligned}\dot{v} &= a\cos\alpha + \frac{F_{yf}\sin(\alpha-\delta) + F_{yr}\sin\alpha - F_{df}}{m} + \omega v \sin\alpha\\ \dot{\alpha} &= -a\frac{\sin\alpha}{v} + \frac{F_{yf}\cos(\alpha-\delta) + F_{yr}\cos\alpha}{mv} - \omega(1 + v\cos\alpha) \;\cdot\\ \dot{\omega} &= \frac{F_{yf} l_f \cos\delta - F_{yr} l_r}{I}\end{aligned} \tag{3.18}$$

Remark 3.1. Note that, although the slip angles on the front and rear wheels are the same in both Cartesian and Polar representation, the mathematical formulation that allows their calculation in Cartesian coordinates (3.12) differs from the Polar one being

$$\alpha_f = \delta - \alpha - \frac{l_f \omega}{v\cos\alpha} \tag{3.19}$$

$$\alpha_r = -\alpha + \frac{l_r \omega}{v\cos\alpha} \; . \tag{3.20}$$

Thus, forces in Polar coordinates frame are computed as

$$F_{yf} = C_f \alpha_f \tag{3.21}$$

$$F_{yr} = C_r \alpha_r \tag{3.22}$$

$$F_{df} = \frac{1}{2} C_d \rho A_r (v \cos\alpha)^2 + \mu m g \ . \tag{3.23}$$

3.4 Combined kinematic and dynamic models

The purpose of combining dynamic and kinematic models is to generate a complete vehicle model for control, planning and estimation strategies.

3.4.1 Polar dynamic and Reference-based model (PDRB)

Introducing into the same model both the Polar dynamic model (3.18) using the linear formulation of the tire model (3.13) and the reference-based kinematic model (3.5), the following state-space representation is obtained

$$\dot{x} = \begin{bmatrix} \dot{v} \\ \dot{\alpha} \\ \dot{\omega} \\ \dot{x_e} \\ \dot{y_e} \\ \dot{\theta}_e \end{bmatrix} = f(x, u, v_d, \omega_d) \tag{3.24}$$

$$= \begin{bmatrix} a\cos\alpha + \dfrac{\left(C_f\left(\delta{-}\alpha{-}\frac{l_f\omega}{v\cos\alpha}\right)\right)\sin(\alpha{-}\delta){+}\left(C_r\left({-}\alpha{+}\frac{l_r\omega}{v\cos\alpha}\right)\right)\sin\alpha{-}F_{df}}{m} + \omega v \sin\alpha \\ -a\dfrac{\sin\alpha}{v} + \dfrac{\left(C_f\left(\delta{-}\alpha{-}\frac{l_f\omega}{v\cos\alpha}\right)\right)\cos(\alpha{-}\delta){+}\left(C_r\left({-}\alpha{+}\frac{l_r\omega}{v\cos\alpha}\right)\right)\cos\alpha}{mv} - \omega(1 + v\cos\alpha) \\ \dfrac{\left(C_f\left(\delta{-}\alpha{-}\frac{l_f\omega}{v\cos\alpha}\right)\right)l_f\cos\delta{-}\left(C_r\left({-}\alpha{+}\frac{l_r\omega}{v\cos\alpha}\right)\right)l_r}{I} \\ \omega y_e + v_d\cos\theta_e - v \\ -\omega x_e + v_d \sin\theta_e \\ \omega_d - \omega \end{bmatrix} .$$

3.4.2 Cartesian dynamic and Curvature-based model (CDCB)

Putting into a same model representation the Cartesian dynamic model (3.16) using the Magic Formula tire model (3.14) and the curvature-based kinematic model (3.10) the following combined model is obtained

$$\dot{x} = \begin{bmatrix} \dot{v_x} \\ \dot{v_y} \\ \dot{\omega} \\ \dot{y_e} \\ \dot{\theta}_e \\ \dot{s} \end{bmatrix} = f(x, u, \kappa) \tag{3.25}$$

$$= \begin{bmatrix} a - \frac{\sin\delta F_{zf} d \sin\left(c\tan^{-1}\left(b\delta - b\tan^{-1}\left(\frac{v_y}{v_x} - \frac{l_f\omega}{v_x}\right)\right)\right)}{m} - \mu g + \omega v_y \\ \frac{F_{zf} d \cos\delta \sin\left(c\tan^{-1}\left(b\delta - b\tan^{-1}\left(\frac{v_y}{v_x} - \frac{l_f\omega}{v_x}\right)\right)\right)}{m} + \frac{F_{zr} d \sin\left(c\tan^{-1}\left(-b\tan^{-1}\left(\frac{v_y}{v_x} + \frac{l_r\omega}{v_x}\right)\right)\right)}{m} - \omega v_x \\ \frac{l_f F_{zf} d \cos\delta \sin\left(c\tan^{-1}\left(b\delta - b\tan^{-1}\left(\frac{v_y}{v_x} - \frac{l_f\omega}{v_x}\right)\right)\right) - l_r F_{zr} d \sin\left(c\tan^{-1}\left(-b\tan^{-1}\left(\frac{v_y}{v_x} + \frac{l_r\omega}{v_x}\right)\right)\right)}{I} \\ v_x \sin\theta_e + v_y \cos\theta_e \\ \omega - \frac{v_x \cos\theta_e - v_y \sin\theta_e}{1 - y_e\kappa}\kappa \\ \frac{v_x \cos\theta_e - v_y \sin\theta_e}{1 - y_e\kappa} \end{bmatrix}.$$

3.4.3 CDCB model in space domain representation

Model (3.25) is convenient for control purposes due to its time-domain representation. However, optimal time racing focuses on minimizing lap time and thus, the time variable needs to be converted into an optimization variable. For that reason, we propose using the model reparametrization from Gao, Yiqi, et al. [2012]. Such a new formulation is based on the space domain and the road curvature is given as a function of s hence, the trajectory optimization under the space-based domain becomes feasible. On the contrary, using a time-based domain would not be possible to solve a predictive optimization problem like the one presented in next section because time would not appear explicitly as an optimization variable.

Denoting $x_c = \begin{bmatrix} v_x & v_y & \omega & y_e & \theta_e & s \end{bmatrix}^T$ as the state vector of the control model (3.25), then, a new state vector $\tilde{x_c} = \begin{bmatrix} \tilde{v_x} & \tilde{v_y} & \tilde{\omega} & \tilde{y_e} & \tilde{\theta_e} & \tilde{t} \end{bmatrix}^T$ is obtained by applying ($\tilde{(\cdot)}$ denotes a variable in the space domain)

$$\dot{\tilde{x}}_c = \frac{dx_c}{ds} = \frac{dx_c}{dt}\frac{dt}{ds} = \dot{x}_c\frac{1}{\dot{s}} = \dot{x}_c\dot{\tilde{t}} \ , \tag{3.26}$$

leading consequently to the following model equations

$$\begin{aligned} \dot{\tilde{y_e}} &= \frac{\tilde{v_x}\sin\tilde{\theta}_e + \tilde{v_y}\cos\tilde{\theta}_e}{\dot{s}} \\ \dot{\tilde{\theta}}_e &= \frac{\tilde{\omega}}{\dot{s}} - \kappa \\ \dot{\tilde{t}} &= \frac{1}{\dot{s}} \ . \end{aligned} \tag{3.27}$$

Note the time ($\tilde{t}$) is now in the space domain and hence, as function of s. The explicit model equations are given by

$$\dot{\tilde{x}} = \begin{bmatrix} \dot{\tilde{v_x}} \\ \dot{\tilde{v_y}} \\ \dot{\tilde{\omega}} \\ \dot{\tilde{y_e}} \\ \dot{\tilde{\theta}}_e \\ \dot{\tilde{t}} \end{bmatrix} = f(\tilde{x}, u, \kappa) \tag{3.28}$$

$$= \begin{bmatrix} \dfrac{a - \frac{\sin\delta\, d\sin\left(c\tan^{-1}\left(b\delta - b\tan^{-1}\left(\frac{\tilde{v_y}}{\tilde{v_x}} - \frac{l_f\omega}{\tilde{v_x}}\right)\right)\right)}{m} - \mu g + \tilde{\omega}\tilde{v_y}}{\dot{s}} \\ \dfrac{\frac{d\cos\delta\sin\left(c\tan^{-1}\left(b\delta - b\tan^{-1}\left(\frac{\tilde{v_y}}{\tilde{v_x}} - \frac{l_f\tilde{\omega}}{\tilde{v_x}}\right)\right)\right)}{m} + \frac{d\sin\left(c\tan^{-1}\left(-b\tan^{-1}\left(\frac{\tilde{v_y}}{\tilde{v_x}} + \frac{l_r\tilde{\omega}}{\tilde{v_x}}\right)\right)\right)}{m} - \tilde{\omega}\tilde{v_x}}{\dot{s}} \\ \dfrac{\frac{l_f d\cos\delta\sin\left(c\tan^{-1}\left(b\delta - b\tan^{-1}\left(\frac{\tilde{v_y}}{\tilde{v_x}} - \frac{l_f\tilde{\omega}}{\tilde{v_x}}\right)\right)\right) - l_r d\sin\left(c\tan^{-1}\left(-b\tan^{-1}\left(\frac{\tilde{v_y}}{\tilde{v_x}} + \frac{l_r\tilde{\omega}}{\tilde{v_x}}\right)\right)\right)}{I}}{\dot{s}} \\ \frac{\tilde{v_x}\sin\tilde{\theta}_e + \tilde{v_y}\cos\tilde{\theta}_e}{\dot{s}} \\ \frac{\tilde{\omega}}{\dot{s}} - \kappa \\ \frac{1}{\dot{s}} \end{bmatrix} ,$$

where

$$\dot{s} = \frac{\tilde{v_x} \cos \tilde{\theta}_e - \tilde{v_y} \sin \tilde{\theta}_e}{1 - \tilde{y_e} \kappa} \ . \tag{3.29}$$

3.5 Conclusions

This chapter discussed a variety of models that describe the vehicle motion. These models can be used to design control, planning and estimation strategies. Furthermore, these models can be extended to consider roll, pitch and vertical motions.

The models described in this chapter were

- Reference-based kinematic model
- Curvature-based kinematic model
- Lateral tire model
- Cartesian dynamic model
- Polar dynamic model
- Polar dynamic and reference-based model (PDRB)
- Cartesian dynamic and curvature-based model (CDCB)
- CDCB in space domain representation

On the one hand, the kinematic representation can be useful for applications where the range of speed does not exceed 5 $\frac{m}{s}$. Besides, its low parameter dependency makes this simple vehicle model very interesting for a wide range of mobile robotics applications. On the other hand, dynamic and combined models encourage thinking about more complex autonomous driving applications, to the point of considering driving situations near the dynamic limit of the vehicle, i.e. autonomous racing.

Note that, Cartesian and Polar dynamic models simply differ in their representation, one being the trigonometric transformation of the other. However, the Polar formulation contains more non-linear expressions that, from a control perspective, can create difficulties in the design process.

In addition, a space domain representation was presented for a particular planning strategy discussed in next sections. The transformation from vehicle body coordinates to inertial coordinates frame was also presented.

Chapter 4

LPV representation

4.1 Introduction

In this chapter, the non-linear embedding approach is used for obtaining an LPV representation of each one of the models presented in Chapter 3. In addition, this chapter will focus on state-space (SS) representation to build a set of models suitable for planning, control and estimation techniques.

Note that, the set of LPV models developed hereafter are CT (continuous-time) and that its DT (discrete-time) formulation can be obtained using Euler techniques or more sophisticated ones [Toth, Roland, Peter SC Heuberger, and Paul MJ Van den Hof., 2010]. In the following sections, the LPV formulation is presented without the output equation since, unless the use case indicates the contrary, we consider that we can either measure or estimate the complete set of vehicle states.

4.2 Reference-based kinematic LPV model

Denoting the state and control vectors, respectively, as

$$x = \begin{bmatrix} x_e \\ y_e \\ \theta_e \end{bmatrix}, \; u = \begin{bmatrix} v \\ \omega \end{bmatrix}, \tag{4.1}$$

we can obtain the LPV representation for the reference-based kinematic model (3.5). Then, considering ω, v_d, $\theta_e \in \mathbb{R}$ as the scheduling variables, the LPV system becomes

$$\dot{x} = A(\omega, v_d, \theta_e)x + Bu - Br \ , \tag{4.2a}$$

where

$$A\left(\omega, v_d, \theta_e\right) = \begin{bmatrix} 0 & \omega & 0 \\ -\omega & 0 & v_d \frac{\sin\theta_e}{\theta_e} \\ 0 & 0 & 0 \end{bmatrix} \ , \tag{4.2b}$$

$$B = \begin{bmatrix} -1 & 0 \\ 0 & 0 \\ 0 & -1 \end{bmatrix} \ , \ r = \begin{bmatrix} v_d \cos\theta_e \\ \omega_d \end{bmatrix} \ . \tag{4.2c}$$

4.3 Curvature-based kinematic LPV model

Denoting the state and control vectors, respectively, as

$$x = \begin{bmatrix} \theta_e \\ s \\ y_e \end{bmatrix} \ , \ u = \begin{bmatrix} v_x \\ v_y \\ \omega \end{bmatrix} \ , \tag{4.3}$$

the LPV form for the curvature-based kinematic model (3.10) can be obtained. Then, considering v_x, v_y, θ_e, y_e and $\kappa \in \mathbb{R}$ as the scheduling variables, the LPV system becomes

$$\dot{x} = B(v_x, v_y, \theta_e, y_e, \kappa)u \ , \tag{4.4a}$$

where

$$B(v_x, v_y, \theta_e, y_e, \kappa) = \begin{bmatrix} B_{11} & B_{12} & 1 \\ B_{21} & 0 & 0 \\ B_{31} & B_{32} & 0 \end{bmatrix} \tag{4.4b}$$

with

$$\begin{gathered} B_{11} = -\frac{\kappa\cos\theta_e}{1 - y_e\kappa} \ , \ B_{12} = \frac{\kappa\sin\theta_e}{1 - y_e\kappa} \\ B_{21} = \frac{v_x\cos\theta_e - v_y\sin\theta_e}{(1 - y_e\kappa)v_x} \ , \ B_{31} = \sin\theta_e \ , \ B_{32} = \cos\theta_e \ \cdot \end{gathered} \tag{4.4c}$$

4.4 Simplified Magic Formula LPV model

The lateral force tire model formulated using the simplified Magic Formula is now converted into an LPV representation.

First, the arctangent function (3.12) is neglected to simplify the formulation. Note that, this assumption is coherent since the slip angle value remains under 0.1745 rad where $\tan^{-1}(\alpha) \approx \alpha$.
Second, the Magic Formula model equations for front and rear wheels in (3.14) are reformulated in a LPV representation for a proper introduction in the final LPV vehicle model. Hence, a least-squares algorithm is used to find two polynomials as

$$F_y(\alpha) = p_1\alpha^n + p_2\alpha^{n-1} + ... + p_n\alpha + p_{n+1} \ , \tag{4.5}$$

where p constants are the estimated coefficients that define the particular model structure and n represents the order of the corresponding polynomial.

Once the polynomial is adjusted, the non-linear embedding approach is used to obtain its LPV representation. Then, the following formulation is proposed

$$F_y = C(\alpha)\ \alpha \ , \tag{4.6}$$

where

$$C(\alpha) = p_1\alpha^{n-1} + p_2\alpha^{n-2} + ... + p_n + p_{n+1}/(\alpha + \epsilon) \tag{4.7}$$

is the tire stiffness coefficient as a function of the slip angle and ϵ is a very small constant. In Figure 3.6, both, front and rear tire stiffness coefficient curves are presented as dashed lines. Note that, as α becomes close to zero in (4.7), $C(\alpha)$ grows exponentially. To avoid such behavior, it is advised to add a saturation in the $\alpha \in [0, 0.0075]$ range such that $C(\alpha)$ remains limited.

4.5 Cartesian dynamic LPV model

The model presented in (3.16) can be considered as highly non-linear. That is why certain simplifications are applied prior to its LPV reformulation.

Note that, since the slip angles always remain at very small angles under non-aggressive driving, we can assume the elimination of the arctangent function in (3.12). This will help in the process of embedding non-linearities into linear parameters.

Then, substituting (3.12) into (4.6), the result into (3.16) and grouping by state and input variables we obtain

$$\begin{aligned}
\dot{v}_x &= \frac{-F_{df}}{mv_x}v_x + \frac{C_f \sin\delta}{mv_x}v_y + \left(\frac{C_f l_f \sin\delta}{mv_x} + v_y\right)\omega - \frac{C_f \sin\delta}{m}\delta + a \\
\dot{v}_y &= -\frac{(C_r + C_f \cos\delta)}{mv_x}v_y - \left(\frac{C_f l_f \cos\delta - C_r l_r}{mv_x} - v_x\right)\omega + \frac{C_f \cos\delta}{m}\delta \ . \\
\dot{\omega} &= -\frac{C_f l_f \cos\delta - C_r l_r}{Iv_x}v_y - \frac{C_r l_r^2 + C_f l_f^2 \cos\delta}{Iv_x}\omega + \frac{C_f l_f \cos\delta}{I}\delta
\end{aligned} \tag{4.8}$$

Note that, the term $\frac{-F_{df}}{m}$ is independent of any state and input variable. Then, the solution is to multiply and divide by the same state or input variable to obtain the desired relationship.

From here, we can represent (4.8) as an LPV model by embedding the non-linearities inside linear parameters that vary in a defined interval (polytope). Denoting the scheduling variables as $\delta \in \mathbb{R}$, $v_x \in \mathbb{R}$ and $v_y \in \mathbb{R}$, the vehicle dynamic LPV model is presented as

$$\dot{x} = A(\delta, v_x, v_y)x + B(\delta, v_x, v_y)u \ , \tag{4.9a}$$

where state and input vectors are

$$x = \begin{bmatrix} v_x \\ v_y \\ \omega \end{bmatrix} \ , \ u = \begin{bmatrix} \delta \\ a \end{bmatrix} \tag{4.9b}$$

and matrices A and B are

$$A(\delta, v_x, v_y) = \begin{bmatrix} A_{11} & A_{12} & A_{13} \\ 0 & A_{22} & A_{23} \\ 0 & A_{32} & A_{33} \end{bmatrix} \tag{4.9c}$$

$$B(\delta, v_x, v_y) = \begin{bmatrix} B_{11} & B_{12} \\ B_{21} & 0 \\ B_{31} & 0 \end{bmatrix} \tag{4.9d}$$

$$A_{11} = \frac{-F_{df}}{mv_x} \ , \ A_{12} = \frac{C_f \sin\delta}{mv_x} \ , \ A_{13} = \frac{C_f l_f \sin\delta}{mv_x} + v_y \tag{4.9e}$$

$$A_{22} = -\frac{C_r + C_f \cos\delta}{mv_x} \ , \ A_{23} = -\frac{C_f l_f \cos\delta - C_r l_r}{mv_x} - v_x \tag{4.9f}$$

$$A_{32} = -\frac{C_f l_f \cos\delta - l_r C_r}{I v_x} \ , \ A_{33} = -\frac{C_f l_f^2 \cos\delta + l_r^2 C_r}{I v_x} \tag{4.9g}$$

$$B_{11} = -\frac{C_f \sin\delta}{m} \ , \ B_{12} = 1 \ , \ B_{21} = \frac{C_f \cos\delta}{m} \ , \ B_{31} = \frac{C_f l_f \cos\delta}{I} \ . \tag{4.9h}$$

4.6 Polar dynamic LPV model

Before jumping directly to the final LPV formulation, a previous representation is presented by developing (3.18) and grouping terms as function of the states and input variables

$$\begin{aligned}
\dot{v} &= (\sin\alpha - \frac{F_{df}}{mv})v - \frac{C_f \sin(\alpha-\delta) + C_r \sin\alpha}{m}\alpha + \frac{C_r l_r \sin\alpha - C_f l_f \sin(\alpha-\delta)}{mv\cos\alpha}\omega \\
&+ \frac{C_f \sin(\alpha-\delta)}{m}\delta + \cos\alpha a \\
\dot{\alpha} &= -\frac{C_f \cos(\alpha-\delta) + C_r \cos\alpha}{mv}\alpha + \left(\frac{1}{mv}\left(\frac{C_r l_r \cos\alpha - C_f l_f \cos(\alpha-\delta)}{v\cos\alpha}\right)\right. \\
&\left. - (1 + v\cos\alpha)\right)\omega + \frac{C_f \cos(\alpha-\delta)}{mv}\delta - \frac{\sin\alpha}{v}a \\
\dot{\omega} &= \frac{C_r l_r - C_f l_f \cos\delta}{I}\alpha - \frac{C_f l_f^2 \cos\delta + C_r l_r^2}{I v\cos\alpha}\omega + \frac{C_f l_f \cos\delta}{I}\delta
\end{aligned} \ . \tag{4.10}$$

From here, we can express the previous non-linear model as a pseudo-linear one by using the non-linear embedding approach. Denoting the scheduling variables as $\delta, \in \mathbb{R}$, $\alpha, \in \mathbb{R}$ and $v, \in \mathbb{R}$, the vehicle dynamic LPV model is presented as

$$\dot{x} = A(\delta, \alpha, v)x + B(\delta, \alpha, v)u \tag{4.11a}$$

with state and input vectors

$$x = \begin{bmatrix} v \\ \alpha \\ \omega \end{bmatrix} \ , \ u = \begin{bmatrix} \delta \\ a \end{bmatrix} \tag{4.11b}$$

and matrices A and B as

$$A(\delta, \alpha, v) = \begin{bmatrix} A_{11} & A_{12} & A_{13} \\ 0 & A_{22} & A_{23} \\ 0 & A_{32} & A_{33} \end{bmatrix} \tag{4.11c}$$

$$B(\delta, \alpha, v) = \begin{bmatrix} B_{11} & B_{12} \\ B_{21} & B_{22} \\ B_{31} & 0 \end{bmatrix} \tag{4.11d}$$

$$A_{11} = (\sin\alpha - \frac{F_{df}}{mv}) \ , \ A_{12} = -\frac{C_f \sin(\alpha - \delta) + C_r \sin\alpha}{m} \tag{4.11e}$$

$$A_{13} = \frac{C_r l_r \sin\alpha - C_f l_f \sin(\alpha - \delta)}{mv\cos\alpha} \ , \ A_{22} = -\frac{C_f \cos(\alpha - \delta) + C_r \cos\alpha}{mv} \tag{4.11f}$$

$$A_{23} = \left(\frac{1}{mv}\Big(\frac{C_r l_r \cos\alpha - C_f l_f \cos(\alpha - \delta)}{v\cos\alpha}\Big) - (1 + v\cos\alpha)\right) \tag{4.11g}$$

$$A_{32} = \frac{C_r l_r - C_f l_f \cos\delta}{I} \ , \ A_{33} = -\frac{C_f l_f^2 \cos\delta + C_r l_r^2}{Iv\cos\alpha} \ , \tag{4.11h}$$

$$B_{11} = \frac{C_f \sin(\alpha - \delta)}{m} \ , \ B_{12} = \cos\alpha \tag{4.11i}$$

$$B_{21} = \frac{C_f \cos(\alpha - \delta)}{mv} \ , \ B_{22} = -\frac{\sin\alpha}{v} \tag{4.11j}$$

$$B_{31} = \frac{C_f l_f \cos\delta}{I} \ . \tag{4.11k}$$

4.7 Polar dynamic with reference-based LPV model (PDRB-LPV)

In this section, models (4.2) and (4.11) are introduced into a same model representation

$$\dot{x} = A(\delta, \alpha, v, \omega, v_d, \theta_e)x + B(\delta, \alpha, v)u + B_r r \tag{4.12a}$$

with state and input vectors

$$x = \begin{bmatrix} v \\ \alpha \\ \omega \\ x_e \\ y_e \\ \theta_e \end{bmatrix}, \; u = \begin{bmatrix} \delta \\ a \end{bmatrix}, \; r = \begin{bmatrix} v_d \cos\theta_e \\ \omega_d \end{bmatrix} \tag{4.12b}$$

and matrices A and B as

$$A(\delta, \alpha, v, \omega, v_d, \theta_e) = \begin{bmatrix} A_{11} & A_{12} & A_{13} & 0 & 0 & 0 \\ 0 & A_{22} & A_{23} & 0 & 0 & 0 \\ 0 & A_{32} & A_{33} & 0 & 0 & 0 \\ -1 & 0 & 0 & 0 & \omega & 0 \\ 0 & 0 & 0 & -\omega & 0 & A_{56} \\ 0 & 0 & -1 & 0 & 0 & 0 \end{bmatrix} \tag{4.12c}$$

$$B(\delta, \alpha, v) = \begin{bmatrix} B_{11} & B_{12} \\ B_{21} & B_{22} \\ B_{31} & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix}, \; B_r = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 1 & 0 \\ 0 & 0 \\ 0 & 1 \end{bmatrix} \tag{4.12d}$$

$$A_{56} = v_d \frac{\sin(\theta_e)}{\theta_e} \; . \tag{4.12e}$$

4.8 Cartesian dynamic with curvature-based LPV model (CDCB-LPV)

In this section, we present the LPV representation after joining the curvature-based (4.4) and Cartesian models (4.9). The state space representation is given by

$$\dot{x} = A(v_x, v_y, \theta_e, \kappa, y_e, \delta)x + B(\delta)u \tag{4.13a}$$

with state and input vectors

$$x = \begin{bmatrix} v_x \\ v_y \\ \omega \\ \theta_e \\ s \\ y_e \end{bmatrix}, \ u = \begin{bmatrix} \delta \\ a \end{bmatrix} \tag{4.13b}$$

and matrices A and B as

$$A(v_x, v_y, \theta_e, \kappa, y_e, \delta) = \begin{bmatrix} A_{11} & A_{12} & A_{13} & 0 & 0 & 0 \\ 0 & A_{22} & A_{23} & 0 & 0 & 0 \\ 0 & A_{32} & A_{33} & 0 & 0 & 0 \\ A_{41} & A_{42} & 1 & 0 & 0 & 0 \\ A_{51} & 0 & 0 & 0 & 0 & 0 \\ A_{61} & A_{62} & 0 & 0 & 0 & 0 \end{bmatrix} \tag{4.13c}$$

$$B(\delta) = \begin{bmatrix} -\frac{1}{m}\sin\delta C_f & 1 \\ \frac{1}{m}\cos\delta C_f & 0 \\ \frac{1}{I}\cos\delta C_f l_f & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix}, \tag{4.13d}$$

where

$$\begin{gathered} A_{11} = \frac{-F_{df}}{v_x}, \ A_{12} = \frac{C_f \sin\delta}{m v_x}, \ A_{13} = \frac{C_f l_f \sin\delta}{m v_x} + v_y \\ A_{22} = -\frac{C_r + C_f \cos\delta}{m v_x}, \ A_{23} = -\frac{C_f l_f \cos\delta - C_r l_r}{m v_x} - v_x \\ A_{32} = -\frac{C_f l_f \cos\delta - l_r C_r}{I v_x}, \ A_{33} = -\frac{C_f l_f^2 \cos\delta + l_r^2 C_r}{I v_x} \\ A_{41} = -\frac{\kappa\cos\theta_e}{1 - y_e\kappa}, \ A_{42} = \frac{\kappa\sin\theta_e}{1 - y_e\kappa} \\ A_{51} = \frac{v_x\cos\theta_e - v_y\sin\theta_e}{(1 - y_e\kappa)v_x}, \ A_{61} = \sin\theta_e, \ A_{62} = \cos\theta_e \end{gathered} \tag{4.13e}$$

Remark 4.1. Note that in last chapter, model CDCB (3.25) was developed using the simplified Magic Formula model for the lateral tire model. However, the CDCB-LPV representation is developed using the linear lateral tire model formulation (3.13).

4.9 Trajectory planning LPV model

The model used for trajectory planning is an extension of the CDCB-LPV representation. The dynamics of the vehicle, represented by states v_x, v_y and ω, have exactly the same LPV representation. However, the rest of the model suffers of some reformulation. In particular, the front and rear wheels slip angles are introduced as states and the s state is removed.

Then, the continuous time state space representation is given by

$$\dot{x} = A(v_x, v_y, \theta_e, \kappa, y_e, \delta)x + B(\delta)u \tag{4.14a}$$

with state and input vectors

$$x = \begin{bmatrix} v_x \\ v_y \\ \omega \\ \theta_e \\ y_e \\ \alpha_f \\ \alpha_r \end{bmatrix}, \ u = \begin{bmatrix} \delta \\ a \end{bmatrix} \tag{4.14b}$$

and matrices A and B as

$$A(v_x, v_y, \theta_e, \kappa, y_e, \delta) = \begin{bmatrix} A_{11} & A_{12} & A_{13} & 0 & 0 & 0 & 0 \\ 0 & A_{22} & A_{23} & 0 & 0 & 0 & 0 \\ 0 & A_{32} & A_{33} & 0 & 0 & 0 & 0 \\ -A_{41} & A_{42} & A_{43} & 0 & 0 & 0 & 0 \\ 0 & A_{52} & 0 & A_{54} & 0 & 0 & 0 \\ 0 & A_{62} & A_{63} & 0 & 0 & 0 & 0 \\ 0 & A_{72} & A_{73} & 0 & 0 & 0 & 0 \end{bmatrix} \tag{4.14c}$$

and

$$B(\delta) = \begin{bmatrix} -\frac{1}{m}\sin\delta C_f & 1 \\ \frac{1}{m}\cos\delta C_f & 0 \\ \frac{1}{I}\cos\delta C_f l_f & 0 \\ 0 & 0 \\ 0 & 0 \\ 1 & 0 \\ 0 & 0 \end{bmatrix} , \tag{4.14d}$$

being

$$\begin{gathered} A_{11} = \left(\tfrac{-\mu g}{v_x} - \tfrac{\rho C_{dA} v_x}{2m}\right) \ , \ A_{12} = \tfrac{C_f \sin\delta}{m v_x} \ , \ A_{13} = \left(\tfrac{C_f l_f \sin\delta}{m v_x} + v_y\right) \\ A_{22} = \left(-\tfrac{C_r + C_f \cos\delta}{m v_x}\right) \ , \ A_{23} = \left(-\tfrac{C_f l_f \cos\delta - C_r l_r}{m v_x} - v_x\right) \ , \ A_{32} = -\tfrac{C_f l_f \cos\delta - l_r C_r}{I v_x} \\ A_{33} = \left(-\tfrac{C_f l_f^2 \cos\delta + l_r^2 C_r}{I v_x}\right) \ , \ A_{41} = \tfrac{\kappa}{(1 - y_e \kappa)} \ , \ A_{42} = \tfrac{\kappa}{(1 - y_e \kappa)} \sin\theta_e \ , \ A_{43} = 1 \\ A_{52} = \cos\theta_e \ , \ A_{54} = v_x \ , \ A_{62} = \tfrac{-1}{v_x} \\ A_{63} = \tfrac{-l_f}{v_x} \ , \ A_{72} = \tfrac{-1}{v_x} \ , \ A_{73} = \tfrac{l_r}{v_x} \end{gathered} \tag{4.14e}$$

Note that this model uses the Magic Formula LPV model presented in Section 4.4. However, for a easier comprehension $C_i(\alpha_i)$ is denoted by C_i being $i = f, r$. Note also that when discretizing this continuous time system the last two states, i.e. α_f and α_r are not time dependent and then they do not need to be discretized.

Remark 4.2. There is not only one non-linear embedding option. Depending on how the designer encapsulates the non-linearities into varying parameters the result will differ and consequently, the quality of the representation.

4.10 Conclusions

In this chapter, the procedure for the generation of LPV systems as well as the formulation of the models of the previous chapter have been addressed.

In particular, it has been shown that starting from the non-linear representation of a system and using analytical methodologies, the LPV formulation in state space can be achieved. Among the analysis and synthesis methodologies, the polytopic approach has been presented giving rise to the LPV polytopic system.

LPV representations for the different non-linear models presented in the previous chapter have been developed. Note that the embedding of non-linearities into the varying parameters is not unique and therefore there is an optimal structure that in many cases is not trivial to find.

In the non-linear embedding process, we find the need to construct varying parameters that are singular in zero longitudinal velocity. However, in Chapter 3, the dynamic model as a whole, and the tire model in particular, was presented as a system with the same singularity. Consequently, in the next chapters oriented to control, planning and estimation, this problem will be treated preventing a zero linear velocity ($v_x = 0$).

Finally, comparing both combined LPV models, PDRB and CDCB, a greater mathematical complexity has been observed in the development and management of the first one. In the PDRB-LPV model, the reference matrix stands out, making it a more extensive and less common structure model. However, the biggest difference between the two, and positive for the CDCB-LPV model, is the simplicity and low dependence of scheduling variables in the input matrix (B).

Part II

Control and estimation

Chapter 5

Kinematic Lyapunov-based control with LQR-LMI tuning

The content of this chapter is based on the following work:

- [Alcalá, Eugenio, et al., 2018.A]. Alcalá, E., Puig, V., Quevedo, J., Escobet, T., & Comasolivas, R. (2018). Autonomous vehicle control using a kinematic Lyapunov-based technique with LQR-LMI tuning. *Control engineering practice*, 73, 1-12.

5.1 Introduction

In the last decades, Lyapunov theory has become a standard rule for analyzing stability of non-linear systems [Freeman, Randy, and Petar V. Kokotovic., 2008, Dixon, Warren E., et al., 2001], but also for obtaining model-based strategies for controlling the studied systems [Alcalá, Eugenio, et al., 2016, Blažič, Sašo., 2010, Dixon, Warren E., et al., 2001]. In particular, when working with LPV systems, a LMI-based expression can be used for checking Lyapunov stability. Such a LMI formalism has become a standard for analysis and control design in recent years [Duan, Guang-Ren, and Hai-Hua Yu., 2013].

In this chapter, a non-linear kinematic Lyapunov-based control is proposed for solving both, the lateral and longitudinal control problem. An optimization algorithm for adjusting non-linear controller parameters is also proposed. This algorithm is based on formulating the closed-loop system in LPV form. Then, the Lyapunov controller parameters are obtained based on Linear-Quadratic Regulation-LMI (LQR-LMI) approach. The idea behind the proposed tuning approach is rooted in the work of Farag, A., and H. Werner. [2004], where an approach for fixed structure controller is proposed splitting the problem into a convex and a non-convex sub-problems. A method for solving the convex sub-problem via LMIs is presented in El Ghaoui, L., and V. Balakrishnan. [1994].

The trajectory generation, which uses a map and a global planner to compute the best trajectory for reaching the destination, is briefly presented. This trajectory is coarsely defined by a reduced number of global way-points, which are defined by its global positioning system (GPS) coordinates and the vehicle orientation. In order to execute the maneuvers comfortably, a local planner computes a smooth trajectory by adding intermediate local way-points defined by their GPS position, orientation and the desired linear and angular velocities.

Finally, the proposed techniques for vehicle motion control are first tested in a virtual reality environment (SYNTHIA [Ros, German, et al., 2016]). Then, a real on-field test scenario using an electric Tazzari vehicle is used for showing effectiveness in real conditions.

5.2 Vehicle description

The results presented in this paper are part of the project called *Elektra*[1] that aims to develop autonomous guidance technology for road vehicles. For such a purpose, an electric Tazzari zero vehicle [Tazzari Zero., 2006] is used (see Figure 5.1). This system is a non-holonomic platform that can move like a normal road vehicle. This platform is composed by a set of sensors and actuators, as well as a PC and an electronic control unit (ECU) that manage all algorithms and communications between them. The diagram of the control architecture is depicted in Figure 5.2. On one hand, the vehicle has on board an inertial measurement unit (IMU), GPS and stereo cameras to obtain information about the environment and current state. Proper algorithms have to be employed in order to convert that crude information on convenient data for understanding the environment and localize the vehicle. On the other hand, a set of actuators are employed to perform

[1] http://adas.cvc.uab.es/elektra/

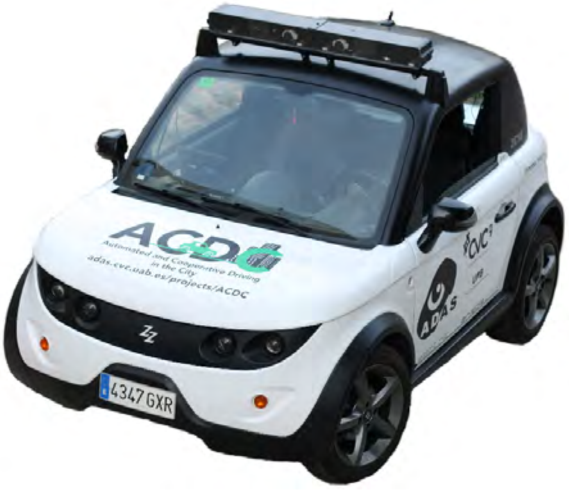


Figure 5.1: Electric Tazzari Zero vehicle used for testing

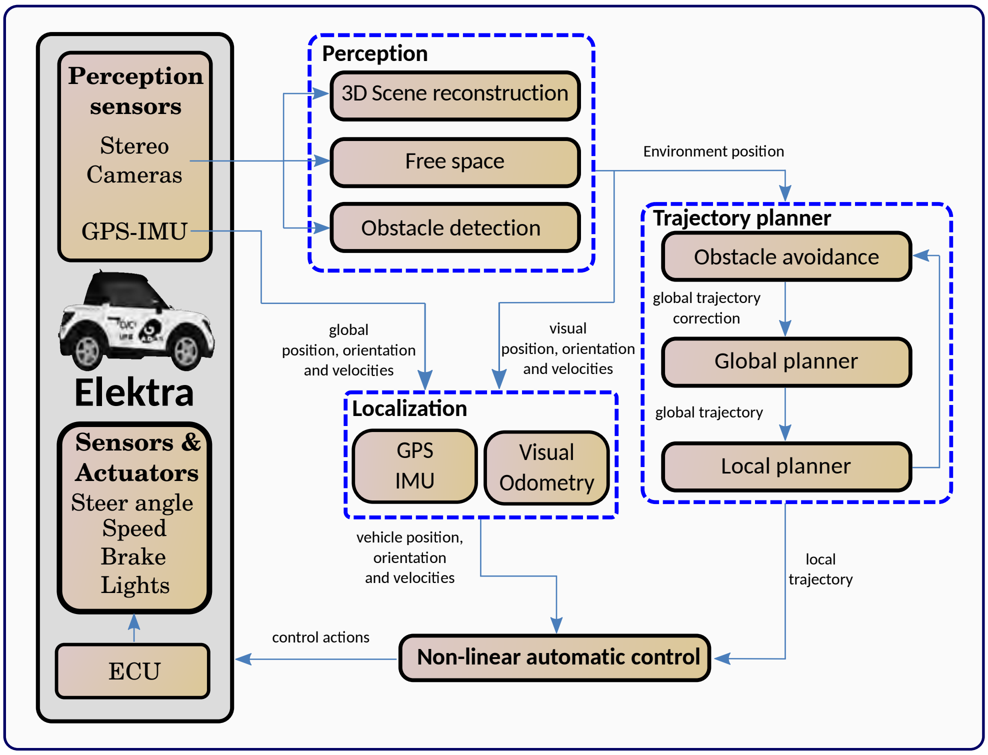


Figure 5.2: Elektra control architecture. The large block on the left represents the physical vehicle devices. The rest of the blocks represent software algorithms

the motion (steering and driven electric motors) as well as turning on the lights and opening doors. The rest of modules in Figure 5.2 (perception, localization, planning and control) compose the software for performing the autonomous guidance task. This paper specially focuses on the non-linear automatic control module. However, the trajectory planning task is introduced for better understanding.

All the algorithms involved run over a trunk PC (6-core i7 5930K, 32GB DDR4) running

ROS on GNU/Linux (Ubuntu distribution). An NVIDIA GTX Titan X board is used to run GPU-based algorithms for perception-image analysis.

The ECU, based on a Cortex-M4 MCU, runs a custom embedded software which communicates the PC control actions to the different car actuators (steering, throttle, brake, lights, horn), as well as reads the values of the car state sensors (steering, throttle, brake, speed, doors, battery).

The communication net is based on CAN bus protocol. Its cycle is currently set to 100 ms, which is sufficient for running all required algorithms.

5.3 Trajectory generation

This section addresses the module responsible of generating the trajectory planning for achieving the desired goal (observe this module in the overall vehicle architecture presented in Figure 5.2). Information from other modules, such as obstacle avoidance and localization, is received in order to compute free-collision trajectories. Figure 5.3 shows the trajectory generation module and its sub-modules as well as the input and output data.

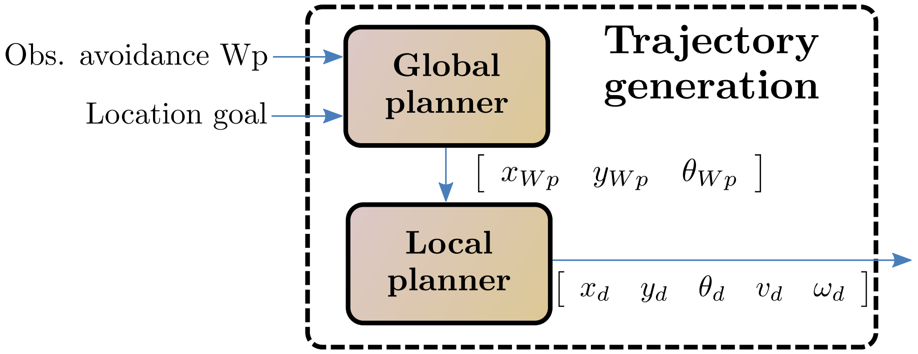

Obs. avoidance Wp
Location goal
Global planner
Trajectory generation
$\left[\ x_{Wp} \quad y_{Wp} \quad \theta_{Wp}\ \right]$
Local planner
$\left[\ x_d \quad y_d \quad \theta_d \quad v_d \quad \omega_d\ \right]$

Figure 5.3: Trajectory generation diagram. Sub-index Wp refers to Way-point and d to desired variables

In case the perception detects an obstacle the obstacle avoidance strategy provides a new way-point (Obstacle avoidance way-point) to avoid such an obstacle. Location goal represents the coordinates of the desired final point. x, y and θ are the current states of the vehicle.

At this point, the vehicle is in charge of managing two planning stages: Global and Local planning. They can be seen as two overlapping and connected layers being the Global planner the upper one. Note that, both planners represent their coordinates and orientation (x, y and θ) with respect to the inertial global frame.

5.3.1 Global Planner

A human-vehicle interface based on the OpenStreetMap [Haklay, Mordechai, and Patrick Weber., 2008] open software is used to introduce the route as a set of way-points along the street (cyan ball in Figure 5.4). Moreover, information about close obstacles is introduced with the goal of recomputing the global trajectory. The position of each way-point is provided in the Universal Transverse Mercator (UTM) coordinate system. Once the global plan is defined, it provides way-point information (x_{Wp}, y_{Wp}, θ_{Wp}) to the local trajectory planner.

5.3.2 Local Planner

Passenger comfort determines driving quality. The most remarkable variables affecting passenger comfort are the lateral and longitudinal accelerations. High accelerations will annoy passengers, who will find it very difficult to maintain posture. The ISO 2631-1 [International Organization for Standarization., 1997] standard recommends an overall vehicle acceleration (a_w) less than 0.315 $\frac{m}{s^2}$, which is defined as

$$a_w = \sqrt{a_{wx}^2 + a_{wy}^2 + a_{wz}^2}\ . \tag{5.1}$$

Following Solea, Razvan, and Urbano Nunes. [2007] and Bianco, CG Lo, Aurelio Piazzi, and Massimo Romano. [2004], a quintic spline-based trajectory planner is implemented that generates smooth trajectories with a velocity profile with continuous acceleration and low levels of jerk, ensuring the passenger comfort. Our work adopts a simplified version of such an algorithm: instead of using smooth but variable velocities in straight sections, which is harder for the tracking control task, constant velocity sections are proposed. The algorithm defines three operation modes

- Acceleration stage: computes a smooth velocity profile under bounded acceleration.

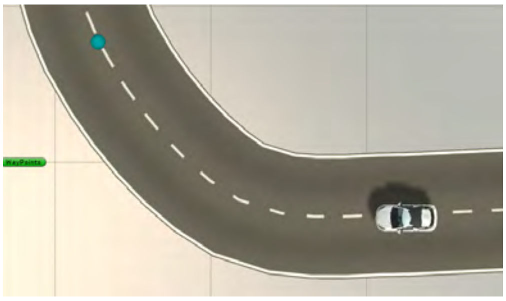


FIGURE 5.4: Global trajectory planning. Cyan point represents a global way-point

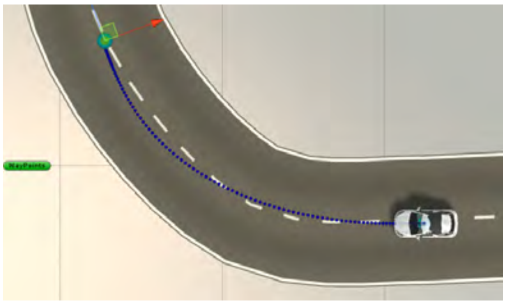


FIGURE 5.5: Local trajectory planning. Small blue points represent the planned route

- Constant velocity stage: maintains a constant velocity reference using the control module.
- Deceleration stage: computes a smooth velocity profile under bounded negative acceleration.

This module will provide a set of local way-points to the control module (small blue points in Figure 5.5). Each local way-point is defined as a set of desired values (x_d, y_d, θ_d, v_d and ω_d) as it is shown in Figure 5.3.

5.4 Automatic vehicle control

The automatic control strategy tackles the problem of generating an appropriate behaviour to follow the desired trajectory. Thus, it is in charge of computing smooth control actions (vehicle speed and steering angle) such that the vehicle is capable of achieving the required speed and orientation at the next local way-point (observe this module in the overall vehicle architecture presented in Figure 5.2).

In this section, a non-linear automatic control strategy based on the Lyapunov theory [Alcalá, Eugenio, et al., 2016, Aicardi, Michele, et al., 1995, Dixon, Warren E., et al., 2001, Samson, Claude, and Karim Ait-Abderrahim., 1990] is introduced for trajectory tracking and navigation among way-points as well as a tuning methodology based on LPV LMI-based linear-quadratic regulation (LQR) approach.

The idea of the Lyapunov method is to define a control law that ensures the stability and the asymptotic elimination of the error between the real and desired vehicle trajectory.

Figure 5.6 shows the Lyapunov control sub-modules as well as input and output variables. The controller receives the set-points of the trajectory provided by the Local planner (x_d, y_d, θ_d, v_d and ω_d) and the current localization of the vehicle (x, y, θ). With all this information, a set of errors (x_e, y_e, θ_e) are computed by using (3.3) which will be used by the Lyapunov-based controller.

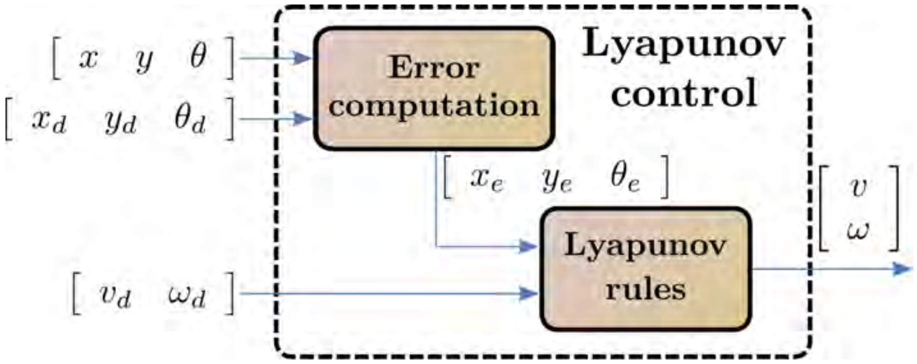

$[x \;\; y \;\; \theta]$
$[x_d \;\; y_d \;\; \theta_d]$
Error computation
Lyapunov control
$[x_e \;\; y_e \;\; \theta_e]$
Lyapunov rules
$[v_d \;\; \omega_d]$
$\begin{bmatrix} v \\ \omega \end{bmatrix}$

FIGURE 5.6: Lyapunov based controller diagram

5.4.1 Lyapunov control design

Using the kinematic model presented in Section 3.2.1, the Lyapunov-based controller is designed according to the following Theorem.

Theorem 5.1. *Given the reference-based kinematic model of the vehicle* (3.5)*, the control law*

$$\begin{cases} v = k_1 x_e + v_d \cos\theta_e \\ \omega = \omega_d + k_2 v_d \frac{\sin\theta_e}{\theta_e} y_e + k_3 \theta_e \end{cases} \tag{5.2}$$

stabilises the closed-loop dynamics in the Lyapunov sense if the controller parameters k_1, k_2 and k_3 are positive.

Proof. Following the Lyapunov's stability, the following Lyapunov function candidate is proposed

$$V(e) = \frac{k_2}{2} x_e^2 + \frac{k_2}{2} y_e^2 + \frac{1}{2} \theta_e^2 \; . \tag{5.3}$$

Its time derivative is

$$\dot{V}(e) = k_2 x_e \dot{x}_e + k_2 y_e \dot{y}_e + \theta_e \dot{\theta}_e \; . \tag{5.4}$$

Now, by substituting the open-loop equations (3.5) in (5.4)

$$\dot{V}(e) = k_2 x_e v_d \cos\theta_e - k_2 x_e v + k_2 y_e v_d \sin\theta_e + \theta_e \omega_d - \theta_e \omega \ . \tag{5.5}$$

By inspection, an expression of the controller in terms of control actions v and ω is determined

$$\begin{bmatrix} v \\ \omega \end{bmatrix} = \begin{bmatrix} k_1 x_e + v_d \cos\theta_e \\ \omega_d + k_2 v_d \frac{\sin\theta_e}{\theta_e} y_e + k_3 \theta_e \end{bmatrix} , \tag{5.6}$$

such that

$$\dot{V}(e) = -k_2 k_1 x_e^2 - k_3 \theta_e^2 < 0 \tag{5.7}$$

fulfilling the Lyapunov's Theorem under the condition that the control parameters satisfy

$$k_1, k_2, k_3 > 0 \ . \tag{5.8}$$

Once the control equations have been obtained the closed-loop error system has the following shape by inserting (5.6) in (3.5)

$$\begin{bmatrix} \dot{x}_e \\ \dot{y}_e \\ \dot{\theta}_e \end{bmatrix} = \begin{bmatrix} \omega y_e - k_1 x_e \\ -\omega x_e + v_d \sin(\theta_e) \\ -k_2 v_d \frac{\sin\theta_e}{\theta_e} y_e - k_3 \theta_e \end{bmatrix} . \tag{5.9}$$

Observe that from (5.7), the convergence of x_e and θ_e is guaranteed, i.e.

$$\lim_{t\to\infty} x_e(t) = 0 \qquad \lim_{t\to\infty} \theta_e(t) = 0 \ . \tag{5.10}$$

However, the convergence of y_e is not ensured. In order to demonstrate that $\lim_{t\to\infty} y_e(t) = 0$, the proof relies on the result presented in Theorem 1.2 of Dixon, Warren E., et al. [2001] that shows

$$\lim_{t\to\infty} \dot{\theta}_e(t) = 0 \tag{5.11}$$

when using the control law (5.6). Hence, (5.11) leads to

$$\lim_{t\to\infty} k_2 \cdot v_d(t) \cdot \frac{\sin\theta_e(t)}{\theta_e(t)} y_e(t) = 0 \tag{5.12}$$

considering (5.10) and (5.9). And, consequently

$$\lim_{t\to\infty} y_e(t) = 0 \tag{5.13}$$

assuming that, when following the desired trajectory, the velocity control action $v_d(t)$ is not null and $\theta_e \cong 0$ such that

$$\lim_{\theta_e\to 0} \frac{\sin\theta_e}{\theta_e} = 1\ . \tag{5.14}$$

Thus, the achievement of the global asymptotic stability can be concluded. □

5.4.2 Lyapunov control adjustment via LQR-LMI

The condition (5.8) guarantees that the controller is stable, but it does not allow to establish performance specifications. In this section, an iterative algorithm for adjusting the non-linear Lyapunov controller using a LQR-LMI based strategy is proposed.

The method starts by rewriting the closed-loop error system (5.9) in the LPV form. The small-angle approximation is applied to simplify the LPV formulation since the error of orientation (θ_e) remains close to zero

$$\frac{\sin\theta_e}{\theta_e} \approx 1\ . \tag{5.15}$$

Considering ω, $v_d \in \mathbb{R}$ as the scheduling variables and $K_s = [k_1\ k_2\ k_3]$

$$\begin{bmatrix} \dot{x}_e \\ \dot{y}_e \\ \dot{\theta}_e \end{bmatrix} = \overbrace{\begin{bmatrix} -k_1 & \omega & 0 \\ -\omega & 0 & v_d \\ 0 & -k_2 v_d & -k_3 \end{bmatrix}}^{A_{cl}(K_s,\omega,v_d)} \begin{bmatrix} x_e \\ y_e \\ \theta_e \end{bmatrix}\ . \tag{5.16}$$

At this point, taken (5.15) in consideration, the control strategy (5.2) can be seen as a state feedback control law in the form $u = Kx + r$

$$\begin{bmatrix} v \\ \omega \end{bmatrix} = \overbrace{\begin{bmatrix} k_1 & 0 & 0 \\ 0 & k_2 v_d & k_3 \end{bmatrix}}^{K(K_s,\omega,v_d)} \begin{bmatrix} x_e \\ y_e \\ \theta_e \end{bmatrix} + \begin{bmatrix} v_d \cos\theta_e \\ \omega_d \end{bmatrix}\ . \tag{5.17}$$

The scheduling variables are bounded in a box (see Figure 5.7) defined by the operating conditions. The controller is scheduled according to the expression $K(K_s, \omega, v_d)$ in (5.17). The controller parameters (5.8) will be determined using the LQR technique via LMI as

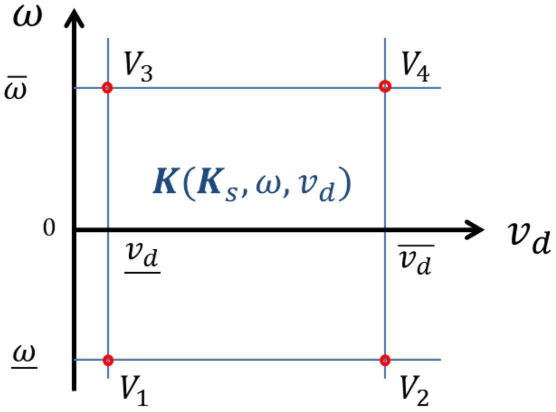


FIGURE 5.7: Bounding box governed by the robust controller $K(K_s, \omega, v_d)$. V_i, with $i = 1, ..., 4$, represent the vertexes of the bounding box

suggested in Duan, Guang-Ren, and Hai-Hua Yu. [2013] using the LMI solution for the H_2 problem given by

$$
\begin{aligned}
& A_{cl}(K_s, \omega, v_d)P + (A_{cl}(K_s, \omega, v_d)P)^T + 2\alpha P > 0 \\
& A_{cl}(K_s, \omega, v_d)P + (A_{cl}(K_s, \omega, v_d)P)^T < 0 \\
& \begin{bmatrix} -Y & R^{\frac{1}{2}}K(K_s, \omega, v_d)P \\ (R^{\frac{1}{2}}K(K_s, \omega, v_d)P)^T & -P \end{bmatrix} < 0 \\
& trace(Q^{\frac{1}{2}}P(Q^{\frac{1}{2}})^T) + trace(Y) < \gamma \\
& P \geq 0,\ Y = Y^T > 0\ , \qquad \forall v_d \in [\underline{v_d}, \overline{v_d}], \omega \in [\underline{\omega}, \overline{\omega}]\ ,
\end{aligned} \tag{5.18}
$$

that is converted into an LMI by means of the following change of variable: $W = K(K_s, \omega, v_d)P$, where $K_s \in \mathbb{R}^{1 \times s}$, $P \in \mathbb{R}^{s \times s}$ is the Lyapunov matrix and the result of the LMI problem. $Q \in \mathbb{R}^{s \times s}$, $R \in \mathbb{R}^{r \times r}$, $Y \in \mathbb{R}^{n \times n}$ and $\gamma \in \mathbb{R}$ are tuning parameters in the LMI-LQR problem being s and n the number of states and inputs, respectively.

However, this procedure would deliver a free structure state feedback controller K, i.e. not keeping the structure of the Lyapunov control law (5.17). To preserve this structure is not an easy task as discussed in El Ghaoui, L., and V. Balakrishnan. [1994] since leads to a non-convex problem.

Here, to preserve the fixed structure of the control law (5.17), an optimization problem that has as decision variables the control parameters (K_s) and as objective function the infinity norm of the Lyapunov matrix (P) eigenvalues, is used to maximize the LQR performance as follows

$$\begin{aligned} &\min_{K_s} \quad \|eig(P)\|_\infty \\ &\text{s.t.} \\ &\qquad P = \text{LQR-LMI-problem}(\{K_{s,i}\}_{i=1}^{4}) \\ &\qquad K_{s,i} \in \left(0, \overline{K}_{s,i}\right] \ , \qquad\qquad\qquad \forall i = 1, ..., 4 \ , \end{aligned} \tag{5.19}$$

where $eig()$ function returns a vector containing the eigenvalues of square matrix P. The function LQR-LMI-problem has as input the set of K_s vectors and solves the following set of LMIs

$$\begin{aligned} &A_{cl}(K_{s,i}, \omega_i, v_{d_i})P + (A_{cl}(K_{s,i}, \omega_i, v_{d_i})P)^T + 2\alpha P > 0 \qquad \forall i = 1, ..., 4 \\ &A_{cl}(K_{s,i}, \omega_i, v_{d_i})P + (A_{cl}(K_{s,i}, \omega_i, v_{d_i})P)^T < 0 \qquad \forall i = 1, ..., 4 \\ &\begin{bmatrix} -Y & R^{\frac{1}{2}}K(K_{s,i}, \omega_i, v_{d_i})P \\ (R^{\frac{1}{2}}K(K_{s,i}, \omega_i, v_{d_i})P)^T & -P \end{bmatrix} < 0 \quad \forall i = 1, ..., 4 \\ &trace(Q^{\frac{1}{2}}P(Q^{\frac{1}{2}})^T) + trace(Y) < \gamma \\ &P \geq 0, \ Y = Y^T > 0 \ . \end{aligned} \tag{5.20}$$

Note that, in (5.19) and (5.20), i represents each one of the polytope vertexes in Figure (5.7), $K_{s,i}$ represents the i optimization vectors K_s. The parameter α represents the boundary for setting the kinematic closed-loop poles (see Figure 5.8). Note that, in (5.19), $\overline{K}_{s,i}$ is the upper boundary for the control parameters. Such a boundary has been chosen as an arbitrary very high value in order to ensure the optimal gains are found.

In order to select the constraint for the pole placement α in (5.20), the fact that the Lyapunov controller provides the set point to the dynamic loop has to be taken into account. This internal loop has been implemented by using two decoupled PI controllers adjusted by means of the pole placement technique. In order to achieve a good kinematic reference tracking, the dynamic control loop has been considered four times faster than the kinematic one. This leads to locate the kinematic closed loop poles in a specific region between 0 and α (see Figure 5.8). Such a restriction is presented in the form of a LMI in the optimization problem (5.19) as: $A_{cl}(K_{s,i}, \omega_i, v_{d_i})P + (A_{cl}(K_{s,i}, \omega_i, v_{d_i})P)^T + 2\alpha P > 0$. At this point, four polytopic controllers, one at each vertex (K_i), are considered (see Figure 5.7). Then, at every control iteration the control gain is computed as

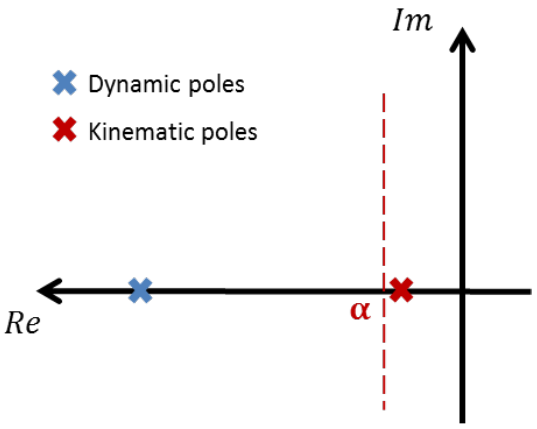


FIGURE 5.8: Representation of the pole placement issue. Dominant kinematic poles must be slower than the dominant dynamic poles. α represents the hyper-plane location

$$K = \sum_{i=1}^{4} \mu_i K_i \ , \tag{5.21}$$

where the vertex membership function μ_i is calculated using (2.2)-(2.5).

It is important to note that due to the high non-convexity of the optimization problem, common gradient-based solvers are not applicable and thus, genetic algorithms have been useful for solving it. This heuristic algorithm does not ensure a global optimal solution but only a local optimal solution.

5.5 Simulation results

In this section, the behaviour of each module previously introduced is evaluated, i.e. global and local planning, and automatic control. For this purpose the modules have been evaluated in SYNTHIA [2] [Ros, German, et al., 2016]. It runs over Unity [3] which is a game development platform.

[2] http://synthia-dataset.net/
[3] https://unity3d.com/es

In this environment, localization is considered ideal and neither noises nor disturbances have been added. However, the point of interest is that vehicle dynamics is modeled in realistic manner considering the complex vehicle physics [Garcia Angel., 2010]. The global planner defines the route composed with a set of way-points along the scenario (cyan points in Figure 5.10). The local trajectory planner has been adjusted and constrains the overall vehicle acceleration as explained in Section 5.3.2. All algorithms are executed in a regular manner within a period of 0.1 s. The complete diagram for simulation can be seen in Figure 5.9.

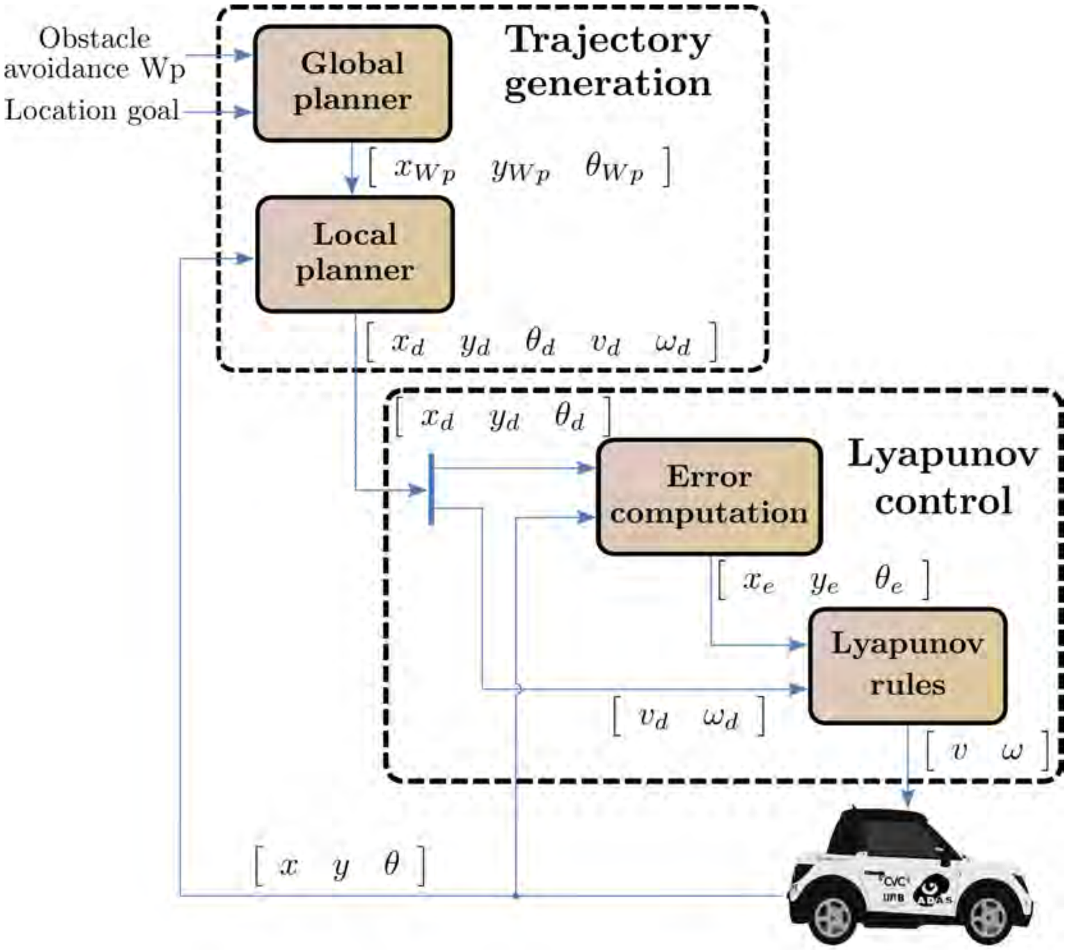

Obstacle avoidance Wp
Location goal
Global planner
Trajectory generation
$[\ x_{Wp} \quad y_{Wp} \quad \theta_{Wp}\ ]$
Local planner
$[\ x_d \quad y_d \quad \theta_d \quad v_d \quad \omega_d\ ]$
$[\ x_d \quad y_d \quad \theta_d\ ]$
Error computation
Lyapunov control
$[\ x_e \quad y_e \quad \theta_e\ ]$
Lyapunov rules
$[\ v_d \quad \omega_d\ ]$
$[\ v \quad \omega\ ]$
$[\ x \quad y \quad \theta\ ]$

FIGURE 5.9: Complete motion diagram composed by the trajectory planning and the Lyapunov control

The section is divided in two simulations tests. On one hand, the Test A is presented. It is composed by a circuit designed for testing the algorithms along the whole range of urban velocities (i.e. 0 - 60 $\frac{km}{h}$). It offers also different curvature curves. On the other hand, Test B is composed by a geometrically simpler circuit in which the vehicle works

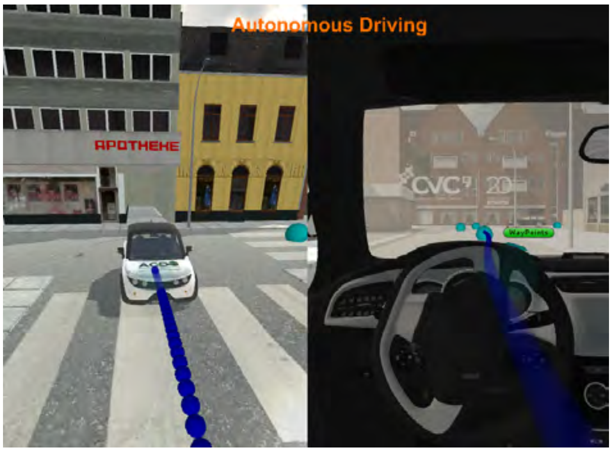


FIGURE 5.10: SYNTHIA scenario: Screen shot of the simulation. External view in the left side and view from inside in the right side

at lower velocities (i.e. 0 - 18 $\frac{km}{h}$). It is designed with the idea of simulating the real experiment.

5.5.1 Test A

For this test, the scheduling variables have been bounded in the following intervals: $\omega \in [-1.417, 1.417]\frac{rad}{s}$ and $v_d \in [0.1, 16.7]\frac{m}{s}$. The v_d interval starts in $0.1\frac{m}{s}$ since the null velocity is a singular point for the controller and it has to be avoided. Both in simulation and in the real test, the vehicle begins the performance by applying a little force over the rear wheels in order to achieve this low velocity bound and being inside v_d interval. The matrices and variables used for adjusting the LQR optimization are

$$Q = \begin{bmatrix} 10 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 1 \end{bmatrix}, R = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \gamma = 0.0001 \; .$$

The control parameters are presented in Table 5.1 for each bounding box vertex. Note that the controller gains are only function of the linear velocity. This is because of the vehicle geometry is symmetric with respect to its longitudinal axis. The results of applying the Lyapunov control technique adjusted by means of the LQR-LMI approach are shown in Figure 5.11 and 5.12. Figure 5.11 shows the proposed circuit and the trajectory result. Such a circuit has been designed with the idea of offering the vehicle different levels of difficulty. Then, a set of curves with different curvature has been introduced. From Figure 5.12.a, it can be seen that the velocity profile is divided in

	v_d	ω	k_1	k_2	k_3
i=1	0.1	-1.42	3.9	1.1	1.5
i=2	16.7	-1.42	3.6	1.2	2.1
i=3	0.1	1.42	3.9	1.1	1.5
i=4	16.7	1.42	3.6	1.2	2.1

TABLE 5.1: Test A - Control parameters for each vertex of the bounding box (see Figure 5.7)

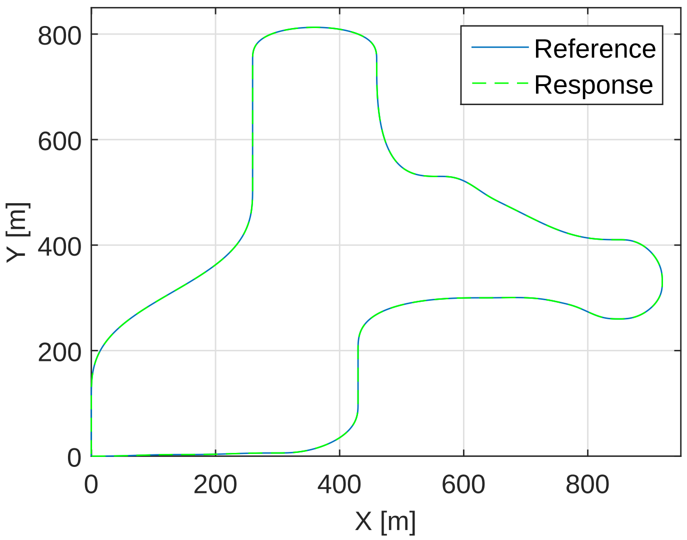
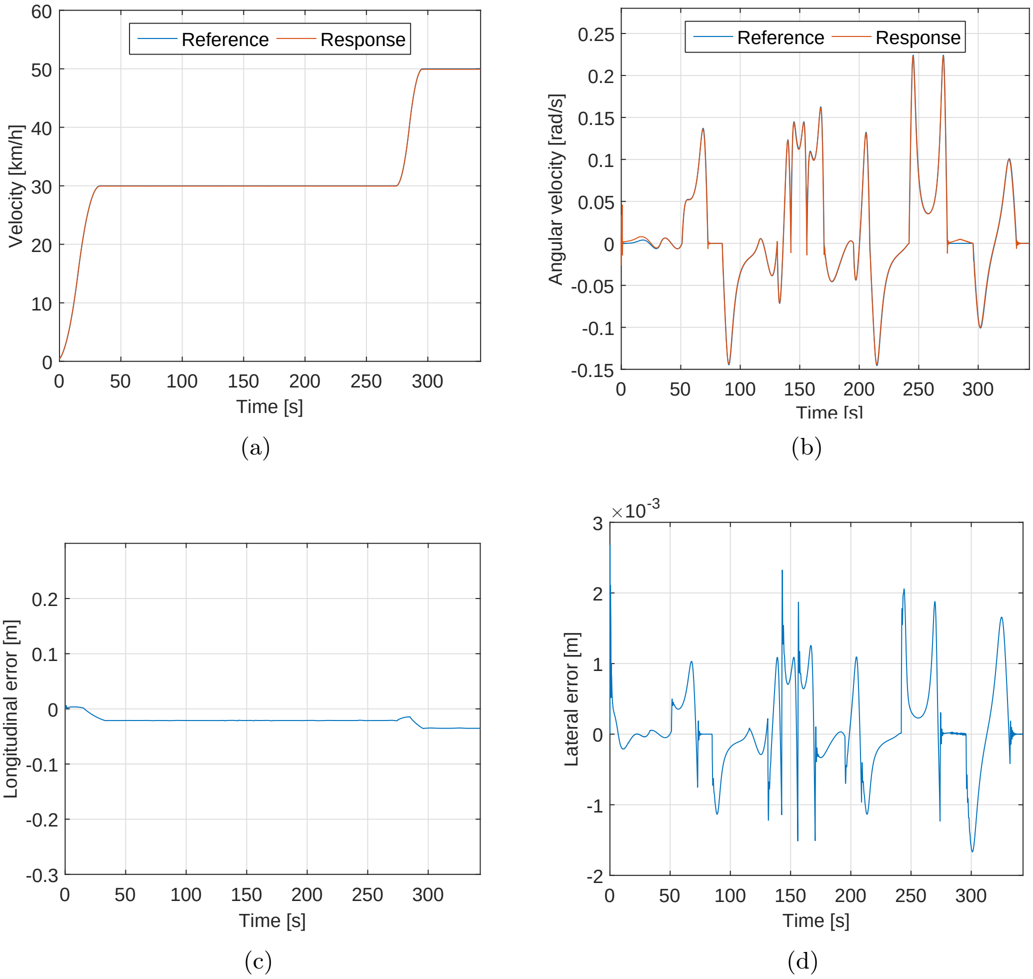


FIGURE 5.11: Results on SYNTHIA (Test A): Desired path (blue line) and real trajectory (dashed green line)

two different velocity sections. Observe that it is complex to differentiate reference and response signals. Thus, this means that barely exists velocity error and that the velocity tracking is working correctly. Figure 5.12.b depicts the evolution of the angular velocity. The reference proposed by the trajectory planner is followed quite well allowing the vehicle to mitigate the possible lateral error that can exist. Regarding the position errors (see Figure 5.12.c and d), the longitudinal one reaches a quite low error (around 0.02 meters) and the lateral error evolves in the millimeter scale.

An evaluation of the control in terms of quadratic error has been done obtaining 0.0231 m^2 of MSE (Mean Square Error) for the longitudinal position and an amount of 0.0087 m^2 for the lateral position.



(a)

(b)

(c)

(d)

FIGURE 5.12: Results on SYNTHIA (Test A): (a) Linear velocity reference and response. (b) Angular velocity reference and response. (c) Longitudinal error obtained during the simulation. (d) Vehicle lateral error

5.5.2 Test B

In this test, due to the differences of the scenario with respect to Test A (i.e. velocity and geometry) the scheduling variables have been bounded in the following intervals: $\omega \in [-1.417, 1.417]\frac{rad}{s}$ and $v_d \in [0.1, 5]\frac{m}{s}$. As in Test A, v_d interval starts in $0.1\frac{m}{s}$. The matrices and variables used for adjusting the LQR optimization are

$$Q = \begin{bmatrix} 10 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 1 \end{bmatrix}, R = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \gamma = 0.0001 \; .$$

The set of control parameters found for this scenario are presented in Table 5.2 for each bounding box vertex. The results of applying the Lyapunov control technique adjusted by means of the LQR-LMI approach are shown in Figure 5.13 and 5.14.

	v_d	ω	k_1	k_2	k_3
i=1	0.1	-1.42	0.27	0.23	0.31
i=2	5	-1.42	0.78	1.07	1.2
i=3	0.1	1.42	0.27	0.23	0.31
i=4	5	1.42	0.78	1.07	1.2

TABLE 5.2: Test B - Control parameters for each vertex of the bounding box (see Figure 5.7)

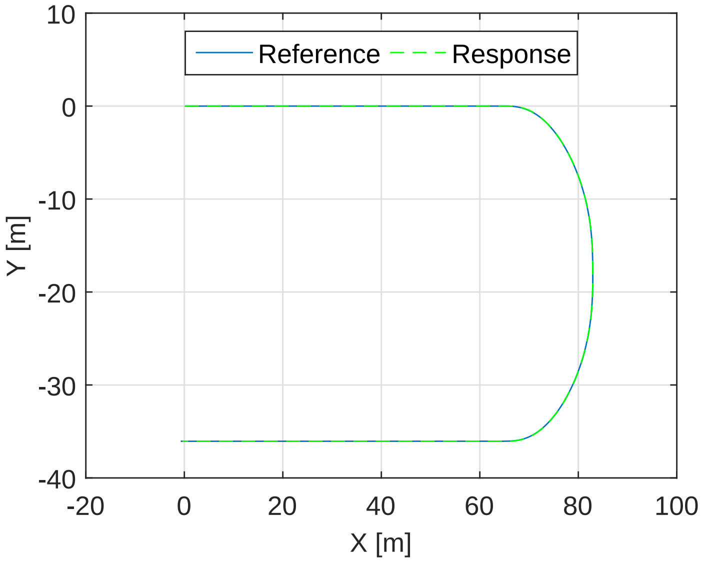
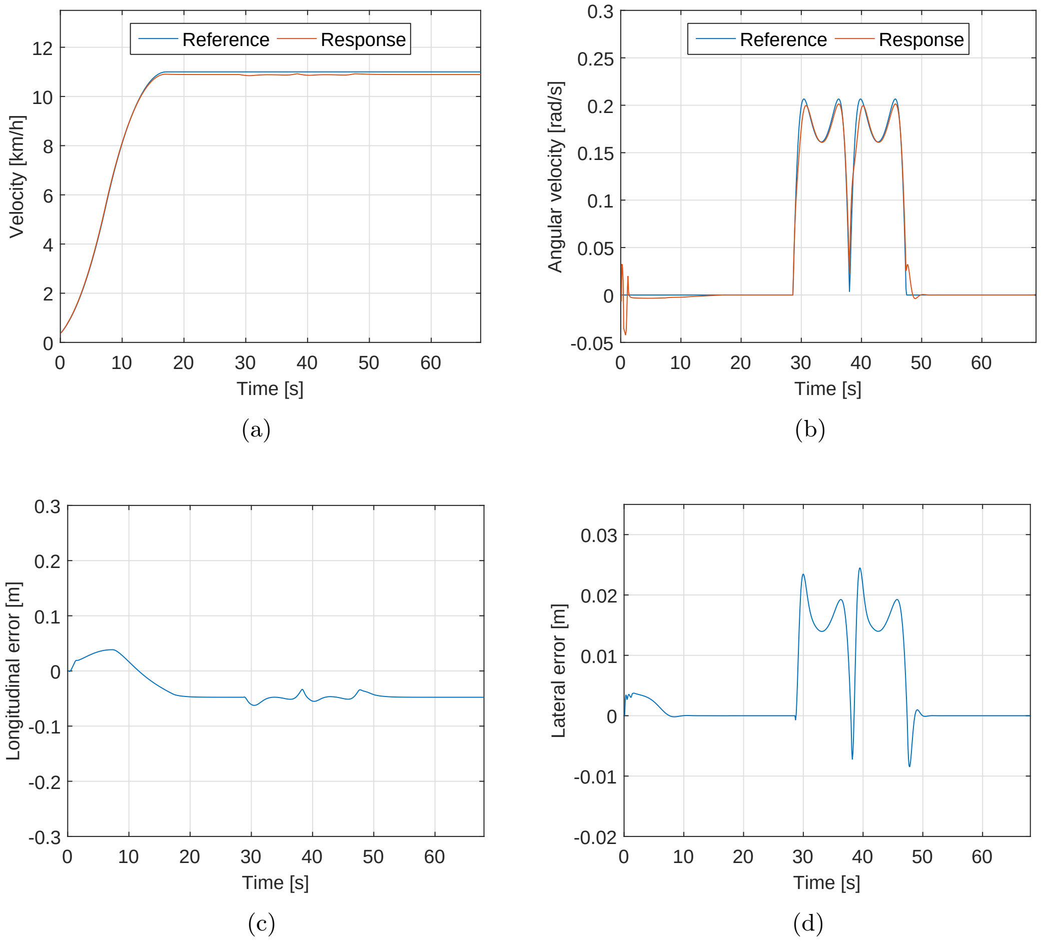


FIGURE 5.13: Results on SYNTHIA (Test B): Desired path (blue line) and real trajectory (dashed green line)

Figure 5.13 shows the proposed circuit and the trajectory result. This circuit has been proposed with the idea of simulating the real experimental test.

From Figures 5.14.a and b, it can be seen that both velocity references are followed presenting low levels of error. However, the relevant signals when performing a trajectory tracking task are the position errors. They are depicted in Figures 5.14.c and d. Observe that the longitudinal error is in the range of ± 0.05 m and the lateral error evolves in a range of few centimeters.

Furthermore, in order to value the simulation an evaluation of the control performance in terms of quadratic error has been done. The obtained results are 0.0269 m^2 of MSE for the longitudinal position and an amount of 0.0053 m^2 of MSE for the lateral position. Note that due to the differences in geometry and velocities on both circuits (Tests A and B) they cannot be compared. Test A develops at higher velocity than Test B, but has wider and longer curves than Test B achieving in this manner a lower lateral acceleration in the curves. Hence, we can affirm that lateral error is function of the lateral acceleration that the vehicle suffers when arrives to a curve and that the results are consistent.



(a)

(b)

(c)

(d)

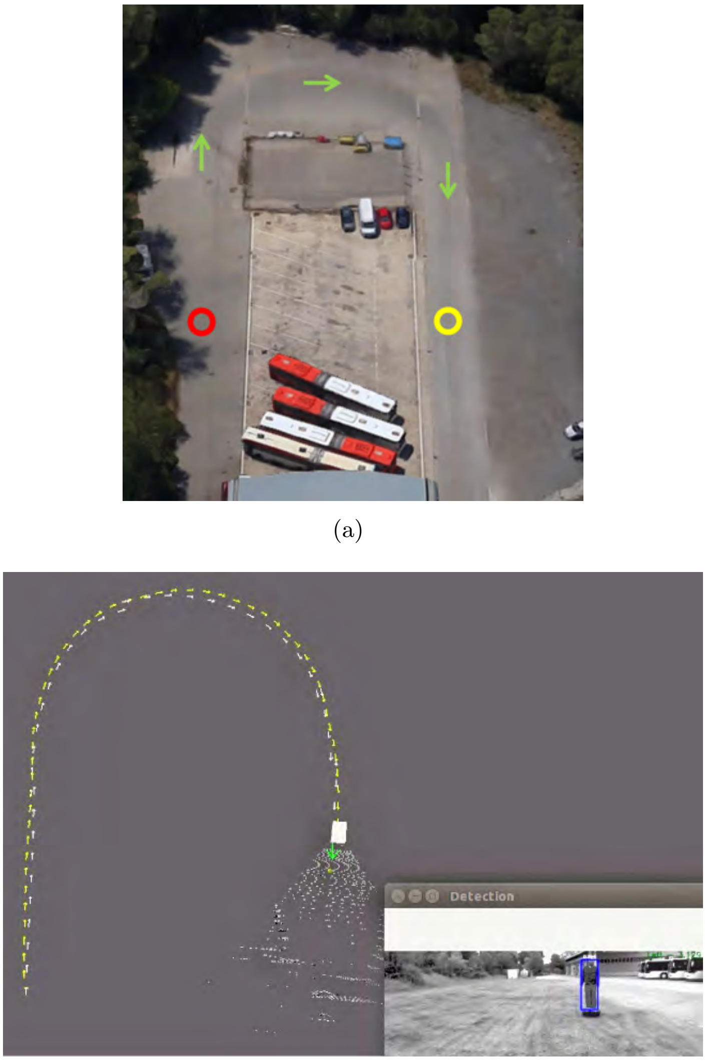
FIGURE 5.14: Results on SYNTHIA (Test B): (a) Linear velocity reference and response. (b) Angular velocity reference and response. (c) Longitudinal error obtained during the simulation. (d) Vehicle lateral error

Remark 5.2. The algorithms in SYNTHIA have been programmed by using *C#* for Unity environment, linking at the same time functions programmed in C++ over Visual Studio 2012.

Remark 5.3. The LMIs have been solved with YALMIP and SeDuMi solvers, while the optimization problem (5.19) has been solved by using Matlab genetic algorithm "*ga*".

5.6 Experimental results

Once planning and control systems have been evaluated and the simulations have proved to be satisfactory, a real scenario is used to validate their integrated applicability. The validation results of the complete vehicle behaviour over a real scenario are presented (see Figure 8.6.a). The test consists in starting from an initial position (red circle), reaching a constant velocity while following the desired trajectory and finally stopping in front of a detected pedestrian (yellow circle). The scenario where the test has been performed is a geometrically simple circuit.



(a)

(b)

Figure 5.15: In field test: (a) Space used to perform tests. (b) Visualization in real time of testing results in ROS

The results of perception, localization and obstacle avoidance modules have been omitted in this work. However, they are used in the real validation (e.g. obstacle avoidance module is in charge of varying the position of the next global way-point if an obstacle is detected). The experimental controller has been adjusted with the same parameters than in simulation (Test B). They can be seen in Table 5.2.

Remark 5.4. For this real scenario, the algorithms have been programmed in C++ over a ROS-Ubuntu platform.

The experimental vehicle has as control inputs the linear velocity and the steering angle. Then, the angular control action (ω) computed by the controller is converted into steering angle as

$$\delta = tg^{-1}\left(\frac{(l_f + l_r)\omega}{v}\right) , \tag{5.22}$$

where l_f and l_r are the distance from the CG to the front and rear axes, respectively. The vehicle has two main hardware constraints: the maximum resolution of steering and velocity. On one hand, the steering system has a maximum resolution of two degrees. This is a hard constraint that limits the lateral control. Such limitation produces a nervous steering angle action while trying to achieve the null error. On the other hand, the speed system has a maximum resolution of 1km/h. This issue generates a limitation when controlling the longitudinal behaviour of the vehicle. Thus, it is easy to have an error in the longitudinal speed control and in the longitudinal position.

In Figure 8.6.b, the real trajectory is shown through rviz ROS tool [4]. It shows multiple data in real time: the stereo visualization, the global way points (green arrow), the completed path (white lines) and the real trajectory (yellow lines).

Figure 5.16 shows the resultant behaviour of the system during the experimental test. Figure 5.16.a depicts how the vehicle follow the velocity reference although being not able to eliminate completely the error in steady state. This issue may be caused by the resolution constraint of 1 km/h and the localization errors as well as the controller adjustment. In this experimental test, the control action measured and sent to the steering actuator is the steering angle. Hence, such a steering variable has been depicted in Figure 5.16.b. It can also be seen the problem of resolution, i.e. small jumps of 2 degrees, and how the response does not exactly match the reference produced by the trajectory planner. Figures 5.16.c and 5.16.d show the longitudinal and lateral errors, respectively. These graphs are used to validate the performance of the vehicle. In this case the response is not exactly the expected one. However, the vehicle is stable and

[4] http://wiki.ros.org/rviz

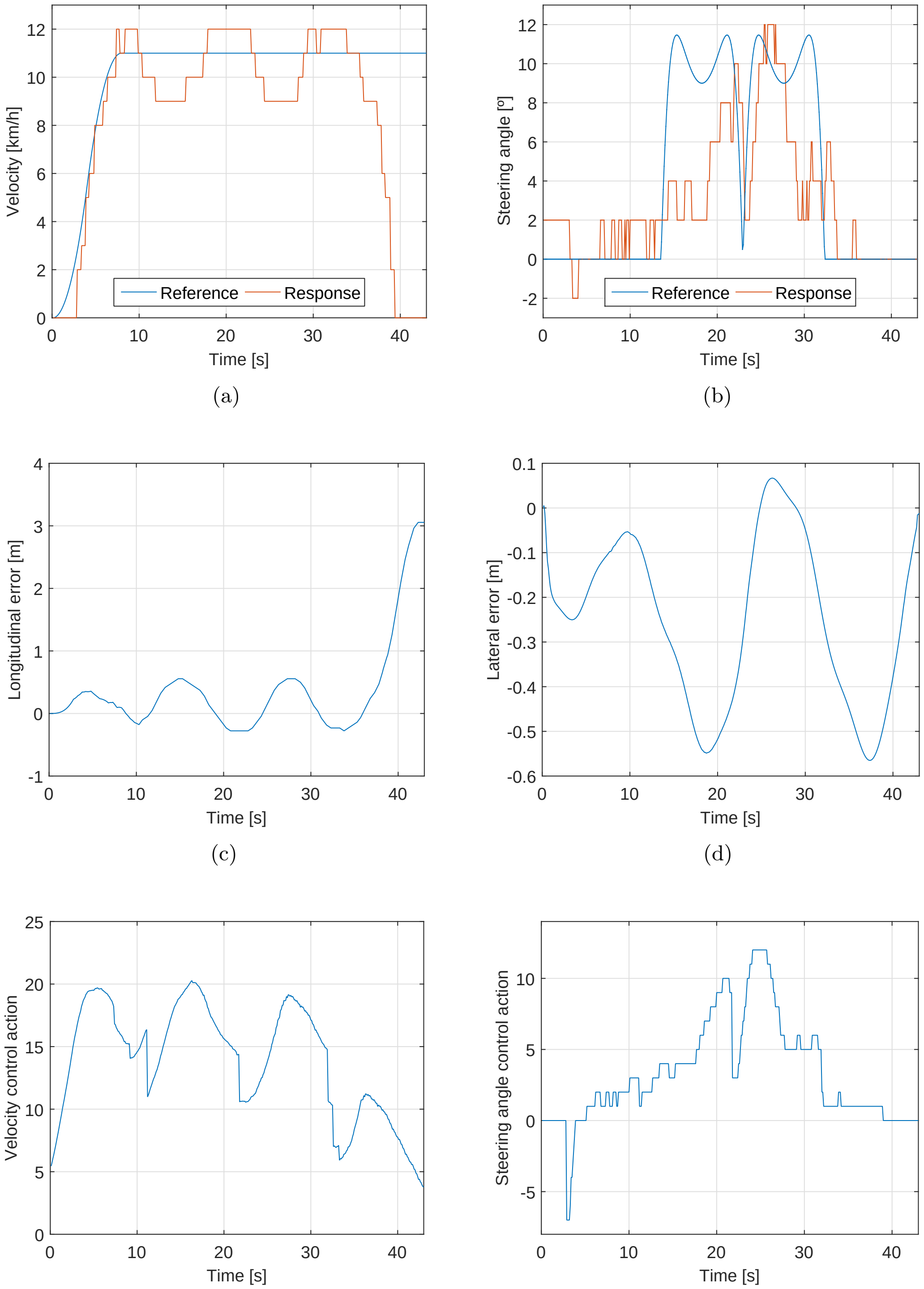
can correct the trajectory at every instant of time in spite of the constraints appointed and localization errors. Finally, Figures 5.16.e and 5.16.f represent the signals sent by the controller to the actuators of the vehicle. The Lyapunov controller produces a sharp velocity action trying to reduce the error, while the steering signal is smoother in spite of the huge difference between the reference and the response. Moreover, as in simulation, an evaluation of the control performance in terms of quadratic error has been done. The obtained results are 0.8178 m^2 of MSE for the longitudinal position and an amount of 0.3099 m^2 of MSE for the lateral position. Comparing these results with the ones obtained in simulation Test B, it can be appreciated the huge difference that exists in between of simulation and reality scenarios.

The objective of this experimental test is to follow the trajectory proposed minimizing the lateral and longitudinal errors at the fastest rate possible. The validation is performed graphically and using the Mean Square Error method. We can conclude that the goals have been achieved although with localization and hardware problems. The vehicle is able to go through the way-points being stable and mitigating the errors. The test was performed 50 times and the goal was achieved in 41 of them. The main problems are due to localization drift. An example video of the vehicle performing in SYNTHIA and real scenarios can be seen in YouTube[5].

5.7 Conclusions

In this chapter, a non-linear control strategy based on Lyapunov theory has been introduced for solving the control problem of autonomous vehicle guidance. This chapter has also proposed an iterative algorithm for adjusting the parameters of the non-linear controller to achieve not only stability but also performance specifications. This algorithm relies on a LQR-LMI based strategy using a LPV representation of the closed-loop kinematic error model. Furthermore, such an adjustment is complemented by adding a restriction between dominant dynamic poles and dominant kinematic poles for holding a correct physical behaviour. The obtained LPV-Lyapunov controller jointly with a trajectory generation module are in charge of moving the vehicle. It has been presented the performance of the vehicle in simulation obtaining satisfactory results, and it has been

[5] https://www.youtube.com/watch?v=K0omhJXawTo



(a)

(b)

(c)

(d)

(e)

(f)

Figure 5.16: Result of relevant vehicle variables during one of the experimental tests

achieved the expected goal of moving autonomously from a starting point to a final point in a comfortable way in a real test.

Chapter 6

GS-LPV control including friction force estimation and compensation mechanism

The content of this chapter is based on the following work:

- [Alcalá, Eugenio, et al., 2018.B]. Alcalá, E., Puig, V., Quevedo, J., & Escobet, T. (2018). Gain-scheduling LPV control for autonomous vehicles including friction force estimation and compensation mechanism. *IET Control Theory & Applications*, 12(12), 1683-1693.

6.1 Introduction

Friction forces are one of the most significant disturbances that affect a vehicle. These forces are dependent on the type of materials involved in the wheel-road contact. The most common, rubber-asphalt, generates a magnitude of friction force that can be drastically altered if the vehicle suddenly crosses a wet or even frozen area. For this reason, the estimation and subsequent compensation is of great interest in the field of autonomous driving. Different works have stated the importance of estimating the friction force [Grip, Havard Fjaer, et al., 2008, Rajamani, Rajesh, et al., 2011, Khaleghian, Seyedmeysam, Anahita Emami, and Saied Taheri., 2017].

Several approaches have been recently studied in the state-disturbance estimation area for autonomous vehicles including observer-based [Yi, Kyongsu, Karl Hedrick, and Seong-Chul Lee., 1999, Svendenius, Jacob., 2007] and statistical [Dakhlallah, Jamil, et al., 2008] methods. Among observer-based techniques, the unknown input observer (UIO) is gaining interest. This strategy has been widely used for fault detection and isolation as, e.g. in Rotondo, Damiano, et al. [2018]. These type of observers allow to estimate the states of a system as well as disturbances or unmodeled uncertainty in the system. In the automotive field, authors in Wang, Yan, David M. Bevly, and Shih-ken Chen. [2013] use the UIO method to estimate the longitudinal tire force of the vehicle.

This chapter constitutes an improvement with respect to the previous chapter. A solution for the integrated longitudinal and lateral dynamic control problem of urban autonomous vehicles is presented. This solution is based on a gain-scheduling LPV control approach combined with the use of an UIO for estimating the vehicle states and friction force. The contribution of this article is three-fold. First, we present a novel LMI formulation for the LPV-UIO observer design based on an optimal approach. It follows the control-observer duality principle and introduces also a constraint for the decay rate. Second, a friction force compensation mechanism based on the estimation provided LPV-UIO is proposed for reducing the control effort and increasing the response when such a disturbance actuates. Third, we present the design of an LPV-LQR approach for solving the integrated lateral and longitudinal control for autonomous vehicles. This approach is based on a cascade design of the the kinematic and dynamic controllers. Such a cascade scheme is based on the idea that the dynamic closed-loop behaviour is designed to be faster than the kinematic closed-loop one.

The proposed scheme is integrated with a trajectory generation module and tested in a simulated scenario. A comparative study is also presented considering the cases that the friction force estimation is used or not to show its usefulness.

6.2 Vehicle description and control-oriented modeling

The vehicle used for testing the strategies addressed in this chapter is the Tazzari Zero vehicle [Tazzari Zero., 2006] (see Figure 5.1). Notice that it is the same vehicle than in the last chapter.

The use of a cascade control scheme implies two model-based control techniques that consequently, implies two models. On one hand, the outer layer is governed by a kinematic behaviour (see Section 3.2). This is based on the velocity vector movement in order to compute longitudinal and lateral velocities referenced to a global inertial frame. External forces are not considered in this case. On the other hand, the inner layer is governed by a dynamic behaviour (see Section 3.3). The motion is generated by applying forces over the driven wheels and mass, inertial and tire parameters are considered.

For a control design purpose, the LPV representation is used to transform non-linear formulations into pseudo-linear formulations. Both kinematic and dynamic LPV models employed in this chapter have been presented in (4.2) and (4.11), respectively. Table (6.1) presents the characteristic vehicle parameters used in the models.

Table 6.1: Kinematic and dynamic model parameters

Parameter	Description	Value
l_f	Distance from CG to front axle	0.758 m
l_r	Distance from CG to rear axle	1.036 m
m	Vehicle mass	683 kg
I	Vehicle yaw inertia	560.94 $kg\ m^2$
C_d	Drag coefficient	0.36
A_r	Vehicle frontal area	1.91 m^2
ρ	Air density at 25°C	1.184 $\frac{kg}{m^3}$
μ_o	Nominal friction coefficient	0.5
C_f	Front tire stiffness coefficient	25000 $\frac{N}{rad}$
C_r	Rear tire stiffness coefficient	25000 $\frac{N}{rad}$

Remark 6.1. Due to the unknown real static friction force, i.e. μmg in (3.17), a nominal constant friction coefficient (μ_o) is considered in the friction force part of the dynamic model for control and estimation as $\mu_o mg$.

6.2.1 Dynamic LPV model reformulation

At this point, the polar dynamic LPV representation (4.11) is pretended to be used. However, such a model is not completely ready for performing good control behaviour in the application that will be proposed later and thus, some changes have to be made before.

First, considering an unknown friction force disturbance, denoted as F_{fr}, acting on the vehicle then, the state-space model for the polar dynamic representation (4.11) is reformulated as

$$\dot{x} = A(\delta, v, \alpha)x + B(\delta, v, \alpha)u_D + EF_{fr} \ , \tag{6.1a}$$

where disturbance input matrix is

$$E = \begin{bmatrix} \frac{-1}{m} \\ 0 \\ 0 \end{bmatrix} . \tag{6.1b}$$

Note that the real friction force actuating in the vehicle wheels is the sum of the nominal one and the estimated variation.

Second, the stability analysis of an open-loop polytopic LPV system using LMIs consists in the resolution of a problem of as many LMIs as vertices have the polytope. However, when the problem consists in looking for a set of controllers for each vertex of the polytope, i.e. polytopic control design, the complexity of such a problem size and the degree of conservatism grow significantly [Apkarian, Pierre, Pascal Gahinet, and Greg Becker., 1995], sometimes even saturating the problem. In Tanaka, Kazuo, and Hua O. Wang. [2004], a more complex LMI-based formulation is presented that allows working with a matrix B of the system dependent on variant parameters. On the other hand, Apkarian, Gahinet and Becker proposed a method based on the transfer of such variant parameters from matrix B to matrix A through the extension of new states [Apkarian, Pierre, Pascal Gahinet, and Greg Becker., 1995].

In this chapter, the Apkarian proposal has been used and the LPV system (6.1) can be further simplified by incorporating the parameter dependency of matrix B into matrix A leading to

$$\dot{\tilde{x}} = \tilde{A}(\delta, v, \alpha)\tilde{x} + \tilde{B}u_f \ . \tag{6.2}$$

To do so, the system is augmented by adding a fast dynamic filter as suggested by Apkarian, Pierre, Pascal Gahinet, and Greg Becker. [1995] in the form

$$\dot{x}_f = A_f x_f + B_f u_f \tag{6.3}$$

$$\begin{bmatrix} \dot{\delta} \\ \dot{a} \end{bmatrix} = \begin{bmatrix} -\psi & 0 \\ 0 & -\psi \end{bmatrix} \begin{bmatrix} \delta \\ a \end{bmatrix} + \begin{bmatrix} \psi & 0 \\ 0 & \psi \end{bmatrix} \begin{bmatrix} u_F \\ u_\delta \end{bmatrix} ,$$

where ψ represents the filter gain, u_F is the new longitudinal behaviour input and u_δ is the new lateral behaviour input. Note that, this new added states have fast dynamics and will not disturb the dynamic model (6.1).

Then, the system (6.1) is transformed into a new fifth order system with state and input vectors as

$$\tilde{x} = \begin{bmatrix} v \\ \alpha \\ \omega \\ \delta \\ a \end{bmatrix} , \; u_f = \begin{bmatrix} u_\delta \\ u_a \end{bmatrix} , \tag{6.4a}$$

and matrices $\tilde{A}$, $\tilde{B}$ and $\tilde{E}$ as

$$\tilde{A}(\delta, v, \alpha) = \begin{bmatrix} A_{11} & A_{12} & A_{13} & B_{11} & B_{12} \\ 0 & A_{22} & A_{23} & B_{21} & B_{22} \\ 0 & A_{32} & A_{33} & 0 & B_{32} \\ 0 & 0 & 0 & -\psi & 0 \\ 0 & 0 & 0 & 0 & -\psi \end{bmatrix} , \tag{6.4b}$$

$$\tilde{B} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ \psi & 0 \\ 0 & \psi \end{bmatrix} , \; \tilde{E} = \begin{bmatrix} \frac{1}{m} \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} . \tag{6.4c}$$

At this point, the model still presents some features that will difficult the control design task. One of them is that the input $\delta = 0$ has been identified as a singular point. Hence, the third step is to avoid it and a change of variable has been applied by shifting the δ interval

$$\delta \in \left[\underline{\delta}, \overline{\delta}\right] \rightarrow \sigma \in \left[\underline{\delta} - \varepsilon, \overline{\delta} + \varepsilon\right] \; , \tag{6.5}$$

converting σ into the new scheduling variable and being ε a constant value greater than $\overline{\delta}$.

In addition to all these arrangements, it was found that the angular velocity channel lacks integral action, thus leading to a steady state error. Then, the final step consists in adding such an integral action through the controller. Then, a new state (i_p) has been added as the integral of the state ω

$$\dot{i}_p = -\omega \; . \tag{6.6}$$

Therefore, starting from (6.4), taking into account these considerations and denoting the scheduling variables as $\sigma, v, \alpha \in \mathbb{R}$, the vehicle dynamic LPV model can be expressed as follows

$$\dot{x}_D = A_D(\sigma, v, \alpha)x_D + B_D u_f + E_D F_{fr} \; , \tag{6.7a}$$

with state and input vectors

$$x_D = \begin{bmatrix} v \\ \alpha \\ \omega \\ \sigma \\ a \\ i_p \end{bmatrix} \; , \; u_f = \begin{bmatrix} u_\delta \\ u_a \end{bmatrix} \; , \tag{6.7b}$$

and matrices A_D, B_D and E_D as

$$A_D(\sigma, v, \alpha) = \begin{bmatrix} A_{11} & A_{12} & A_{13} & B_{11} & B_{12} & 0 \\ 0 & A_{22} & A_{23} & B_{21} & B_{22} & 0 \\ 0 & A_{32} & A_{33} & 0 & B_{32} & 0 \\ 0 & 0 & 0 & -\psi & 0 & 0 \\ 0 & 0 & 0 & 0 & -\psi & 0 \\ 0 & 0 & -1 & 0 & 0 & 0 \end{bmatrix} \; , \tag{6.7c}$$

$$B_D = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ \psi & 0 \\ 0 & \psi \\ 0 & 0 \end{bmatrix} \; , \; E_D = \begin{bmatrix} \frac{-1}{m} \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \; . \tag{6.7d}$$

The model (6.7) will be used for designing the dynamic state feedback control. Hereafter, in order to simplify the notation, the scheduling variables dependency of state space matrices is omitted.

6.3 LPV-UIO Design

Owing to the lack of available sensors for measuring all states, i.e. there is no one that measures the slip angle, the design of a state estimator has been considered in this section (see Figure 6.1). The LPV-UIO tackles the problem of estimating both the dynamic states and the friction force affecting the vehicle. Such an estimator has been designed following an optimal approach exploiting the duality between the LQR and Kalman filter approaches. In a recent work [Pletschen, Nils, and Klaus J. Diepold., 2017], Pletschen and Diepold present a TS Kalman filter strategy with a decoupled stability and performance methodology. Unlike that approach, this chapter presents an LPV Kalman filter with both, stability and performance criteria integrated in a single design procedure.

6.3.1 System description

The proposed UIO estimation scheme is developed for LPV systems affected by external disturbances. The measurement model for the dynamical one presented in (6.1) considering the available sensors leads to consider the following output matrix

$$C = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \tag{6.8}$$

due to the lack of measuring of dynamic states as the slip angle and the estimated vector state is denoted as

$$\hat{\boldsymbol{x}}_{DO} = \begin{bmatrix} \hat{v} \\ \hat{\alpha} \\ \hat{\omega} \end{bmatrix}. \tag{6.9}$$

The proposed disturbance estimation is based on the UIO approach. Such a procedure is based on computing the difference between the real system and the model used for observation

$$CEF_{fr} = \dot{y} - C\big(A\hat{x}_{DO} + Bu\big) \ . \tag{6.10}$$

Thus, considering $\Gamma = (CE)^{+}$, the friction force disturbance can be obtained as

$$F_{fr} = \Gamma\Big(\dot{y} - C\big(A\hat{x}_{DO} + Bu\big)\Big) \ . \tag{6.11}$$

And consequently, decoupling the considered disturbance, the system (6.1) can be rewritten as follows

$$\dot{\hat{x}}_{DO} = A_o\hat{x}_{DO} + B_ou - E\Gamma\dot{y} \ , \tag{6.12}$$

where

$$A_o = (I - E\Gamma C)A$$
$$B_o = (I - E\Gamma C)B \ .$$

Then, the state estimation will depend on the observer gain L and presents the form

$$\dot{\hat{x}}_{DO} = \big(A_o - LC\big)\hat{x}_{DO} + B_ou - E\Gamma\dot{y} + Ly \ . \tag{6.13}$$

6.3.2 Description of the design method

To design the observer gain L in (6.13), the polytopic approximation of system (6.1) is used obtaining

$$A(\zeta) = \sum_{i=1}^{N}\mu_i(\zeta)A_i \ , \tag{6.14}$$

where A_i are the dynamic matrics at the vertexes of the defined polytope, N is the number of polytopic vertices, ζ is the vector containing the scheduling variables and the vertex membership function $\mu_i(\zeta)$ is given by

$$\mu_i(\zeta) = \prod_{j=1}^{n_\zeta}\xi_{ij}(\eta_0^j, \eta_1^j) \ , \quad \forall i = 1, ..., N \ , \tag{6.15}$$

$$\begin{aligned} \eta_0^j &= \frac{\overline{\zeta_j} - \zeta_j(t)}{\overline{\zeta_j} - \underline{\zeta_j}} \\ \eta_1^j &= 1 - \eta_0^j \ , \end{aligned} \tag{6.16}$$

where each variable ζ_j is known and varies in a defined interval $\zeta_j \in \left[\underline{\zeta_j}, \overline{\zeta_j}\right] \in \mathbb{R}$, n_ζ is the number of scheduling variables and $\xi_{ij}(\cdot)$ corresponds with the function that performs the N possible combinations. In addition, next conditions must be satisfied

$$\sum_{i=1}^{N} \mu_i(\zeta) = 1, \;\; \mu_i(\zeta) \geq 0, \; \forall \zeta \in \Theta \; . \tag{6.17}$$

Then, the observer gain is given by

$$L(\zeta) = \sum_{i=1}^{N} \mu_i(\zeta) L_i \; , \tag{6.18}$$

where L_i are obtained using the following proposition that provides an optimal design based on the Riccati equations of the Kalman filter.

Proposition 6.2. *Let the observer tuning parameters $Q = Q^T \geq 0$, $R = R^T > 0$, the optimal performance bound $\gamma > 0$, the decay rate $\lambda > 0$, the output matrix C in (7.10) and the matrices A_i in (6.14). Then, the polytopic observer gains in (6.18) are obtained by finding Y and W_i satisfying the following LMIs*

$$\begin{aligned} &\begin{bmatrix} YA_i + A_i^T Y - W_i C - C^T W_i^T + Y2\lambda & Y(Q^{\frac{1}{2}})^T & W_i \\ Q^{\frac{1}{2}} Y & -I & 0 \\ W_i^T & 0 & -R^{-1} \end{bmatrix} < 0 \; , \\ &\begin{bmatrix} \gamma I & I \\ I & Y \end{bmatrix} > 0 \; , \qquad\qquad \forall i = 1, ..., N \; , \end{aligned} \tag{6.19}$$

considering $Y = Y^T > 0$ and applying the transformation $L_i = Y^{-1} W_i$.

Proof. Considering the Kalman filter Ricatti equation for every vertex of the polytopic model (6.14), the following inequality is obtained

$$\dot{P} = (A_i - L_i C)P + P(A_i - L_i C)^T + Q + L_i R L_i^T < 0 \; ,$$

where $Q \in \mathbb{R}^{s \times s}$, $R \in \mathbb{R}^{n \times n}$, $P \in \mathbb{R}^{s \times s}$ and $L_i \in \mathbb{R}^{s \times n}$. At this point, we introduce an extra performance term, i.e. the decay rate (λ), for ensuring a fast dynamic response of the observer

$$(A_i - L_i C)P + P(A_i - L_i C)^T + 2\lambda P + Q + L_i R L_i^T < 0.$$

Then, by multiplying first each term of the last inequality by $Y = P^{-1}$ from the left-hand and the right-hand sides and then by introducing $W_i = YL_i$, the following inequality is obtained

$$YA_i - W_iC + A_i^T - C^TW_i^T + Y2\lambda + YQY + W_iRW_i^T < 0.$$

From here, we reformulate the inequality in order to use the Schur complement

$$YA_i + A_i^TY - W_iC - C^TW_i^T + Y2\lambda$$

$$-\begin{bmatrix} Y(Q^{\frac{1}{2}})^T & W_i \end{bmatrix} \begin{bmatrix} -I & 0 \\ 0 & -R \end{bmatrix} \begin{bmatrix} Q^{\frac{1}{2}}Y \\ W_i^T \end{bmatrix} < 0.$$

Applying such a complement to this inequality, the first LMI of (6.19) is obtained. The second LMI starts from bounding the Lyapunov matrix

$$P < \gamma I.$$

Applying first the change of variable $Y = P^{-1}$ and then the Schur complement

$$\gamma I - IY^{-1}I > 0,$$

$$\begin{bmatrix} \gamma I & I \\ I & Y \end{bmatrix} > 0 \ ,$$

where the condition $P > 0$ is included. □

Note that the problem has solution if and only if there exist $Y \in \mathbb{R}^s$ and $W_i \in \mathbb{R}^{s\times p}$, being s the number of states and p the number of measurable states. Matrices Q and R represent the process noise covariance and the sensor noise covariance, respectively.

Remark 6.3. Note that, unlike the control design problem where the stability criteria does depend on the input matrix (B), the observer design process by duality depends on the output matrix (C) and not on the input matrix. That is why it is not necessary to increase the state-space model as presented in (6.2.1) for the design of observers.

6.3.3 Dynamic LPV-UIO design

The dynamic LPV-UIO addresses the problem of estimating the dynamic state vector in (6.9) as well as estimating the friction force acting over the vehicle. At this point, the LPV model developed in (6.1) is used for solving the Proposition 1 using the output matrix (7.10). The scheduling variables, i.e. σ, v and α, are bounded in the following intervals

$$\sigma \in [0.087, 0.96] rad \qquad \text{and} \qquad v \in [1, 18] \; \frac{m}{s}$$

$$\alpha \in [-0.1, 0.1] \; rad \; .$$

The proposed design matrices and parameters are: $R = 0.01 I_{2\times2}$, $Q = 0.01 I_{3\times3}$, $\gamma = 0.1$ and $\lambda = 12$. The solution of such a Proposition 6.2 returns the polytopic observer gains (see Table 6.2). Then, at every time step, the interpolated observer gain is obtained by means of (6.18).

TABLE 6.2: Dynamic controllers L_i for each one of the vertices of the polytope

$L_1 = 10^3 \begin{bmatrix} 0.4183 & -0.0000 \\ 0.0000 & 6.7870 \\ -0.0000 & 0.9276 \end{bmatrix}$,	$L_2 = 10^3 \begin{bmatrix} 0.4183 & -0.0000 \\ -0.0000 & 6.7660 \\ -0.0000 & 0.9257 \end{bmatrix}$
$L_3 = 10^3 \begin{bmatrix} 0.4183 & -0.0000 \\ -0.0000 & 7.4632 \\ -0.0000 & 0.9985 \end{bmatrix}$,	$L_4 = 10^3 \begin{bmatrix} 0.4183 & -0.0000 \\ -0.0000 & 7.2325 \\ -0.0000 & 0.9765 \end{bmatrix}$
$L_5 = 10^3 \begin{bmatrix} 0.4183 & -0.0000 \\ -0.0000 & 6.8785 \\ -0.0000 & 1.1487 \end{bmatrix}$,	$L_6 = 10^3 \begin{bmatrix} 0.4183 & -0.0000 \\ -0.0000 & 6.8547 \\ -0.0000 & 1.1467 \end{bmatrix}$
$L_7 = 10^3 \begin{bmatrix} 0.4183 & -0.0000 \\ -0.0000 & 6.6182 \\ -0.0000 & 0.7658 \end{bmatrix}$,	$L_8 = 10^4 \begin{bmatrix} 0.0418 & -0.0000 \\ -0.0000 & 1.1831 \\ -0.0000 & 0.1677 \end{bmatrix}$

6.4 Control Design using LPV Approach

The automatic control strategy addresses the problem of generating an appropriate vehicle behaviour from a desired reference. In this work, two cascade state feedback LPV controllers are proposed for controlling appropriately the behaviour of the vehicle (see Figure 6.1). Furthermore, a trajectory planner [Bianco, CG Lo, Aurelio Piazzi, and Massimo Romano., 2004] is used being in charge of providing the correspondent position and velocities references to the kinematic controller.

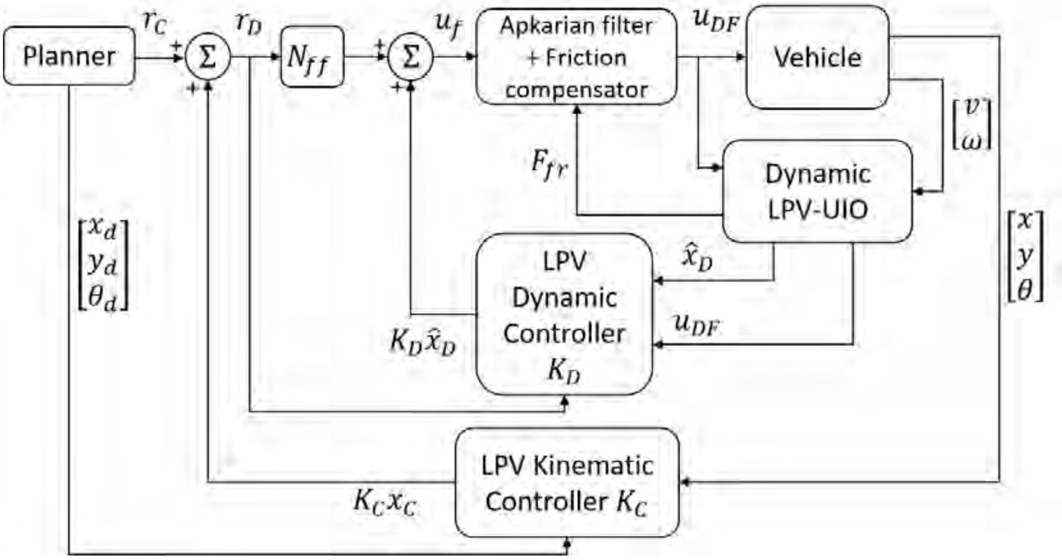

Planner
r_C
r_D
N_{ff}
u_f
Apkarian filter + Friction compensator
u_{DF}
Vehicle
$\begin{bmatrix} v \\ \omega \end{bmatrix}$
F_{fr}
Dynamic LPV-UIO
$\begin{bmatrix} x_d \\ y_d \\ \theta_d \end{bmatrix}$
$\hat{x}_D$
LPV Dynamic Controller K_D
$K_D\hat{x}_D$
u_{DF}
$\begin{bmatrix} x \\ y \\ \theta \end{bmatrix}$
LPV Kinematic Controller K_C
$K_C x_C$

FIGURE 6.1: Complete autonomous driving control scheme with two LPV controllers and a LPV-UIO with friction force compensator. Note that both K_D and K_C have the negative sign embedded

In this approach, a cascade methodology is employed where the internal and fast loop corresponds to the dynamic control and the external one to the kinematic control. On one hand, the kinematic control (K_C in Figure 6.1) is in charge of computing smooth control actions (linear and angular velocities) such that the vehicle is capable of achieving the required speed, position and orientation at the next local way-point. On the other hand, the dynamic control strategy (K_D in Figure 6.1) allows the vehicle to follow the angular and linear velocity references provided by the kinematic control loop. To this aim, the dynamic control generates forces to the rear wheels and a steering angle signal for the front wheels.

6.4.1 Description of the design method

Note that the design overview has been developed for the case of the dynamic model in (6.7). Furthermore, the same procedure is used for designing the kinematic controller by just considering the kinematic model in (4.2).

To design the controller K_D, the polytopic approach of the system in (6.7) is used in the same way it was presented in (6.14). Then, using (6.15) and (6.16), the controller gain is obtained by means of

$$K_D(\zeta) = \sum_{i=1}^{N} \mu_i(\zeta) K_{D_i} \ , \tag{6.20}$$

where K_{D_i} are obtained using the following proposition which presents a LMI based formulation for solving the LPV-LQR problem.

Proposition 6.4. *Given the LQR parameters $Q = Q^T \geq 0$, $R = R^T > 0$, the optimal performance bound $\gamma > 0$, the decay rate $\eta > 0$ and the matrices A_{D_i} obtained using (6.14). Then, the polytopic control gains in (6.20) are obtained by finding P and W_i satisfying the following LMIs*

$$\begin{aligned}
&A_{D_i}P + PA_{D_i} - (B_D W_i)^T - B_D W_i + 2\eta P < 0 \\
&\begin{bmatrix} W_i^T R W_i & P(Q^{\frac{1}{2}})^T \\ P & (-Q^{\frac{1}{2}})^{-1} \end{bmatrix} < 0 \ , \qquad \forall i = 1, ..., N \ , \\
&0 < P < \gamma
\end{aligned} \tag{6.21}$$

and applying the transformation $K_{D_i} = W_i P^{-1}$.

Proof. Considering the LQR Riccati equation for every vertex of the polytopic model (6.7), the following inequality is obtained

$$(A_i - B_D K_{D_i})^T P + P(A_i - B_D K_{D_i}) + Q + K_{D_i}^T R K_{D_i} < 0 \ .$$

At this point, we split the last inequality in two new ones. The first resultant is the pure Lyapunov stability term

$$(A_{D_i} - B_D K_{D_i})^T P + P(A_{D_i} - B_D K_{D_i}) < 0 \ ,$$

where, to ensure a fast dynamic response of the controller, a decay rate term (η) has been added obtaining

$$(A_{D_i} - B_D K_{D_i})^T P + P(A_{D_i} - B_D K_{D_i}) + 2\eta P < 0$$

and applying $K_{D_i} = W_i P^{-1}$ we obtain the following LMI

$$A_{D_i} P + P A_{D_i} - (B_D W_i)^T - B_D W_i + 2\eta P < 0 \ .$$

The second LMI establishes the LQR performance. Then, multiplying by the left and right the second part of the Riccati equation we achieve

$$P(Q^{\frac{1}{2}})^T Q^{\frac{1}{2}} P + P K_{D_i}^T R K_{D_i} P < 0$$

and applying the change $K_{D_i} = W_i P^{-1}$ it is converted to

$$P(Q^{\frac{1}{2}})^T Q^{\frac{1}{2}} P + W_i^T R W_i < 0 \ .$$

At this point, by rearranging the elements we obtain

$$W_i^T R W_i - P(Q^{\frac{1}{2}})^T (-Q^{\frac{1}{2}}) P < 0$$

and using the Schur complement, the resulting LMI is as follows

$$\begin{bmatrix} W_i^T R W_i & P(Q^{\frac{1}{2}})^T \\ P & (-Q^{\frac{1}{2}})^{-1} \end{bmatrix} < 0 \ .$$

□

Note that the problem has solution if and only if there exist $P \in \mathbb{R}^{s \times s}$, $H \in \mathbb{R}^{n \times n}$ and $W_i \in \mathbb{R}^{n \times s}$, being n the number of control actions and s the number of states. Observe also that decreasing the parameter γ increases the performance of the control loop. Next subsections provide details of the particular control design for the dynamic and kinematic vehicle controllers.

Remark 6.5. Assuming that in the first LMI of (6.21) the matrix B_D was dependent on varying parameters, i.e. $B_D(\zeta)$, this would result in

$$B_D(\zeta)W(\zeta) = \sum_{i=1}^{N} \mu_i(\zeta) \sum_{j=1}^{N} \mu_j(\zeta) B_{D_i} Wi \; . \tag{6.22}$$

This entails the resolution of a much more complex and large problem due to the double polytopic sum. But, it also implies a more conservative degree since the controller of each vertex of the polytope has to be robust against all possible values of $B_D(\zeta)$.

6.4.2 Dynamic LPV control design

The dynamic control addresses the tracking of the linear and angular velocity references of the vehicle by applying force to the wheels and an angle to the front wheels.

At this point, the augmented LPV model previously developed (6.7) is used for solving the Proposition 2. The chosen scheduling variables are σ, v and α which are bounded in the same intervals than the ones presented in Section 6.3.3.

The proposed design matrices Q and R are presented in Table 6.5. Parameters γ and η are set as 0.001 and 3, respectively. The solution of Proposition 6.4 returns the polytopic control gains K_{D_i}. Then, the controller obtained at each control iteration follows the rule presented in (6.20).

The proposed control scheme for this dynamic loop is a state feedback plus feedforward control. The function of feedforward matrix is to make the gain of the system unitary. Such a matrix is computed following the next expression

$$N_{ff} = \left[\tilde{C} \left(-\tilde{B}K - \tilde{A} \right)^{-1} \tilde{B} \right]^{-1} \; , \tag{6.23}$$

where matrices $\tilde{A}$ and $\tilde{B}$ are the ones presented in (6.4), K is a sub-block of K_D in which the last column has been omitted as it is proposed in Franklin, Gene F., J. David Powell, and Michael L. Workman. [1998]. Matrix $\tilde{C}$ is of the form

$$\tilde{C} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \end{bmatrix} . \tag{6.24}$$

Therefore, the complete control law is expressed as

$$u_f = K_D \left[\begin{array}{cccc} \hat{x}_{DO} & \sigma & a & i_p \end{array} \right]^T + N_{ff} r_D \ , \tag{6.25}$$

where the state vector is the one presented in (6.7b) although with the estimated dynamic states shown in (6.9), r_D represents the reference vector which corresponds with the kinematic control signal u_C and u_f is the control input to the Apkarian filter added (6.3).

At this point, the dynamic control action u_{DF} (see Figure 6.1) is computed as the sum of the Apkarian filter result (x_f) and the vector generated by the friction force compensator in (6.11)

$$u_{DF} = \begin{bmatrix} \delta \\ a \end{bmatrix} + \frac{1}{m} \begin{bmatrix} 0 \\ F_{fr} \end{bmatrix} . \tag{6.26}$$

It is interesting to note that, the dynamic model is singular at longitudinal velocity equal to zero (see Section 3.3). As seen in previous sections, linear speed is a scheduling variable and that is why it moves in a defined range. However, the membership of zero and consequently values close to zero to this interval, e.g. $v \in [0, 1]$, may be a problem in the control design causing the obtaining of very large gain controllers in such an interval. The solution involves translating the problematic variable, in this case the linear speed, so as to avoid operating in a range close to zero. Thus, this means that when computing the controller at $v = 0$, we are actually computing the controller at $v = 1$ and using it as we were in $v = 0$. In this way we solve a problem that otherwise it should be addressed with hybrid control techniques.

6.4.3 Kinematic LPV control design

Kinematic control is in charge of controlling the position and orientation by means of actuating over the linear and angular velocities of the vehicle.

At this moment, the kinematic LPV model (4.2) is employed for solving the Proposition 6.4. Three scheduling variables (v_d, ω and θ_e) are bounded in the following intervals

$$v_d \in [1, 18] \ \frac{m}{s} \ , \ \omega \in [-1.417, 1.417] \ \frac{rad}{s} \ , \ \theta_e \in [-0.14, 0.14] rad \ .$$

Table 6.3: Dynamic controllers K_i for each one of the vertexes i of the polytope

$$K_1 = 10^4 \begin{bmatrix} -0.7845 & -0.1760 & -0.0802 & -0.0002 & 0.0027 & -0.3280 \\ 0.0000 & 0.1073 & -0.2441 & 0.0000 & -0.0107 & 0.5575 \end{bmatrix}$$

$$K_2 = 10^4 \begin{bmatrix} -0.6129 & -0.7835 & -0.2095 & -0.0000 & 0.0010 & -1.3048 \\ 0.0003 & 0.1559 & -0.2622 & 0.0000 & -0.0111 & 0.6480 \end{bmatrix}$$

$$K_3 = 10^4 \begin{bmatrix} -1.7823 & -0.1366 & -0.1164 & 0.0001 & 0.0046 & -0.3888 \\ 0.0003 & 0.0988 & -0.2684 & 0.0000 & -0.0111 & 0.5564 \end{bmatrix}$$

$$K_4 = 10^4 \begin{bmatrix} -0.6104 & -0.4180 & -0.2686 & -0.0000 & 0.0044 & -3.1728 \\ 0.0002 & 0.1591 & -0.2621 & 0.0000 & -0.0111 & 0.6489 \end{bmatrix}$$

$$K_5 = 10^4 \begin{bmatrix} -0.6104 & -0.4180 & -0.2686 & -0.0000 & 0.0044 & -3.1728 \\ 0.0002 & 0.1591 & -0.2621 & 0.0000 & -0.0111 & 0.6489 \end{bmatrix}$$

$$K_6 = 10^4 \begin{bmatrix} -0.6104 & -0.4180 & -0.2686 & -0.0000 & 0.0044 & -3.1728 \\ 0.0002 & 0.1591 & -0.2621 & 0.0000 & -0.0111 & 0.6489 \end{bmatrix}$$

$$K_7 = 10^4 \begin{bmatrix} -0.6104 & -0.4180 & -0.2686 & -0.0000 & 0.0044 & -3.1728 \\ 0.0002 & 0.1591 & -0.2621 & 0.0000 & -0.0111 & 0.6489 \end{bmatrix}$$

$$K_8 = 10^4 \begin{bmatrix} -0.6104 & -0.4180 & -0.2686 & -0.0000 & 0.0044 & -3.1728 \\ 0.0002 & 0.1591 & -0.2621 & 0.0000 & -0.0111 & 0.6489 \end{bmatrix}$$

The control design matrices Q and R are presented in Table 6.5 and parameter γ is set as 0.01. Proposition 6.4 returns for this kinematic case the control matrices K_{C_i} for each one of the polytopic vertexes. Then, the controller obtained at each control iteration (K_C) follows the rule presented in (6.20).

It is important to remark that, in this kinematic case, the Proposition 6.4 has a different configuration with respect to the dynamic case. The first inequality of (6.21) is negative and an additional LMI has been added to the Proposition 6.4

$$A_{D_i}P + PA_{D_i} - (B_D W_i)^T - B_D W_i + 2\beta P < 0. \tag{6.27}$$

Being $\beta = 0$, the LMI establishes a threshold for ensuring only stability. Thus, in order to increase the kinematic loop performance β can be increasing while being always positive. Here, it has been set to 0.1. The following state feedback control law has been used for controlling the kinematic behaviour loop

$$u_C = K_C x_C + r_C \ , \tag{6.28}$$

where x_C and r_C are the kinematic state and position reference vectors, respectively, presented in (4.3). Such a reference is provided by a trajectory planner (see Figure 6.1).

Table 6.4: Kinematic controllers K_i for each one of the vertexes i of the polytope

$$K_1 = \begin{bmatrix} 0.7099 & 0.5078 & -0.0238 \\ 0.1899 & 0.3083 & 1.5405 \end{bmatrix} , \qquad K_2 = \begin{bmatrix} 0.7099 & 0.5078 & -0.0238 \\ 0.1899 & 0.3083 & 1.5405 \end{bmatrix}$$

$$K_3 = \begin{bmatrix} 0.7373 & 0.2156 & 0.0158 \\ 0.1792 & 2.0131 & 4.0841 \end{bmatrix} , \qquad K_4 = \begin{bmatrix} 0.7373 & 0.2156 & 0.0158 \\ 0.1792 & 2.0131 & 4.0841 \end{bmatrix}$$

$$K_5 = \begin{bmatrix} 0.7099 & -0.5078 & 0.0238 \\ -0.1899 & 0.3083 & 1.5405 \end{bmatrix} , \qquad K_6 = \begin{bmatrix} 0.7099 & -0.5078 & 0.0238 \\ -0.1899 & 0.3083 & 1.5405 \end{bmatrix}$$

$$K_7 = \begin{bmatrix} 0.7373 & -0.2156 & -0.0158 \\ -0.1792 & 2.0131 & 4.0841 \end{bmatrix} , \qquad K_8 = \begin{bmatrix} 0.7373 & -0.2156 & -0.0158 \\ -0.1792 & 2.0131 & 4.0841 \end{bmatrix}$$

6.5 Simulation Results

The simulation scenario (see Figure 6.2) chosen for testing the automatic control strategy tries to cover different driving situations as acceleration stages and velocity reduction on curves as well as driving on different road conditions, as e.g. asphalt or ice.

To deal with this changing road conditions, the friction force compensation mechanism is used and compared its result with the case of unknown friction.

Considering this information (circuit shape and varying velocity), a trajectory planner is in charge on generating a feasible trajectory by means of using a polynomial curve generation method [Bianco, CG Lo, Aurelio Piazzi, and Massimo Romano., 2004]. This

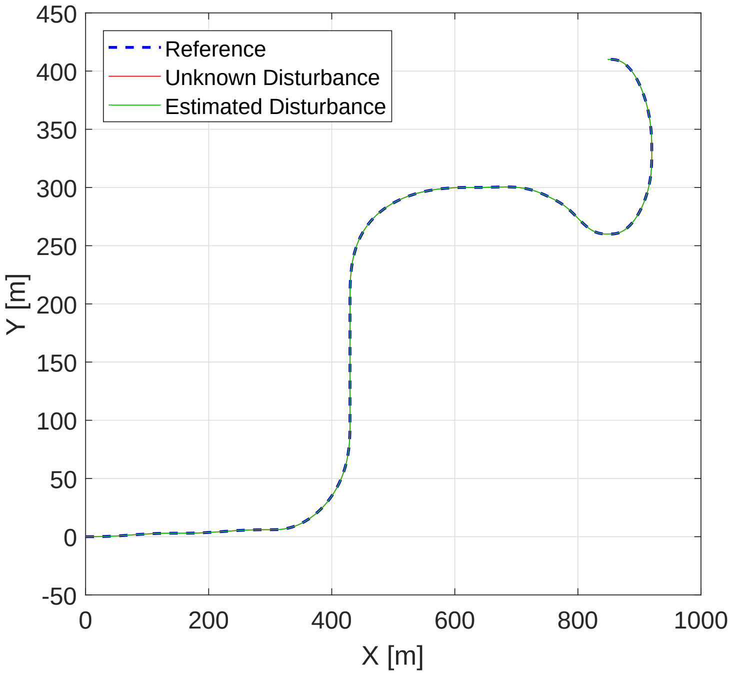


FIGURE 6.2: Proposed circuit for simulation and the result of solving the mixed control problem

method consists on computing continuous and differentiable curves (velocities and accelerations) under an overall constrained vehicle acceleration. Thus, in an offline mode, this planner algorithm generates the linear and angular velocity references as well as desired positions and orientations for the outer control loop (i.e. the kinematic control).

The adjustment of the LPV-LQR parameters (Q, R and γ) is made by means of using the root mean square error (RMSE) approach. This measure allows to find suitable control parameters by minimizing it. Linear velocity, angular velocity and lateral error are chosen by an exhaustive search. Moreover, η and β have been selected with the aim of increasing the performance of the closed loop system. Table 6.5 shows some RMSE results for different control adjustments and the one considered in the simulations (bold row). Note that the observer adjustment was presented in Section 6.3.3.

TABLE 6.5: RMSE obtained for three different configurations of the LQR controllers. The values of Q and R represent the diagonal values of each matrix

RMSE			Kinematic control design		Dynamic control design	
V	ω	Y	$\boldsymbol{Q}$	$\boldsymbol{R}$	$\boldsymbol{Q}$	$\boldsymbol{R}$
0.121	0.035	0.0177	[1, 1, 1]	[0.004, 0.0001]	[0.01, 0.01, 0.01, 0.01, 10, 3000]	[0.005, 0.6]
0.124	0.031	0.0196	[3, 5, 15]	[0.04, 0.01]	[0.01, 0.01, 0.01, 0.01, 100, 30000]	[0.005, 0.6]
0.076	0.0127	0.0213	[10, 3, 15]	[0.4, 0.001]	[0.01, 0.01, 0.01, 0.01, 1000, 90000]	[0.005, 0.6]
0.045	**0.0077**	**0.05**	$\mathbf{[3, 2, 20]}$	$\mathbf{[0.5, 0.001]}$	$\mathbf{[0.01, 0.01, 0.01, 0.01, 100000, 90000]}$	$\mathbf{[0.01, 10]}$

In the tuning process, we have observed that the vehicle lateral behaviour implies a more complex control situation due to the changing reference. Hence, for the dynamic control case, the weight in Q corresponding to the dynamic integral state has been set much larger than the rest. The same occurs in matrix R.

The sample times used in both control loops are 0.1 and 0.01 s for kinematic and dynamic loops, respectively. The control strategy jointly with the trajectory planner are tested in MATLAB environment. Figures 6.2-6.5 show the vehicle results with known and unknown friction disturbance in the simulated circuit. Figure 6.6(a) depicts the applied disturbance profile, i.e. friction force profile depending on the type of road (asphalt and ice). Finally, Figure 6.6(b) represents the location of the closed loop poles of kinematic and dynamic controllers, and the thresholds for the decay rate (η and β) used in their design.

Figure 6.2 depicts the trajectory proposed and the result of both known and unknown disturbance scenarios. Figure 6.3(a) shows the velocity response and that the friction force compensation mechanism works. In the case of unknown friction force, the controller is able to reject the disturbance. However, the estimation of the friction force by means of the UIO allow us to implement a compensation mechanism that makes the controller to reject the disturbance faster than in the case such estimation is not available. Figure 6.3(b) depicts how the angular velocity performance is higher than the linear velocity one with respect to the reference. In addition, it can be appreciated how the compensation mechanism corrects also faster than in the case of unknown disturbance. Even so, the angular response presents some overshoot behaviour at some time instants. The controller adjustment may be one of the reasons, but the main reason is the high abruptness of the angular velocity reference at the end of the curves producing a rough behaviour on the vehicle.

Figure 6.4 presents position errors for both cases. It can be seen the better performance when the disturbance is compensated. The mitigation of these errors is crucial for achieving a good autonomous guidance. However, a near zero lateral error is more important since it ensures the driving of the vehicle through the center of the road. In our results, longitudinal error is no longer than 0.5 m in normal driving (i.e. neither accelerating nor braking). Lateral error remains in the scale of few decimeters being increased when both velocities (angular and linear) increase. In addition to this graphical comparison, a quantitative one in terms of the RMSE has been performed (see Table 6.6). Such a results verify the improvement of using the friction force compensation mechanism.

Figure 6.5 shows the resulting control actions. The compensation mechanism allows to reduce the control effort being the action also smoother than in the unknown friction case. Note also that the steering angle signal in the first part of the simulation is quite abrupt. This behaviour is due to longitudinal and angular behaviours are highly coupled and the starting stage deals with high linear accelerations.

Table 6.6: Comparison of both approaches using a quadratic measure and the maximum error values in meters

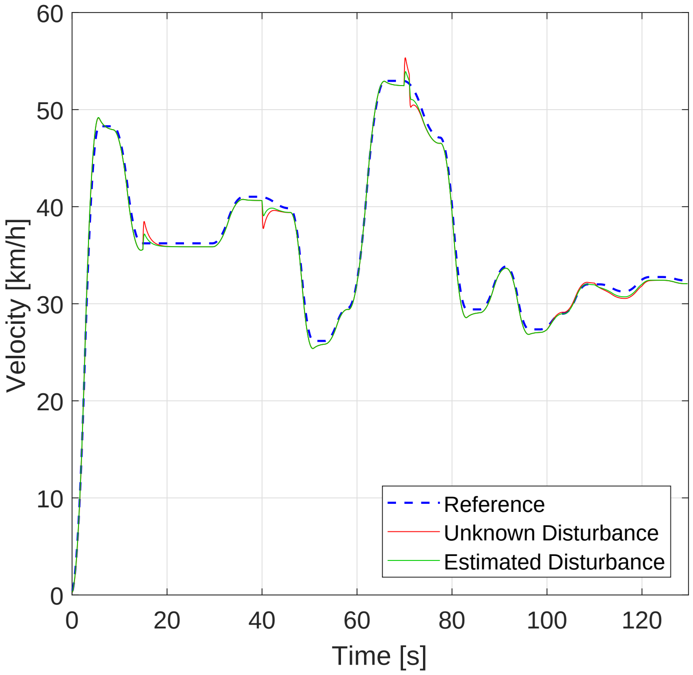
Approach	RMSE$_v$	RMSE$_\omega$	RMSE$_y$	$x_{e_{max}}$	$y_{e_{max}}$
unknown	0.3834	0.0043	0.0204	1.2815	1.1055
estimated	0.2337	0.0041	0.0201	0.082	0.067

Figure 6.6(a) shows the real friction force considered along the circuit simulated and the estimated force.

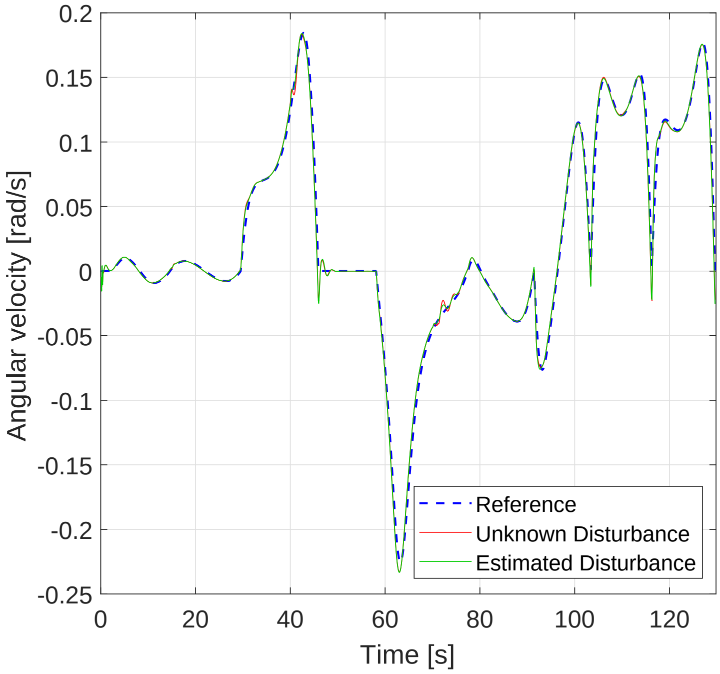
Figure 6.6(b) illustrates the closed poles for the kinematic and dynamic loops at a given operating point. It can be observed that the poles of both loops satisfy the constraints imposed by the corresponding decay rates η and β (see (6.21) and (6.27)). The satisfaction of this condition allows to design both loops separately, since the dynamic control presents a faster dynamic behaviour than the kinematic one.

6.6 Conclusions

A gain-scheduling LPV-LQR control scheme has been introduced for solving the mixed control problem. To this aim, two models, i.e. kinematic and dynamic, have been expressed in the polytopic LPV form and an approach based on cascade design of the kinematic and dynamic controllers has been adopted with the aim of increasing the performance of the system. This is achieved by forcing the inner closed loop dynamics to behave faster than the outer closed-loop one. Moreover, a novel LPV-UIO design following the control-observer duality has been presented. This dynamic estimator solves the problem of the lack of measurability in the case of the slip angle by estimating all the dynamic states as well as the friction force affecting the vehicle. Then, a friction force compensation mechanism is presented allowing the vehicle to compensate faster

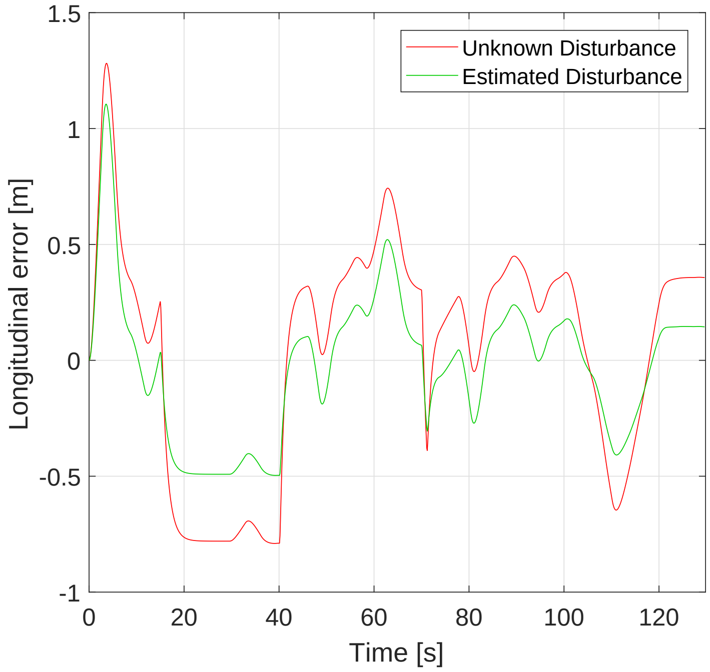


(a)

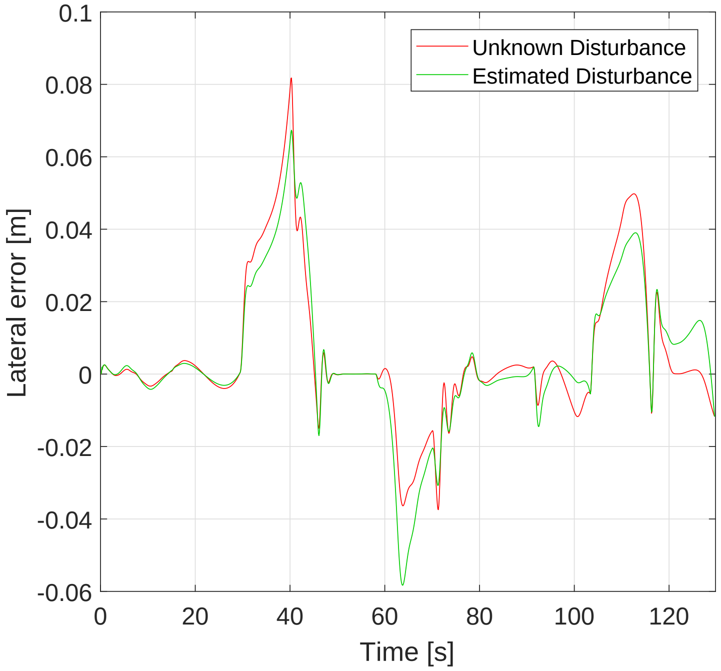


(b)

FIGURE 6.3: Velocities of the vehicle: a) Linear velocity reference and response. b) Desired and simulated angular velocities

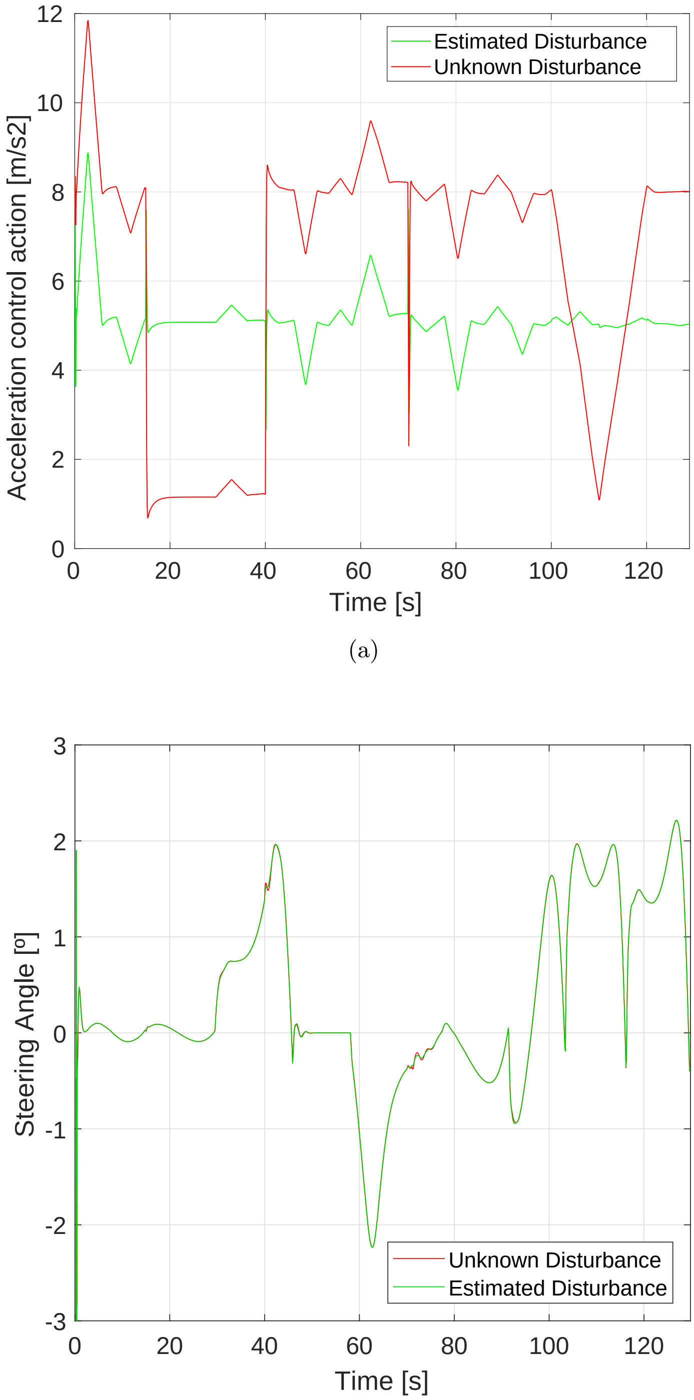


(a)



(b)

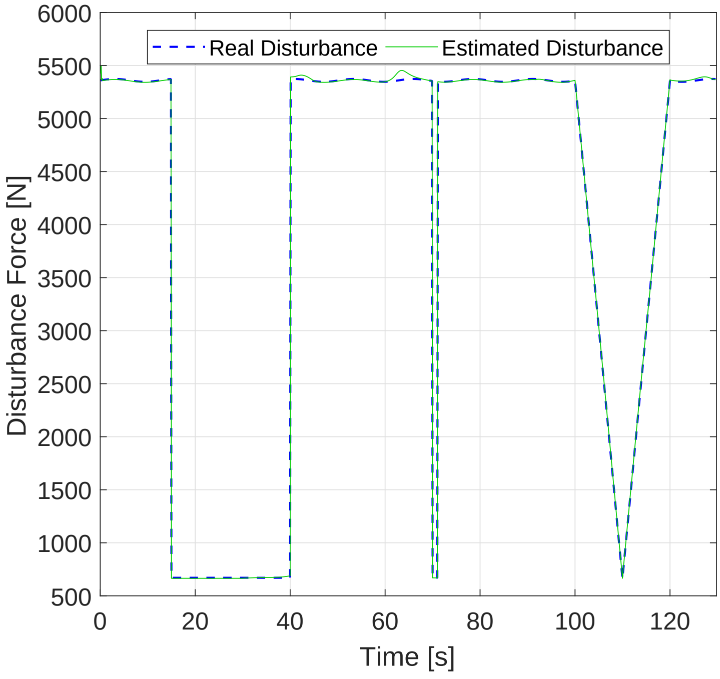
FIGURE 6.4: Resulting position errors: a) Vehicle longitudinal error along the circuit, b) Vehicle lateral error



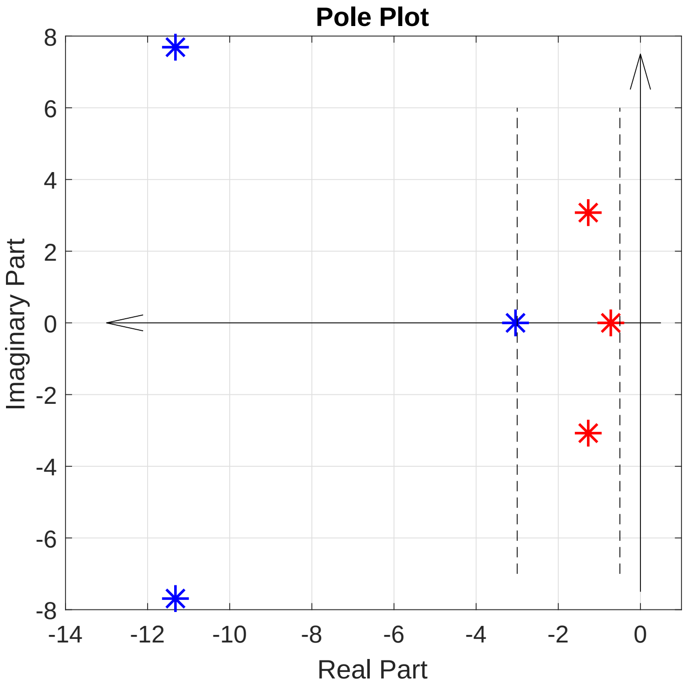
(a)

(b)

FIGURE 6.5: Resulting control actions: (a) Rear wheels acceleration vector, (b) Steering angle at the front wheels



(a)



(b)

FIGURE 6.6: Disturbance profile and system poles location: a) Real and estimated disturbances. b) Pole locus of the system in a particular operating point ($v = 8.33\frac{m}{s}$, $\omega = 0.05\frac{rad}{s}$ and $\alpha = 0.013rad$). Blue marks are the three slower poles of the dynamic loop and the red ones are the kinematic poles. Vertical dashed lines represent the hyper-planes $\eta = 3$ and $\beta = 0.1$

the disturbances caused by changes in the friction force as well as reducing the control effort.

Two novel LMI-based optimal designs for LPV observer and controller have been introduced. They also present new integrated constraints for ensuring a certain level of control and observation performance.

The obtained gain scheduling LPV-LQR control approach, jointly with the UIO and a trajectory planning module, has presented suitable results in a simulated scenario. In the same way, a comparison is shown about the friction force estimation, which shows the usefulness of this approach.

Chapter 7

Autonomous racing using TS-MPC including a TS-MHE-UIO estimator

The content of this chapter is based on the following work:

- [Alcalá, Eugenio, Puig, Vicenç and Quevedo, Joseba., 2019.A].
  Alcalá, E., Cayuela, V. P., & Casin, J. Q. (2019). TS-MPC for Autonomous Vehicles including a TS-MHE-UIO estimator. *IEEE Transactions on Vehicular Technology.*

7.1 Introduction

This chapter presents the concept of autonomous racing and, consequently, a control scheme capable of solving the problem of autonomous driving in such a scenario. As presented in the state of the art (Section 2), there is a few number of approaches that address the problem of control for racing vehicles. However, some peculiarities such as the use of an uncoupled control structure motivate the research for another complementary solution. This motivation lies in the ability to solve two optimization problems (kinematic control and dynamic observer) at a low computational cost using TS-based models and allowing real-time calculation on embedded platforms.

The contribution of this chapter is two-fold and focuses on the use of TS polytopic models for the design of the control and observation layers. First, the MPC technique is designed using a TS kinematic representation that leads to a quadratic problem. In addition, the introduction of the terminal set concept allows guaranteeing system asymptotic stability.

Second, the Moving Horizon Estimator (MHE) strategy is merged with the use of a dynamic vehicle model formulated as TS as well as with the UIO concept, thus allowing the estimation of states and disturbances through a very fast predictive optimization (TS-MHE-UIO).

7.2 Overview of the proposed solution

We consider the problem of autonomous guidance of a vehicle in a racing scenario. To do so, two important tasks have to be carried out: the trajectory planning and the automatic control.

On one hand, the planning of the trajectory to be followed by the vehicle has to fulfill a set of racing specifications as well as certain characteristics such as continuous and differentiable velocity profiles. Thus, this module is in charge of providing discrete and smooth racing references to the automatic control stage. This racing-oriented trajectory planner will be presented in Chapter 10. On the other hand, the automatic control is in charge of following the planned references, thus, moving the vehicle between two ground coordinates as well as generating smooth control actions for achieving a comfortable journey. In Figure 7.1, we show the planning-control-estimation diagram proposed in this work. Observe that two control levels have been designed, one for the position control and the other one to control the dynamic behaviour of the vehicle, i.e. linear and angular velocities. In addition, the lack of measurement of certain vehicle states as well

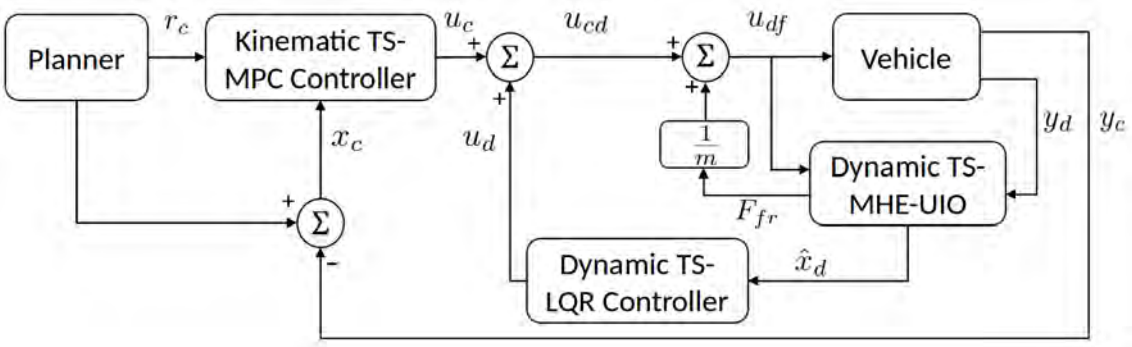


FIGURE 7.1: Autonomous guidance diagram with kinematic and dynamic control layers and a dynamic state estimator with friction force compensator

as the lack of knowledge of external disturbances can generate a problem when applying the designed control. Thus, a dynamic estimator is introduced to solve this problem (see Figure 7.1).

The level of difficulty of the control problem applied to vehicle guidance when using model-based techniques is often determined by two aspects: the behaviour to be controlled (lateral, longitudinal or mixed) and the complexity of the model used for control design (several options from the simplest kinematic representation to the most elaborated dynamic, through its linear or non-linear formulation). In the following, one of the most complex configuration, the mixed non-linear dynamic problem, is covered.

7.3 TS control-oriented modeling

The use of a cascade control scheme implies two model-based control techniques that consequently, implies two dynamics representation. Hence, two control layers are studied in this chapter. In the outer layer, the kinematic behaviour is governed (see Section 3.2). This is based on the velocity vector movement in order to compute longitudinal and lateral velocities referenced to a global inertial frame. On the other hand, in the inner layer the dynamic behaviour is controlled (see Section 3.3), where the motion is generated by applying an acceleration vector on the driven wheels and mass, inertial and tire parameters are considered.

Both kinematic and dynamic TS models employed in this chapter have been presented in Chapter 4. With regard to the kinematic model, the reference-based formulation (4.2) has been adopted for this control scheme using the polytopic formulation

$$\dot{x}_c = A_c(\zeta_c)x_c + B_c u_c - B_c r_c \ , \tag{7.1}$$

where its vector of scheduling variables is defined as: $\zeta_c := [\omega, v_d, \theta_e]$ and the scheduling variables are bounded in

$$\omega \in [-3, 3] \ \frac{rad}{s} \quad , \quad v_d \in [0.1, 3.5] \ \frac{m}{s} \ ,$$

$$\theta_e \in [-0.15, 0.15] \ rad \ .$$

Note that these interval limits define the polytopic region (Θ) of the TS kinematic model. Regarding the dynamic model, the continuous Cartesian frame (4.9) has been chosen given by

$$\dot{x}_d = A_d(\zeta_d)x_d + B_d u_d + E_d F_{fr} \ , \tag{7.2}$$

where its vector of scheduling variables is defined as: $\zeta_d := [\delta, v_x, v_y]$ and the scheduling variables are bounded in

$$\delta \in [-0.35, 0.35] \ rad \quad , \quad v_x \in [0.1, 3.5] \ \frac{m}{s} \ ,$$

$$v_y \in [-2, 2] \ \frac{m}{s} \ .$$

In the same way than in previous chapter, an unknown friction force disturbance acting on the vehicle F_{fr} has been considered and its input matrix E_d already presented in Chapter 6. Notice that c and d subindexes denote kinematic and dynamic belonging, respectively.

7.4 Control Design

In this section, we present the control scheme proposed in this chapter as well as its design. The control strategy of the vehicle has been divided into two nested layers, see Figure 7.1. The outermost layer controls the vehicle's kinematics, i.e. its position and orientation, and works at a frequency of 20 Hz. On the other hand, the internal loop controls the dynamic behavior of the vehicle, i.e. its speeds, at a frequency of 200 Hz. In the following, both control loops are described separately.

7.4.1 Kinematic TS-MPC Design

At this point, we present the formulation of the TS-MPC strategy, which focuses on solving position and orientation control of the vehicle. This strategy is based on the resolution of a QP optimization problem by using the non-linear reference.based kinematic model in its TS polytopic representation. However, there exist the problem associated with the lack of knowledge of the matrix of scheduling variables through the entire prediction horizon. In Cisneros, Pablo SG, Sophia Voss, and Herbert Werner. [2016], the use of the optimized state sequence which is obtained after each optimization is proposed.

In this work, the scheduling variables are states of the system whose desired values are known since the trajectory planner generates them. That is why we propose the use of such references as known scheduling variables for the entire optimization horizon (H_p) allowing to compute the sequence of varying parameters $\zeta := [\zeta_1, ..., \zeta_{H_p}]$. In this way, we can calculate the evolution of the model more accurately and in anticipation. In addition, since the basic MPC formulation cannot guarantee the overall stability of the system, we propose the addition of a terminal constraint and a terminal cost to the optimization problem.

To formulate the problem, the continuous-time polytopic TS system presented in (4.2) has been considered. The system discretization is made online by using Euler approach with a sampling time (T_s) of 50 ms. Notice that in order to avoid a difficult reading, the sub-index c is omitted in the rest of the subsection. Then, the focus is on a MPC scheme where the cost function is defined as

$$\mathrm{J_k} = \sum_{i=0}^{H_p-1} \left(x_{k+i}^T Q x_{k+i} + \Delta u_{k+i} R \Delta u_{k+i} \right) + x_{k+H_p}^T P x_{k+H_p} \ , \tag{7.3}$$

where $Q = Q^T \in \mathbb{R}^{s\times s} \geq 0$, $R = R^T \in \mathbb{R}^{n\times n} > 0$ and $P = P^T \in \mathbb{R}^{s\times s} > 0$ represent the states, inputs and terminal states weights, respectively. At each discrete time k the values of x_k and u_{k-1} are known and the following optimization problem can be solved

$$\begin{aligned}
&\underset{\Delta U_k}{\text{minimize}} \quad \mathrm{J_k}(\Delta U_k, X_k) \\
&\text{subject to} \\
&\qquad x_{k+i+1} = x_{k+i} + \left(\sum_{j=1}^{N} \mu_j(\zeta_{k+i}) A_j x_{k+i} + B u_{k+i} - B r_{k+i} \right) T_s \\
&\qquad u_{k+i} = u_{k+i-1} + \Delta u_{k+i} \ , \quad \forall i = 0, ..., H_p - 1 \\
&\qquad \Delta U_k \in \Delta\Pi \\
&\qquad U_k \in \Pi \\
&\qquad x_{k+H_p} \in \chi_f \ ,
\end{aligned} \tag{7.4}$$

where

$$\Delta U_k = \begin{bmatrix} \Delta u_k \\ \Delta u_{k+1} \\ \vdots \\ \Delta u_{k+H_p-1} \end{bmatrix} \in \mathbb{R}^{H_p \times n} \ , U_k = \begin{bmatrix} u_k \\ u_{k+1} \\ \vdots \\ u_{k+H_p-1} \end{bmatrix} \in \mathbb{R}^{H_p \times n} \ , \tag{7.5}$$

being n the number of inputs of the kinematic system. Π and $\Delta\Pi$ are the constraint sets for the inputs and their derivatives, respectively. Note that system matrices A_j and B are in discrete time. The set χ_f represents the terminal state domain. Then, by introducing both P and χ_f we force the states to converge into a stable region and then, to ensure the MPC stability. The computation of these terminal cost and set are computed by solving two offline LMI-based problems.

First, the controller for each polytopic system (A_i) is found by solving the following LQR-LMI for discrete-time systems

$$\begin{bmatrix} Y & (A_iY + BW_i)^T & Y & W_i^T \\ A_iY + BW_i & Y & 0 & 0 \\ Y & 0 & Q_{TS}^{-1} & 0 \\ W_i & 0 & 0 & R_{TS}^{-1} \end{bmatrix} \geq 0 \tag{7.6}$$
$$\forall i = 1, ..., N \ ,$$

with $Y = Y^T \in \mathbb{R}^{s \times s} > 0$, $Q_{TS} = Q_{TS}^T \in \mathbb{R}^{s \times s} \geq 0$, $R_{TS} = R_{TS}^T \in \mathbb{R}^{n \times n} > 0$. This problem returns the matrices Y and W_i. Then, the resulting controllers are obtained by $K_i = W_iY^{-1} \in \mathbb{R}^{n \times s}$. Note that the terminal cost matrix P in (7.3) is found to be equal to Y^{-1}. This LQR design is a particular formulation for the one presented in Theorem 25 of Tanaka, Kazuo, and Hua O. Wang. [2004].

The second problem consists on finding the largest terminal domain χ_f as a positive invariant set. To accomplish this, the following constrained optimization problem using the previously obtained controllers K_i is solved

$$\begin{aligned} \underset{Z}{\text{maximize}} \quad & det(Z) \\ \text{subject to} \quad & \\ & \begin{bmatrix} -Z & Z(A_i + BK_i)^T \\ (A_i + BK_i)Z & -Z \end{bmatrix} < 0 \\ & K_i Z K_i^T - \overline{u}^2 < 0 \quad , \quad \forall i = 1, ..., N \ . \end{aligned} \tag{7.7}$$

The resulting variable is $Z \in \mathbb{R}^{s \times s}$. Hence, we compute the largest terminal set as the ellipsoid $\chi_f = \{x | x^T S x \leq 1\}$, with $S = Z^{-1}$. Note that this problem is totally constrained by the maximum values of the control actions.

7.4.2 Dynamic TS-LQR Design

The gain-scheduling technique is used for controlling the dynamic behaviour of the vehicle in the inner loop. As made with the reference-based kinematic model, the polytopic dynamic model (7.2) is also discretized using Euler approach but using a T_s of 5 ms. Then, we solve offline the discrete-time LQR-LMI problem (7.6) for computing the polytope vertex controllers K_i. Finally, at every dynamic time step T_s, the dynamic controller gain is computed online using the interpolation rule presented in (6.15) - (6.16)

$$K(\zeta) = \sum_{i=1}^{N} \mu_i(\zeta) K_i \ . \tag{7.8}$$

The offline computation of polytopic controllers allows this control strategy to work at the desired frequency of 200 Hz.

7.5 TS-MHE-UIO Design

On one hand, the aim of the MHE is to predict the dynamic states for the next iteration by means of running a constrained optimization and using a set of past allowed measurements. Currently, it is usual to have experimental vehicles with different kind of sensors mounted on it that allows the measurement of almost every dynamic variable of the vehicle. However, most of the cheap versions of sensors are noisy and need of a post processing. Then, it is at this point where the state estimator gains interest.

The UIO approach deals with the estimation of external disturbances. One of the most relevant disturbances in road vehicles is the continuous change of road surface. This is why the coefficient of friction varies producing a remarkable alteration in the total computation of acting forces, drastically affecting the behavior of the vehicle.

In this section, we present a novel approach combining both the MHE and the UIO, to converge to an optimal state estimator able to predict disturbances. In addition, using a TS model formulation for computing the evolution during the established horizon allows the algorithm to run faster than non-linear model-based MHE and guarantee asymptotic convergence. To avoid a difficult reading, the sub-index d is omitted in system vectors but not in systems matrices.

7.5.1 UIO

The UIO goal is to estimate the main disturbances acting over the vehicle. Such a procedure is based on calculating the difference between the observation model and the real system [Keller, Jean-Yves, and Mohamed Darouach., 1999]. In this chapter, we have considered as disturbance the friction force acting on the longitudinal vehicle axis

$$F_{fr_k} = \Gamma\left(y_k - C\Big(\hat{x}_k + \Big(\sum_{i=1}^{N} \mu_i(\zeta) A_{d_i} \hat{x}_k + B_d u_k \Big) T_s \Big) \right), \tag{7.9}$$

with

$$C = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \hat{x} = \begin{bmatrix} \hat{v_x} \\ \hat{v_y} \\ \hat{\omega} \end{bmatrix}, \ \Gamma = (CE_d)^{+} \ , \tag{7.10}$$

where A_{d_i}, B_d and E_d are the system matrices in (7.2). Function $(\cdot)^{+}$ denotes the pseudoinverse function. Then, at every control iteration and once the state estimation has been solved, F_{fr} is computed.

7.5.2 TS-MHE Design

In order to design the MHE-UIO, the polytopic TS system (7.2) is used. In this chapter, it is considered that all the dynamic states are measured being then the filtering task the main work of the MHE.

The MHE optimization problem is based on minimizing the following cost function

$$\begin{aligned} \mathrm{J_k} &= (\hat{x}_{k-H_p} - x_o)^T P_o (\hat{x}_{k-H_p} - x_o) \\ &+ \sum_{i=-H_p}^{0} \big(w_{k+i}^T Q w_{k+i} + s_{k+i}^T R s_{k+i} \big), \end{aligned} \tag{7.11}$$

where s_{k+i} represents the error between the measured and estimated variables, and w_{k+i} is the state estimation error.

Therefore, at each discrete time step k, knowing the vectors

$$U_k = \begin{bmatrix} u_{k-H_p} \\ u_{k-H_p+1} \\ \vdots \\ u_k \end{bmatrix} \in \mathbb{R}^{H_p \times n} \ , \ Y_k = \begin{bmatrix} y_{k-H_p} \\ y_{k-H_p+1} \\ \vdots \\ y_k \end{bmatrix} \in \mathbb{R}^{H_p \times p} \ , \tag{7.12}$$

and the initial state x_o, the constrained optimization problem

$$\begin{aligned} \underset{\hat{X}_k}{\text{minimize}} \quad & \mathrm{J_k}(\hat{X}_k) \\ \text{subject to} \quad & \hat{x}_{k+i+1} = \hat{x}_{k+i} + \Big(\sum_{j=1}^{N} \mu_j(\zeta_d) \big(A_{o_j} \hat{x}_{k+i} + B_{o_j} u_{k+i} \big) \Big) T_s \\ & + w_{k+i} + E_d \Gamma y_{k+i} \quad i = -H_p, ..., 0 \\ & y_{k+i} = C \hat{x}_{k+i} + s_{k+i} \quad i = -H_p, ..., 0 \\ & \hat{X}_k \in X_d \ , \end{aligned} \tag{7.13}$$

is solved online for

$$\hat{X}_k = \begin{bmatrix} \hat{x}_{k-H_p+1} \\ \hat{x}_{k-H_p+2} \\ \vdots \\ \hat{x}_{k+1} \end{bmatrix} \in \mathbb{R}^{H_p \times s} \ , \tag{7.14}$$

where X_d is the constraint region for the dynamic states, $Q = Q^T \in \mathbb{R}^{s\times s} \geq 0$, $R = R^T \in \mathbb{R}^{s\times s} > 0$, $P_o = P_o^T \in \mathbb{R}^{s\times s} > 0$ and

$$A_{o_j} = (I - E_d \Gamma C) A_{d_j}$$

$$B_{o_j} = (I - E_d \Gamma C) B_{d_j} \ ,$$

are the unknown input matrices [Keller, Jean-Yves, and Mohamed Darouach., 1999]. Notice that the time discretization of the continuous model is made online at a T_s of 5 ms.

7.6 Simulation Result

In this section, we validate the performance of the proposed control-observer scheme in a racing scenario through simulation in MATLAB. The considered vehicle for running simulations is a 1/10 scale RWD electric vehicle whose dynamics are described by the following differential equation

$$\begin{aligned}
\dot{x} &= v_x \cos\theta - v_y \sin\theta \\
\dot{y} &= v_x \sin\theta + v_y \cos\theta \\
\dot{\theta} &= \omega \\
\dot{v}_x &= a_r + \frac{-F_{yf}\sin\delta - F_{df}}{m} + \omega v_y \\
\dot{v}_y &= \frac{F_{yf}\cos\delta + F_{yr}}{m} - \omega v_x \\
\dot{\omega} &= \frac{F_{yf} l_f \cos\delta - F_{yr} l_r}{I} \\
\alpha_f &= \delta - \tan^{-1}\left(\frac{v_y}{v_x} - \frac{l_f \omega}{v_x}\right) \\
\alpha_r &= -\tan^{-1}\left(\frac{v_y}{v_x} + \frac{l_r \omega}{v_x}\right) \\
F_{yf} &= d \sin\left(c \tan^{-1}(b\alpha_f)\right) \\
F_{yr} &= d \sin\left(c \tan^{-1}(b\alpha_r)\right) \\
F_{df} &= \mu m g \ ,
\end{aligned} \tag{7.15}$$

where the parameters b, c and d define the shape of the semi-empirical Magic Formula curve.

The geometric and dynamic parameters used to parameterize the control-oriented and simulation models are shown in Table 7.1.

Table 7.1: Dynamic model parameters

Parameter	Value	Parameter	Value
l_f	$0.125\ m$	l_r	$0.125\ m$
m	$1.98\ kg$	I	$0.03\ kg\ m^2$
C_f	$68\ \frac{N}{rad}$	C_r	$71\ \frac{N}{rad}$
d	8.255	c	1.6
b	6.1	μ	0.85

To show the effectiveness of the estimation scheme, we perform the comparison of adding the online estimated friction force to the TS-MPC control action (compensated in figures) against not estimating the friction force (no compensated in figures). As it is shown in Figure 7.1, the current estimated friction force (F_{fr}) is converted into acceleration to be properly added to the control variable (a). In addition, we show the promising results of the TS-MPC approach by performing a comparison against the non-linear MPC approach (NL-MPC in resulting figures).

The TS-MPC uses planning data to instantiate the state space matrices at every time step within the MPC prediction stage. Such references from the planner are obtained by using racing-oriented trajectory planner that will be presented in Section 10.2. To verify the real-time feasibility of the presented strategies, we perform the simulations on a DELL inspiron 15 (Intel core i7-8550U CPU @ 1.80GHzx8).

Then, the optimal control problem (8.2) is solved at a frequency of 20 Hz using the solver GUROBI [Optimization, Gurobi., 2014] through YALMIP [Lofberg, J., 2004] framework. For the non-linear MPC case, the solver IPOPT is used. This solves the position control problem in a outer loop (see Figure 7.1). In the inner loop, the dynamic state feedback control problem (Section 7.4.2) is solved at a rate of 200 Hz to control the velocities of the vehicle. The dynamic states used by this inner control law are provided by solving the optimal problem (7.13).

The vehicle model, TS-MPC, TS-MHE-UIO and dynamic TS-LQR parameters are listed in Tables 7.1, 7.2, 7.3 and 7.4, respectively.

Table 7.2: Kinematic TS-MPC design parameters

Parameter	Value	Parameter	Value
Q	0.99*diag(0.66 0.01 0.33)	$\overline{u}$	[3.5 3]
R	0.01*diag(0.5 0.5)	$\underline{u}$	[0.1 -3]
T_s	50 ms	$\overline{\Delta u}$	[0.3 0.3]
H_p	10	$\underline{\Delta u}$	[-0.3 -0.3]
R_{TS}	diag(1 3)	Q_{TS}	diag(1 1.5 3)

Table 7.3: TS-MHE-UIO design parameters

Parameter	Value	Parameter	Value
Q	0.99*diag(0.33 0.33 0.33)	H_p	15
R	0.01*diag(0.5 0.5)	T_s	5 ms
P_o	diag(2 2 2)	$\overline{\hat{x}}$	[3.5 2 3]
$\underline{\hat{x}}$	[0.1 -2 -3]		

Table 7.4: Dynamic TS-LQR design parameters

Parameter	Value
Q	0.99*diag(0.8 0.01 0.19)
R	0.01*diag(0.5 0.5)
T_s	5 ms

Solving the problem (7.7) to determine the largest terminal set, we obtain matrix S as

$$S = \begin{bmatrix} 0.465 & 0 & 0 \\ 0 & 23.813 & 76.596 \\ 0 & 76.596 & 257.251 \end{bmatrix}. \tag{7.16}$$

The comparison is made in the circuit presented in Figure 7.2.

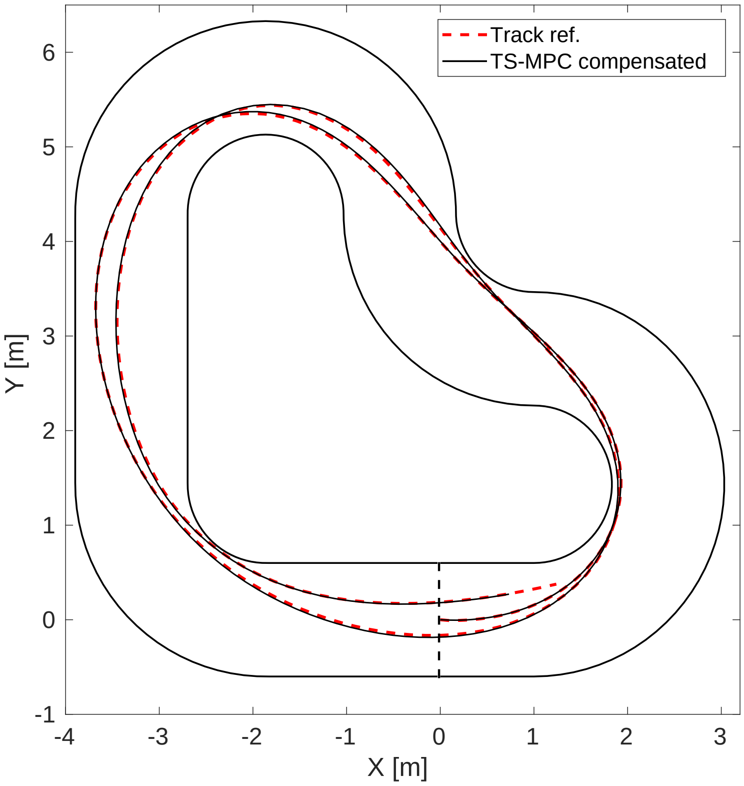


FIGURE 7.2: Simulation circuit used for testing the proposed control technique

It is intended to show a racing situation in which the vehicle goes as fast as possible. To this aim, a reference obtained in an offline way is provided, which optimizes the trajectory by minimizing the lap time. In this way the automatic control of the vehicle becomes a greater challenge having to manage the behavior of the car very close to the dynamic limits of this. In addition, the vehicle performs under the influence of friction force disturbance. Figure 7.3) illustrates such a disturbance profile and its estimation using the TS-MHE-UIO throughout the simulation test.

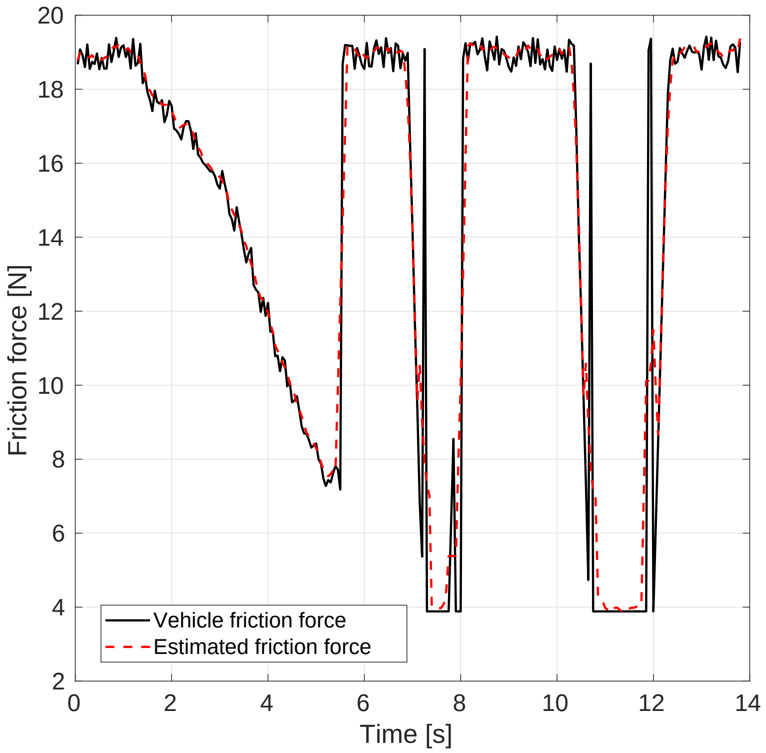


Figure 7.3: Static friction force disturbance and its estimation

It can be observed how the observer is successfully able to estimate the changing disturbance. Below, in Figure 7.4, we show both, the linear and angular speed profiles provided by the trajectory planning and the respective vehicle responses for every compared case.

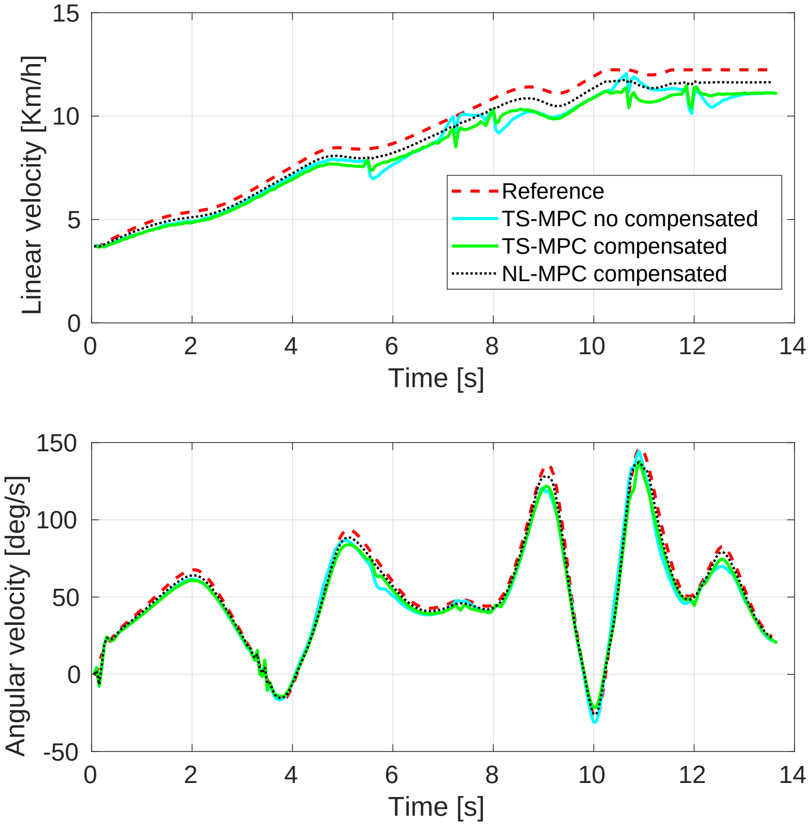


Figure 7.4: Reference and response velocities for the three cases compared

It can be appreciated a little better response in the linear velocity tracking by the NL-MPC approach, but also, by the TS-MPC algorithm with force compensation with respect to the non compensation scenario. In Figure 7.5, we illustrate the complete set of errors, i.e. position, orientation and velocities errors. It is seen the close behaviour between TS-MPC compensated and NL-MPC compensated.

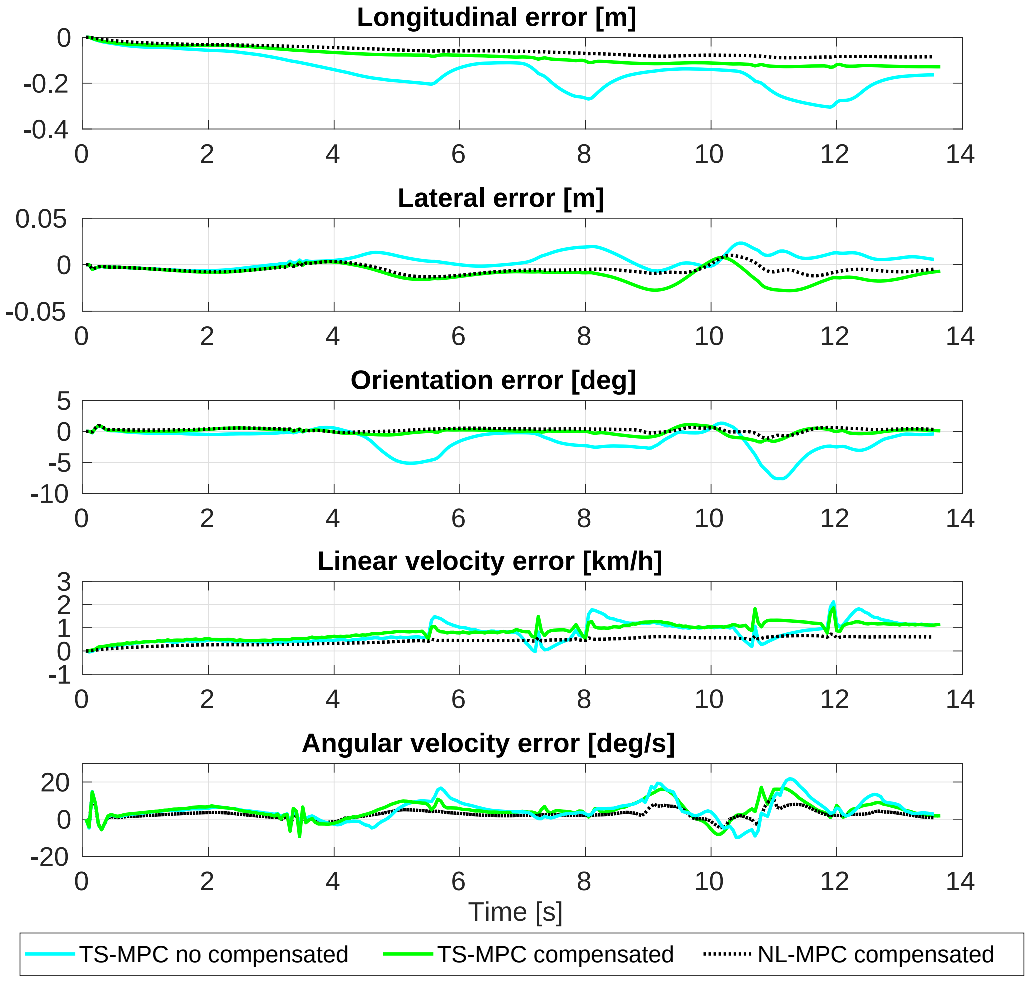


Figure 7.5: Time evolution of the tracking errors for each compared kinematic control strategy

Moreover, it may be appreciated the effectiveness of the compensation mechanism helping the controller to handle the coming external disturbances, and hence, helping to reduce the tracking errors. The respective control actions applied to the simulation vehicle are shown in Figure 7.6.

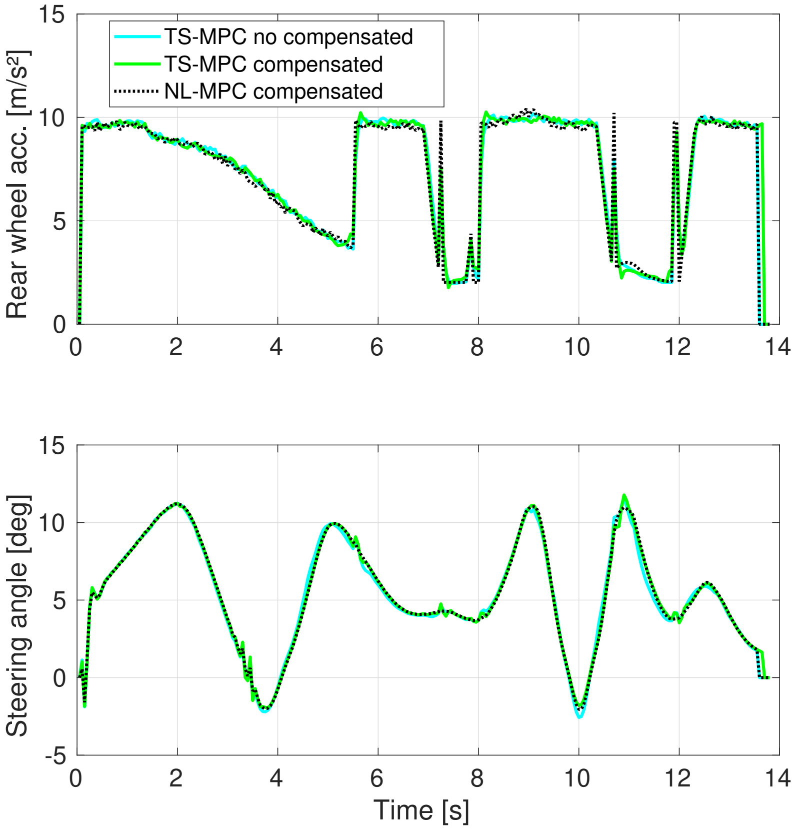


FIGURE 7.6: Resulting control actions. Up: Linear acceleration applied over the rear wheel axle. Down: Steering angle applied on the front wheel

It is observed a similar response throughout the test even in the no compensated case. However, little differences in the actuation variables at high speeds may make the states response to be different.

An interesting aspect of racing behaviour as well as a good difficulty meter of the performance carried out is the slip angle on the wheels. The difference between the front slip angle and the rear slip angle gives us information about whether the vehicle enters the understeer or oversteer situation.

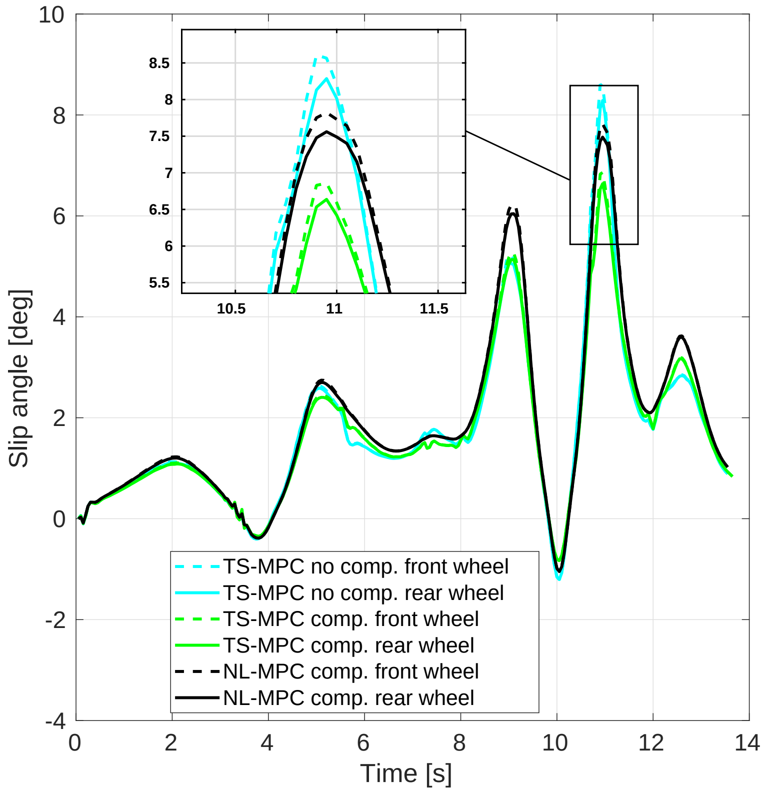


Figure 7.7: Front and rear slip angles for the three compared cases

In Figure 7.7, it can be seen a set of large slip angles at the ending part of the simulation as well as some difference between the front and rear slip angles (see Figure 7.7 zoom). The frontal slip angle situation greater than the rear slip angle is known as an understeer and is a behavior to avoid. In this case, we can appreciate this behavior due to a very fast and extreme driving in certain curves.

From a real time feasibility point of view, an aspect to highlight when dealing with control strategies based on optimization is the computational time spent at each optimization procedure. In Figure 7.8, we show the elapsed time at each kinematic MPC optimization for the TS-MPC and NL-MPC approaches. It is shown the computational time improvement when using the TS-MPC strategy. Note that the non-linear optimization problem has been solved using IPOPT solver.

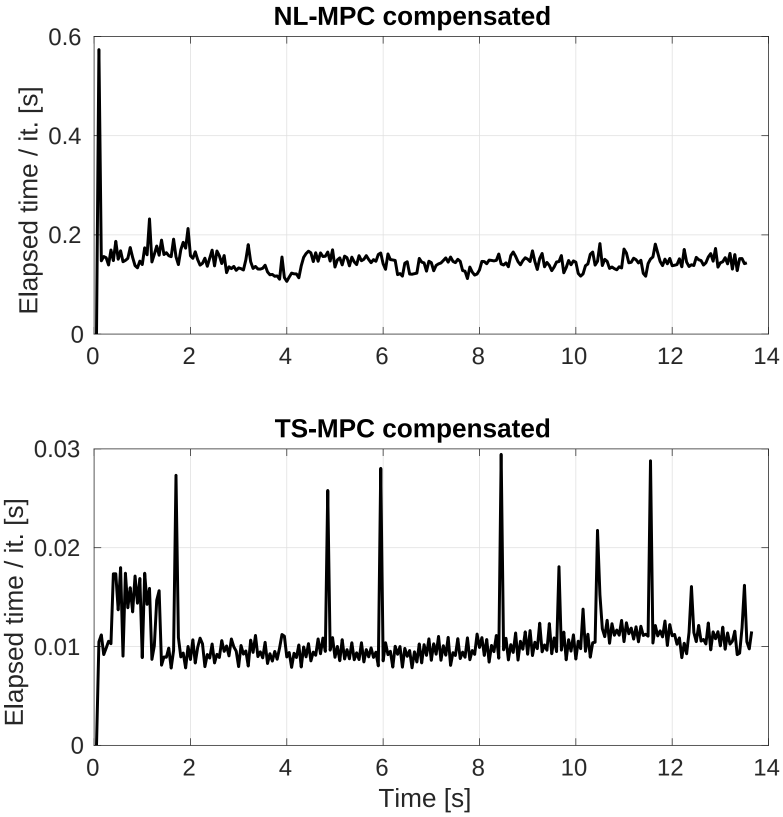


Figure 7.8: Computational time when solving the kinematic MPC for both of the compensated cases

Finally, a quantitative comparison is made using the RMSE criterion as performance measurement. This is shown in Table 7.5.

Table 7.5: Comparison using a quadratic measure

Approach	RMSE$_x$	RMSE$_y$	RMSE$_\theta$	RMSE$_v$	RMSE$_w$
TS-MPC no compensated	0.167	0.015	0.042	0.240	0.131
TS-MPC compensated	0.091	0.013	0.009	0.241	0.116
NL-MPC compensated	0.063	0.008	0.007	0.129	0.063

7.7 Conclusions

In this chapter, a cascade control scheme (kinematic and dynamic) has been presented to solve the problem of integrated control tracking (lateral and longitudinal) for autonomous vehicles in racing situations.

The novel kinematic control has been designed using the MPC technique with the prediction model expressed as a TS model using the non-linear embedding approach. Furthermore, the proper analisys has been made to ensure feasibility and stability by means of Lyapunov and invariant set theory. On the other hand, the discrete-time LQR-LMI formulation using the TS-based modeling of the vehicle has been used to address the dynamic control design.

A comparison has been made between two methods of solving the control problem: using the NL-MPC and using the TS-MPC approach being the TS model instantiated at each prediction step within the prediction stage using planning data. It has been demonstrated that the TS-MPC technique presents a close performance to the non-linear control problem but in a much faster way (between 10 and 20 times).

In addition, the TS-MHE-UIO has been introduced with the aim of estimating dynamic states and disturbances acting on the vehicle, such as the friction force. The estimation of the friction force has been used to compensate the disturbance and allow lower control efforts. It has also been demonstrated the effectiveness of this mechanism in the comparison performed.

Chapter 8

Autonomous racing using LPV-MPC

The content of this chapter is based on the following works:

- [Alcalá, Eugenio, et al., 2020.A]. Alcalá, E., Puig, V., Quevedo, J. & Rosolia, U. Autonomous Racing using Linear Parameter Varying - Model Predictive Control (LPV-MPC). *Control Engineering Practice*, 95, 104270

8.1 Introduction

In this chapter, the problem of designing a more advanced control technique to solve the autonomous racing problem is investigated. In particular, this chapter takes advantage of the properties of polytopic LPV systems and predictive optimal control to solve a more challenging driving problem.

The LPV-MPC approach as a novel option to solve driving control problems has been proposed. In Jungers, Marc, Ricardo CLF Oliveira, and Pedro LD Peres. [2011], Xu, Zuhua, et al. [2009], Besselmann, Thomas, and Manfred Morari. [2009], Besselmann, Thomas, Johan Lofberg, and Manfred Morari. [2012], different ways of dealing with the MPC strategy using an LPV representation are addressed.

To address the racing behaviour, a trajectory planning is computed offline to generate the racing-based references taking into account the vehicle dynamics and adjusted to the

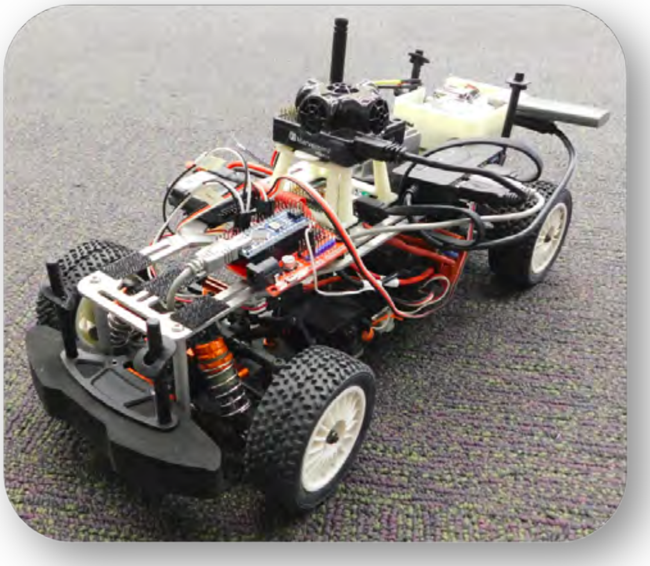


FIGURE 8.1: Vehicle used for experimental tests (BARC)

testing circuit. This racing-oriented trajectory planner will be presented in Section 10.2. The main contribution of this chapter is to merge the LPV paradigm with the MPC technique to compute the online LPV-MPC strategy for tracking the racing trajectory which enables real-time embedded system computation.

8.2 Testing vehicle

The Berkeley Autonomous Race Car [Gonzales, J., et al., 2016] (BARC[1]) is a development platform for autonomous driving to achieve complex maneuvers. This is a 1/10 scale RWD electric remote control (RC) vehicle (see Figure 8.1) that has been modified to operate autonomously. Mechanically speaking, this has been modified with some decks to protect the on-board electronics and sensors. This vehicle includes a basic net of sensors for performing localization. A fusion of IMU, encoders and indoor GPS data is made using a Kalman filter to achieve an accurate localization while testing. An Odroid XU4 is used to run ROS framework and the control and planning algorithms. For more details about ROS see O'Kane, Jason M. [2014].

[1] http://www.barc-project.com/

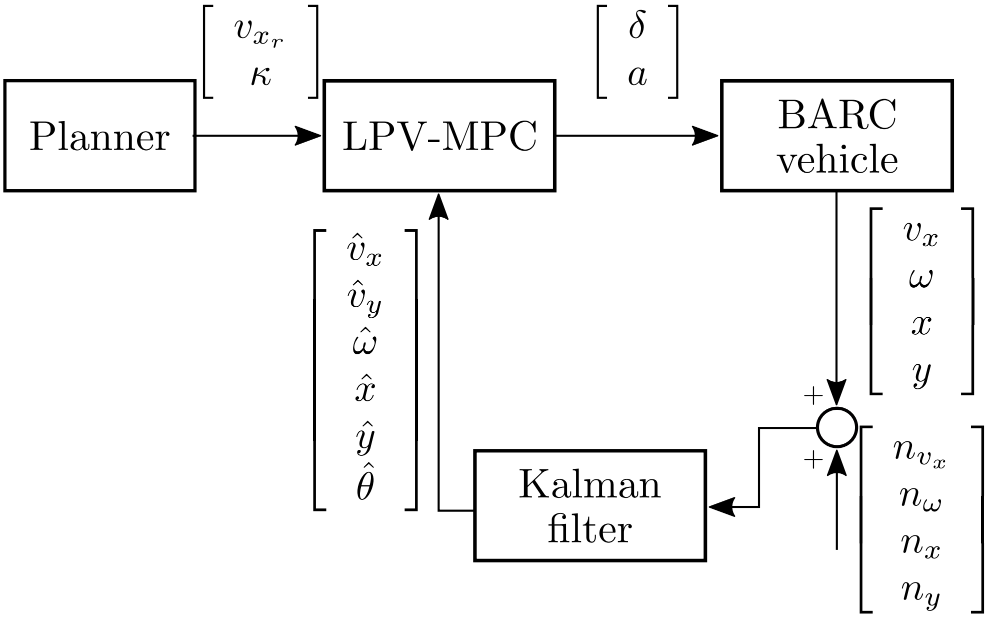

$\begin{bmatrix} v_{x_r} \\ \kappa \end{bmatrix}$
$\begin{bmatrix} \delta \\ a \end{bmatrix}$
Planner
LPV-MPC
BARC vehicle
$\begin{bmatrix} v_x \\ \omega \\ x \\ y \end{bmatrix}$
+
+
$\begin{bmatrix} n_{v_x} \\ n_\omega \\ n_x \\ n_y \end{bmatrix}$
Kalman filter
$\begin{bmatrix} \hat{v}_x \\ \hat{v}_y \\ \hat{\omega} \\ \hat{x} \\ \hat{y} \\ \hat{\theta} \end{bmatrix}$

FIGURE 8.2: Schematical view of the experimental set up

The model used in this chapter for simulating the BARC vehicle is presented in 7.15 with parameters in 7.1. Notice that v_y and θ are unmeasurable variables and then the Kalman filter algorithm is in charge of estimating them. In is this chapter, with the aim of improving the simulation, Gaussian noise has been introduced in the measured variables as

$$n_{(\cdot)} \sim N(0, Co_{(\cdot)}) \tag{8.1}$$

where $Co_{(\cdot)}$ is the signal covariance (see $n_{(\cdot)}$ vector in Figure 8.2).

8.3 LPV-MPC formulation

In this section, we present a novel formulation for the MPC technique using the LPV representation of the non-linear vehicle model. When the MPC technique performs the prediction of future vehicle states, it employs the LPV model. This imply that, at every time instant, an instantiation of the non-linear vehicle model computed with a known scheduling vector is required. Such a vector can be given by the trajectory planner or by the prediction made in the previous MPC optimization.

The model used in this chapter is the CBCD-LPV representation (4.13) where the vector of scheduling variables is $\zeta = \begin{bmatrix} v_x & v_y & \theta_e & \kappa & y_e & \delta \end{bmatrix}$. The use of this model allows to formulate the MPC problem as a quadratic optimization problem that is solved at each time k to determine the control actions considering that the values of x_k and u_{k-1} are known

$$\begin{aligned}
\min_{\Delta U_k} \quad \mathrm{J_k} = & \sum_{i=0}^{H_p-1} \Big((r_{k+i} - x_{k+i})^T Q (r_{k+i} - x_{k+i}) \\
& + \Delta u_{k+i} R \Delta u_{k+i} \Big) + x_{k+H_p}^T Q x_{k+H_p} \\
\text{s.t.} & \\
& x_{k+i+1} = x_{k+i} + \big(A(\zeta_{k+i}) x_{k+i} + B(\zeta_{k+i}) u_{k+i} \big) T_s \\
& u_{k+i} = u_{k+i-1} + \Delta u_{k+i} \\
& \Delta U_k \in \Delta\Pi \\
& U_k \in \Pi \\
& y_e \in [\overline{y_e}, \underline{y_e}] \\
& x_{k+0} = \hat{x}_k \ ,
\end{aligned} \tag{8.2}$$

where $\Pi = \{u_k | A_u u_k \leq b_u\}$ and $\Delta\Pi = \{\Delta u_k | A_{\Delta u} \Delta u_k \leq b_{\Delta u}\}$ constraint the system inputs and their variations, respectively. $x = \begin{bmatrix} v_x & v_y & \omega & \theta_e & s & y_e \end{bmatrix}^T$ is the state vector, $\hat{x}$ is the estimated state vector, $r = \begin{bmatrix} v_{x_r} & 0 & 0 & 0 & 0 & 0 \end{bmatrix}^T$ is the reference vector provided by the trajectory planner (see Section 10), $u = \begin{bmatrix} \delta & a \end{bmatrix}^T$ is the control input vector and H_p is the control prediction horizon. The tuning matrices $Q = Q^T \in \mathbb{R}^{6x6}$ and $R = R^T \in \mathbb{R}^{2x2}$, are semi-positive definite in order to obtain a convex cost function. The time discretization is carried out using Euler approach and the constant sampling time T_s.

8.4 System identification

In this section, we present the identification methodology used for adjusting the parameters of the vehicle dynamic model. The parameter estimation procedure considers the non-linear model (3.16) for the vehicle dynamics and the goal is to identify the tire stiffness coefficients C_f and C_r using a least-squares approach. The rest of parameters are assumed to be known for the particular vehicle and it is assumed to have at disposal M data samples. The identification procedure determines the unknown parameters that provides the minimum of the following objective function

$$\begin{aligned}\underset{C_f, C_r}{\text{minimize}} \quad \mathrm{J_k} = \sum_{i=0}^{M} \Big(& \frac{1}{\overline{v_x}}(v_{x_k} - \hat{v}_{x_k})^2 + \frac{1}{\overline{v_y}}(v_{y_k} - \hat{v}_{y_k})^2 \\ & + \frac{1}{\overline{\omega}}(\omega_k - \hat{\omega}_k)^2 \Big) \ ,\end{aligned} \tag{8.3}$$

where $\hat{v}_{x_k}$, $\hat{v}_{y_k}$ and $\hat{\omega}_k$ are the one-step predictions based on the non-linear equations (3.16) after the corresponding discretization in time and $\overline{v_x}$, $\overline{v_y}$ and $\overline{\omega}$ are their maximum dynamic values.

Note that since the computational cost of this optimization-based approach is high to be run in real-time, this is solved offline using IPOPT non-linear optimization solver [Wächter, Andreas, and Lorenz T. Biegler., 2006].

8.5 Racing results

The way of evaluating the planning and control strategies for racing is by first simulating the whole autonomous driving system and then testing it in a real framework. To do so, we have proposed a circuit where the objective is to minimize the lap time while fulfilling the road constraints. First, solving the racing trajectory planning (Section 10.2) in an offline way allows us to obtain optimal-based racing references which will be used by the controller to perform as fast as possible.

Then, at every sampling period, i.e. 30 Hz, the control problem (8.2) is solved to find the appropriate control actions (δ and a).

The LPV-MPC algorithm is codified in Python 2.7 programming language on ROS framework and solved in real time employing the Operator Splitting Quadratic Program (OSQP) solver [Stellato, Bartolomeo, et al., 2018] running on a DELL inspiron 15 (Intel core i7-8550U CPU @ 1.80GHzx8). The tuning aims to minimize the velocity and lateral errors while computing smooth control actions. The diagonal terms of the weighting matrices in the cost function and prediction horizon of (8.2), found by iterative tuning until the desired performance is achieved, are

$$\begin{aligned} &Q = [\ 120 \quad 1 \quad 1 \quad 40 \quad 0 \quad 800\ ], \\ &R = [\ 6 \quad 2\ ], \\ &H_p = 20\ . \end{aligned} \tag{8.4}$$

The MPC constraints are defined as

$$A_u = \begin{bmatrix} 1 & 0 \\ -1 & 0 \\ 0 & 1 \\ 0 & -1 \end{bmatrix},\ b_u = \begin{bmatrix} 0.249 \\ 0.249 \\ 4 \\ 1 \end{bmatrix}, \tag{8.5a}$$

$$A_{\Delta u} = \begin{bmatrix} 1 & 0 \\ -1 & 0 \\ 0 & 1 \\ 0 & -1 \end{bmatrix},\ b_{\Delta u} = \begin{bmatrix} 0.05 \\ 0.05 \\ 0.5 \\ 0.5 \end{bmatrix}, \tag{8.5b}$$

$$\overline{y_e} = -\underline{y_e} = 0.4m\ . \tag{8.5c}$$

Before validating the presented algorithms in an experimental way, they have been tested in simulation. Besides, the dynamic system has been properly identified using the identification method presented in (8.3). Next subsections present simulation and experimental results from a control perspective.

8.5.1 Simulation test

All simulations are carried out using the ROS framework where control and estimation algorithms are running. We use a non-linear model formulation (7.15) representing the real car for simulation.

This model considers a more precise lateral tire force formulation using the simplified Magic Formula [Pacejka, Hans., 2005] for modeling the non-linear relationship between front and rear slip angles and lateral tire forces. Also, a more accurate computation of the tire slip angles is given and white Gaussian noise magnitudes are added to measured states with zero mean and covariances

$$Co_{v_x} = 1 \times 10^{-6} \ , \ Co_{\omega} = 4 \times 10^{-8} \ , \ Co_{x} = 4 \times 10^{-6} \ , \ Co_{y} = 4 \times 10^{-6} \ . \qquad (8.6)$$

As shown in Figure 8.2, three algorithms are executed every 30 ms. First, the controller instantiates the LPV model matrices for the prediction stage. Then, the optimal problem is solved using for that the current state variables and the references coming from the planner (see Section 10). Once, the optimal control actions are computed they are applied to the simulation vehicle. As a consequence, the vehicle change its state and this is measured by the net of sensors. The sensors are simulated to be realistic by having Gaussian noise. Finally, the state estimator algorithm deals with the current available measurements to obtain a precise and complete state vector.

Figure 8.3 depicts the longitudinal velocity profiles, the reference and the response, for three laps, i.e. the acceleration lap and two consecutive laps. It can be seen that the controller has some troubles when the vehicle is accelerating but it works acceptably after 27 seconds. Such problems are related to the lack of modeling of the traction motor resulting in the controller unable to follow the speed reference perfectly.

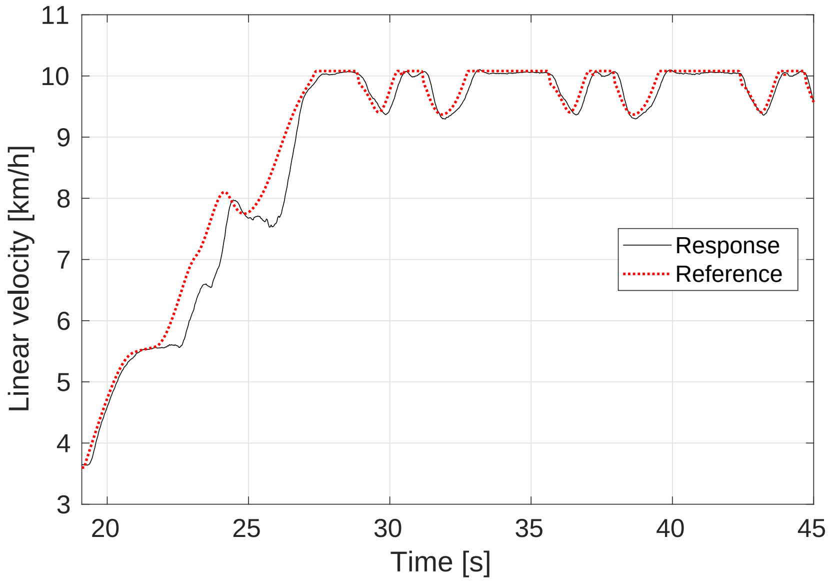


FIGURE 8.3: Linear velocities in simulation. The reference is provided by the NL-MP Planner. The response is the result after treating the measured data from the vehicle sensors

The good performance of the controller can be seen in Figure 8.4. It is seen that the controller is able to force the convergence of the errors to zero in spite of the complexity of driving in a high lateral acceleration situation.

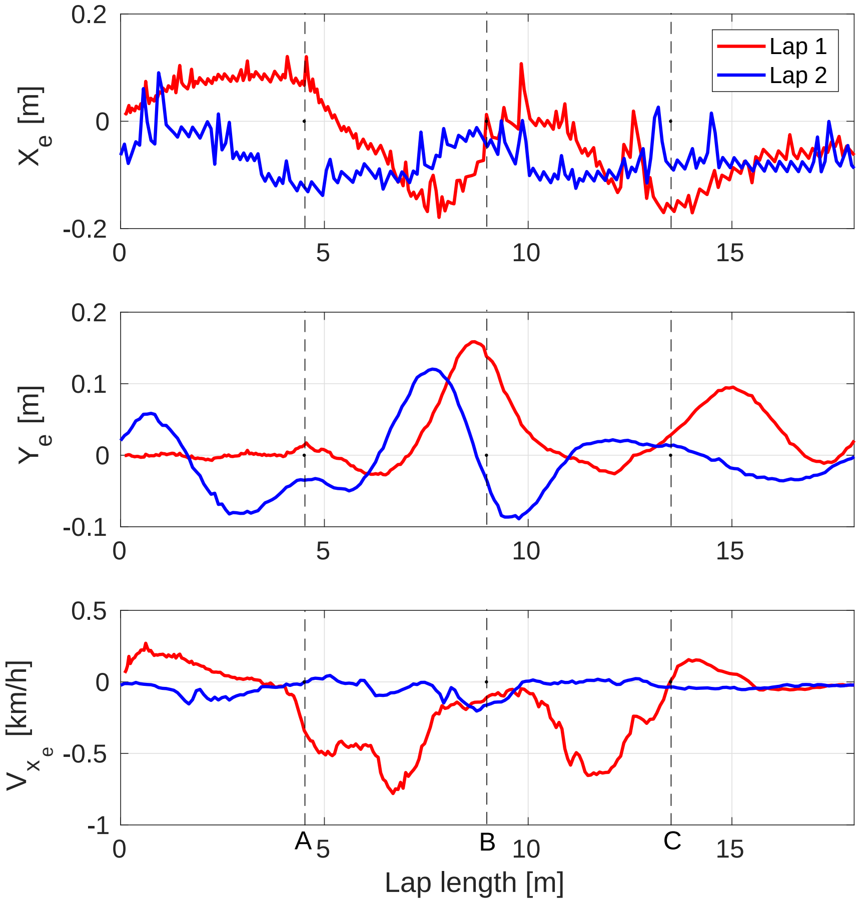


FIGURE 8.4: Errors achieved during two simulated racing laps

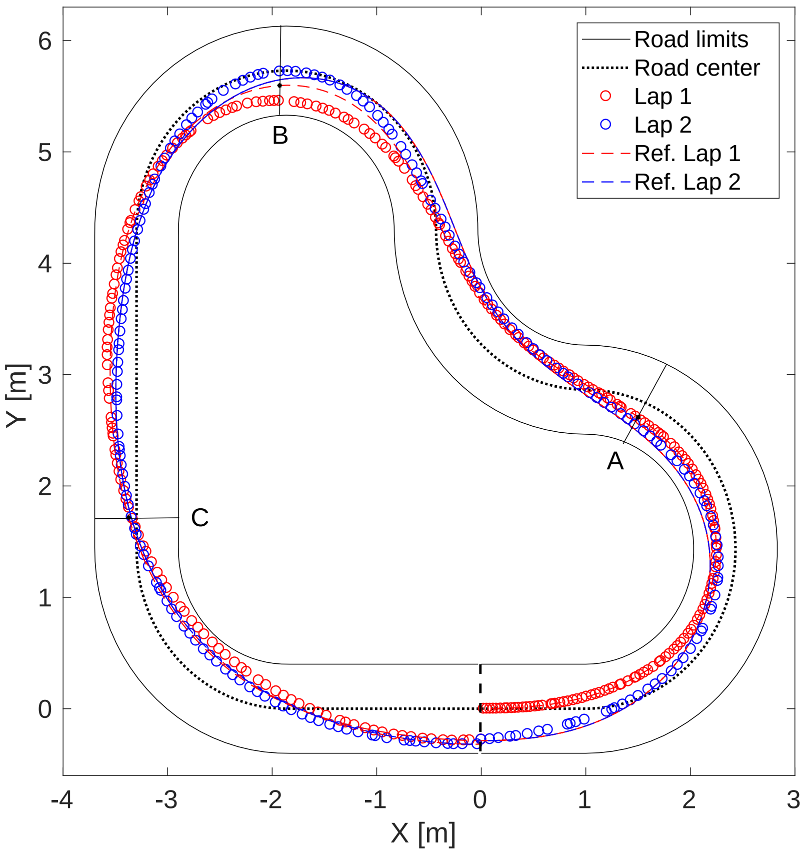
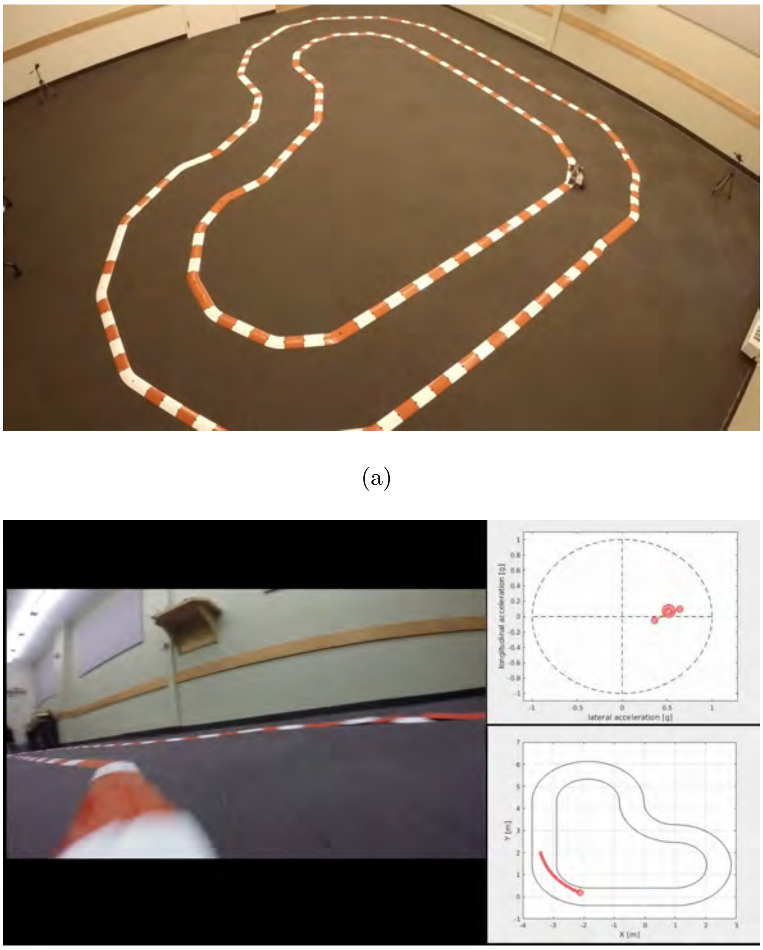


FIGURE 8.5: Two racing laps in simulation controlling the simulation vehicle model

8.5.2 Experimental test

Using the same setup than in simulation, we assess the performance of the LPV-MPC technique in an experimental way using the BARC platform (see Figure 8.1). The racing planning is the same than in simulation tests since the track is the same. A resulting video can be watched at `https://www.youtube.com/watch?v=MXz9InvoVBw`.



(a)

(b)

Figure 8.6: In field test: (a) Bird view. (b) Vehicle frontal view plus acceleration circle and vehicle in track localization

The result of the controlled longitudinal velocity for three laps is shown in Figure 8.7. It shows a good reference tracking although with a bit of steady state error from $t = 27\ s$ and up. In spite of the identification performed, it is possible to attribute this error to modeling and estimation errors.

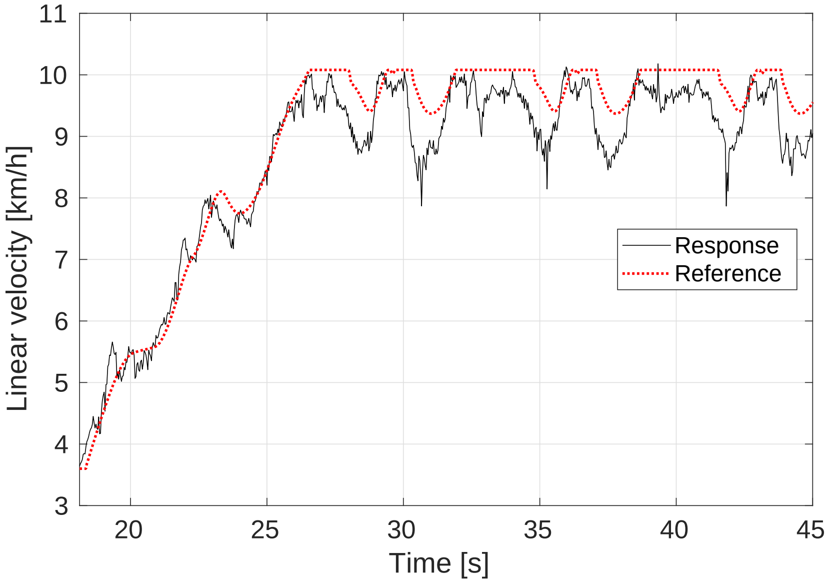


FIGURE 8.7: Linear velocities in experimental test. The reference is provided by the NL-MP Planner. The response is the result after treating the measured data from the vehicle sensors

Figure 8.8 presents the resulting trajectory during the test. The mean lap time achieved disregarding the first accelerating lap is 6.97 s. Some jumps can be observed along the vehicle way which are totally attributed to issues in the GPS system. The indoor GPS works by using ultrasonic sensors allowing to have up to ~ 2 cm of error in localization using triangulation. However, the ultrasonic sensors sometimes experience interference by external signals which results in little jumps in localization. These jumps are treated by the Kalman filter, however, when they are very large and continuous it is very difficult to filter them.

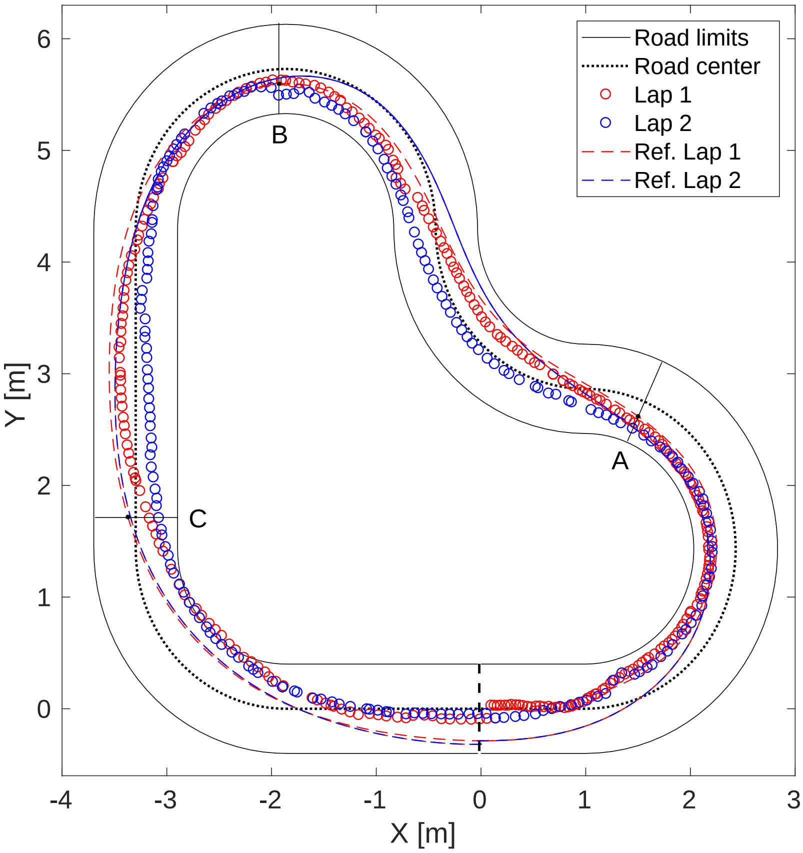


FIGURE 8.8: Two experimental racing laps using the BARC vehicle

The computational time of this approach is one of the most interesting advantages. Figure 8.9 shows the elapsed time when computing the LPV-MPC strategy with a mean time of 0.0149 s for a prediction horizon of 20 steps. The peaks that can be seen outside the permitted area (real-time constraint) are due to sudden locations jumps of the indoor-GPS system, causing the optimizer to solve a more complex and therefore more computationally expensive problem. In these particular cases, the applied control action corresponds with the one predicted in the previous optimization.

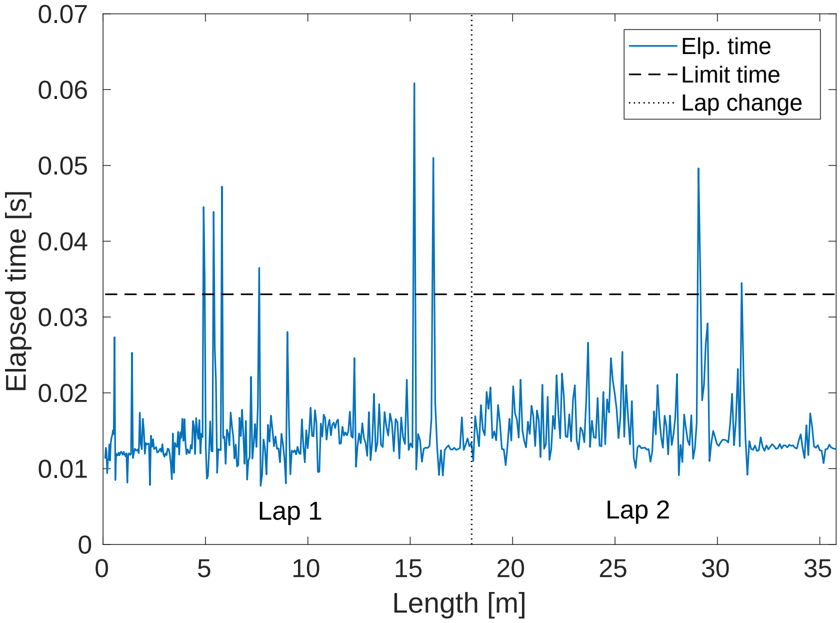


Figure 8.9: LPV-MPC computational time during two experimental racing laps

Finally, we can perform a comparison against the control strategies presented in Rosolia, Ugo, and Francesco Borrelli. [2019] and Rosolia, Ugo, Xiaojing Zhang, and Francesco Borrelli. [2019]. In both references and this work, the results are obtained after testing under the same conditions, i.e. the same track and the same vehicle. The resulting comparison shows us a lap time reduction using the proposed method in this work. The best lap time achieved in these works is more than 7 s however, in the presented strategy the mean lap time achieved disregarding the first accelerating lap is 6.97 s. In addition, the average computational time is also improved being in this work 14.9 ms while in the other works is around 30 ms.

8.6 Conclusions

In this chapter, an LPV-MPC strategy has been proposed as a novel approach to solve autonomous driving control problems under realistic conditions in real-time. In addition, using racing-based references provided by an external planner the controller makes the vehicle to perform in racing mode. For a good control performance, an offline identification of unknown vehicle coefficients has been carried out. The strategies are tested in simulation and in real experiments that show potential and similar results among them, thus strengthening the task of the simulator. In the real test, we showed the contribution of the controller which is able to solve a 20 steps prediction at 33 Hz and thus follow the trajectory although with a certain error due to the non-modeled dynamics (see `https://www.youtube.com/watch?v=MXz9InvoVBw`). The disadvantage found in this strategy is the system initialization due to the need to instantiate the LPV model.

Chapter 9

Fast Zonotope-Tube-based LPV-MPC

The content of this chapter is based on the following works:

- [Alcalá, Eugenio, et al., 2020.B]. Alcalá, E., Puig, V., Quevedo, J. & Sename, O. Fast Zonotope-Tube-based LPV-MPC for Autonomous Vehicles. *International Journal of Control*, 2019 (submitted).

9.1 Introduction

MPC is an effective control strategy that allows to deal with constrained problems and multiple-input multiple-output systems. However, dealing with uncertainty or disturbances is something that conventional MPC algorithms do not handle and then, robust MPC (RMPC) formulations have to be considered. In [Mayne, D., 2016], the author presents a review on current MPC formulations with their limitations and future development directions.

During the last years, two differentiated and consolidated approaches for RMPC have been addressed: Min-max MPC and Tube-based MPC. On the one hand, the min-max or worst-case problem aims to find the optimal solution based on minimizing the maximum value of the cost function. In [Liu, C., Li, H., Gao, J., & Xu, D., 2018], authors present a robust self-triggered min-max MPC approach for constrained non-linear systems with both parameter uncertainties and disturbances. On the other hand, tube-based MPC is based on computing a region around the nominal prediction that ensures the state of the system to remain inside under any possible uncertainty and disturbance [Brunner, F. D., Heemels, M., & Allgöwer, F., 2016].

Different variations of tube-based MPC have been presented in the recent literature. In Gonzalez, R., Fiacchini, M., Alamo, T., Guzmán, J. L., & Rodríguez, F. [2011], an online tube-based MPC is proposed to ensure robustness for time varying systems with additive uncertainties. Reachable sets are computed online using a explicit LQR-based controller and polytopes which results to increase the computation time. Authors defend a reduction on conservatism using this approach. In Sakhdari, B., Shahrivar, E. M., & Azad, N. L. [2017], the authors propose a RMPC approach for Adaptive Cruise Control. They emphasize the high computational effort of computing the reachable set using polytope-based operations and present an approach based on singular value decomposition. A different scheme for RMPC strategy is presented in Darup, M. S., & Mönnigmann, M. [2018]. The authors show the significantly reduction of numerical effort compared to standard RMPC by identifying a terminal set on which a LQR-based control law is sufficient. In Kim, Y., Zhang, X., Guanetti, J., & Borrelli, F. [2018], authors present a RMPC with adjustable uncertainty sets. They propose a less conservative approach in contrast to standard RMPC problems and illustrate the effectiveness on a cooperative adaptive cruise control application.

In this paper, we present a robust tube-based MPC approach faster than the state of the art strategies being able to reject large exogenous disturbances. This optimal algorithm uses a LPV vehicle model for simulating future vehicle behaviour. We summarize the innovative points with respect to the state of the art as follows:

- Using zonotope theory we are able to reduce the computational cost of basic operations, i.e. Minkowski sum and difference, in comparison with current polytopes-based operations.
- The use of fast zonotope-based calculations allows us to not approximate the tube, hence obtaining a less conservative result.

- Using H_∞ control design to obtain a gain scheduling polytopic LPV local controller allows to reject large exogenous disturbances acting over the vehicle. Current tube-based MPC techniques in the state of the art are using LQR technique.
- Currently, most of the works based on robust MPC design use a local controller than runs at the same frequency than the nominal controller (MPC). In this work, we propose a faster loop to achieve a faster and better performance of the control scheme.

9.2 Problem statement

This chapter addresses the problem of designing an online tube-based MPC for controlling a simulated vehicle plant formulated as the following non-linear system

$$x^+ = f(x, u) + e \ , \tag{9.1}$$

where $x \in \mathbb{R}^s$ is the state vector, $u \in \mathbb{R}^n$ the input vector and $f(x, u)$ represents the non-linear map obtained after modeling the physics of the real system. Vector $e \in \mathbb{R}^s$ contains all the unmodeled physics of the real plant and exogenous disturbances acting over it. Note that, in this article the notation x^+ is used for the successor of vector x, i.e. $x = x(k)$ and $x^+ = x(k+1)$.

At this point, the following uncertain, LPV, discrete-time system is formulated

$$x^+ = A_\zeta x + B_\zeta u + w \ , \tag{9.2}$$

where A_ζ and B_ζ are the LPV state space matrices which depend on the varying scheduling vector ζ and $w \in \mathbb{R}^s$ is the exogenous disturbance vector.

Remark 9.1. The system $x^+ = A_\zeta x + B_\zeta u$ in (9.2) is an exact realization of $x^+ = f(x, u)$ in (9.1) inside the considered polytopic region and the scheduling vector ζ is known at each sampling time being a combination of system states and inputs.

The state, control and disturbance vectors are bounded as

$$x \in X \ , \quad u \in U \ , \quad w \in W \ , \tag{9.3}$$

where $X \subseteq \mathbb{R}^s$, $U \subseteq \mathbb{R}^n$ and $W \subseteq \mathbb{R}^s$.

To achieve the tracking-robust control purpose, two problems are handled:

- Reference tracking control problem. The LPV-MPC strategy deals with the following system

  $$\check{x}^+ = A_\zeta \check{x} + B_\zeta u^* \ , \tag{9.4}$$

  for tracking the dynamic references while handling system constraints. This system will be referred as nominal model throughout the work.

- Robust control problem. The main idea is to compensate the mismatch between the states of (9.1) and the nominal state vectors (9.4). This difference is computed as

  $$e = x - \check{x} \ , \tag{9.5}$$

  where $e \in \mathbb{R}^s$ is the error state. In order to minimize such a mismatch, the following control law is considered

  $$\begin{aligned} u &= u^* + u_\infty \\ u_\infty &= K_\zeta^\infty e \ , \end{aligned} \tag{9.6}$$

  where u^* represents the optimal action from the R-LPV-MPC, u_∞ is the corrective action and K_ζ^∞ is the state feedback gain computed online as a gain scheduling controller using H_∞-based LMIs for the design.

  Finally, the closed loop error dynamics are defined as

  $$e^+ = x^+ - \check{x}^+ = (A_\zeta + B_\zeta K_\zeta^\infty)e + w \ . \tag{9.7}$$

9.3 Control and simulation vehicle models

The Driverless UPC Car [1] is a development platform for autonomous racing (see Figure 9.1). This has been properly modified to operate autonomously under racing specifications. The vehicle counts with a complete net of sensors for performing localization and environment understanding. A fusion of IMU, encoders and GPS data is made by a Kalman filter in order to estimate a more precise values of the vehicle states.

[1] https://driverless.upc.edu/

FIGURE 9.1: UPC Driverless vehicle

The Cartesian dynamic LPV representation presented in Section 4.5 is modified to improve the control performance by introducing two new integral states with the aim of completely remove the steady state error. In addition, the simplified Magic Formula LPV model (4.6) is used to represent accurately the lateral tire dynamics. Then, denoting the state and control vectors, respectively, as

$$x = \begin{bmatrix} v_x \\ v_y \\ \omega \\ x \\ \theta \end{bmatrix}, \; u = \begin{bmatrix} \delta \\ a \end{bmatrix}, \tag{9.8a}$$

the continuous-time LPV matrices are expressed as

$$A_\zeta = \begin{bmatrix} A_{11} & A_{12} & A_{13} & 0 & 0 \\ 0 & A_{22} & A_{23} & 0 & 0 \\ 0 & A_{32} & A_{33} & 0 & 0 \\ -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 & 0 \end{bmatrix} \tag{9.8b}$$

and

$$B_\zeta = \begin{bmatrix} B_{11} & B_{12} \\ B_{21} & 0 \\ B_{31} & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix} , \tag{9.8c}$$

where A and B coefficients are varying parameters defined in (4.9). Note that, after increasing the system degree the scheduling vector is still defined as

$$\zeta := [v_x, v_y, \delta] \ . \tag{9.9}$$

The vehicle parameters used for this model are properly defined in Table 9.1 and Table 9.2 shows the coefficients used for representing the LPV formulation of the tire stiffness coefficient (4.7) in the lateral tire model.

TABLE 9.1: Dynamic model parameters of the Driverless UPC Car

Parameter	Value	Parameter	Value
l_f	$0.902\ m$	l_r	$0.638\ m$
m	$196\ kg$	I	$93\ kg\ m^2$
d_f	8.255	c_f	1.6
b_f	6.1	μ	1.4
d_r	8.255	c_r	1.6
b_r	6.1	ρ	$1.225\ kg\ m^3$
C_{dAf}	1.64	g	$9.81\ \frac{m}{s^2}$
C_{dAl}	1.82		

9.3.1 Simulation vehicle

For simulation purposes we use a higher fidelity vehicle model. Unlike the model used for control (9.8), this considers the non-linear simplified Magic Formula model (3.14) where

TABLE 9.2: Polynomial parameters of (4.7) for the front and rear tires (upper indexes f and r)

Parameter	Value	Parameter	Value
n	4	ϵ	10^{-4}
p_1^f	-2.167 $\times 10^6$	p_2^f	1.284 $\times 10^6$
p_3^f	-0.288 $\times 10^6$	p_4^f	0.029 $\times 10^6$
p_5^f	15.038		
p_1^r	-2.130 $\times 10^6$	p_2^r	1.198 $\times 10^6$
p_3^r	-0.252 $\times 10^6$	p_4^r	0.024 $\times 10^6$
p_5^r	14.551		

the parameters b, c and d define the shape of the semi-empirical curve. In addition, a more accurate calculation of the tire slip angles is provided and is sensitive to certain external disturbances. This is given by

$$\begin{aligned}
\dot{v}_x &= a_r + \frac{-F_{yf}\sin\delta - F_{df}}{m} + \omega v_y - g\sin\varphi \\
\dot{v}_y &= \frac{F_{yf}\cos\delta + F_{yr} - F_w}{m} - \omega v_x \\
\dot{\omega} &= \frac{F_{yf}l_f\cos\delta - F_{yr}l_r - F_w(l_f - l_r)}{I} \\
\alpha_f &= \delta - \tan^{-1}\left(\frac{v_y}{v_x} - \frac{l_f\omega}{v_x}\right) \\
\alpha_r &= -\tan^{-1}\left(\frac{v_y}{v_x} + \frac{l_r\omega}{v_x}\right) \\
F_{yf} &= d_f\sin\left(c_f\tan^{-1}(b_f\alpha_f)\right) \\
F_{yr} &= d_r\sin\left(c_r\tan^{-1}(b_r\alpha_r)\right) \\
F_{df} &= \mu m g + \frac{1}{2}\rho C_{dAf}v_x^2 \\
F_w &= \frac{1}{2}\rho C_{dAl}v_w^2
\end{aligned} \quad , \qquad (9.10)$$

where φ and v_w are exogenous disturbances and represent the longitudinal road slope and the lateral wind velocity, respectively. C_{dAl} is the product of drag coefficient and vehicle lateral cross sectional area and C_{dAf} is the product of drag coefficient and vehicle frontal cross sectional area. Parameters d_f,d_r,c_f,c_r,b_f and b_r are the simplified Magic Formula model constants.

9.4 Online Tube-based LPV-MPC using Zonotopes

In this section, we present the zonotope-tube-based LPV-MPC scheme using zonotopes to significantly reduce the cost of computing the tube (see Figure 9.3). The main purpose of this strategy is to achieve robust stability and robust performance in the presence of modeling errors and exogenous disturbances.

A zonotope, represented as $\langle c_w, R_w \rangle$ with the center $c_w \in \mathbb{R}^n$ and the generator matrix $R_w \in \mathbb{R}^{n \times p}$, is a particular form of a polytope defined as the linear image of the unit cube [Girard, A., 2005, Girard, A., Le Guernic, C., & Maler, O., 2006]

$$\langle c_w, R_w \rangle = \{c + Rx : \|x\|_\infty \leq 1\}. \tag{9.11}$$

Note that, the linear image of a zonotope $W = \langle c_w, R_w \rangle$ by a compatible matrix M is defined as

$$M \circ W = M \circ \langle c_w, R_w \rangle = \langle M c_w, M R_w \rangle. \tag{9.12}$$

Along this chapter, zonotopes are treated as centered zonotopes denoted by $\langle 0, R_w \rangle$. Then, the linear image is defined as

$$M \circ W = \langle 0, M R_w \rangle \tag{9.13}$$

and the Minkowski sum of two centered zonotopes $W = \langle c_w, R_w \rangle$ and $G = \langle c_g, R_g \rangle$ is defined as

$$W \oplus G = \langle 0, [R_w, R_g] \rangle. \tag{9.14}$$

In this work, zonotopes are used to compute reachable sets and therefore, the tube to implement the robust MPC architecture. The main reason for the use of zonotopes lies in their simplicity to operate with sets. Therefore, a set operation such as the Minkowski sum is reduced to a simple matrix addition. Note that, the use of Minkowski sum or difference of two polytopes is costly, however, using zonotopes the computational cost is reduced allowing a fast computation of basic sets operations.

The zonotope-tube-based LPV-MPC requires of some steps to complete the strategy. First of all, a polytopic state feedback controller is computed offline using a H_∞-LMI based problem (Section 9.4.1). Furthermore, the maximal robust invariant set and the terminal cost are computed also offline (Section 9.4.2). Then, at every control iteration, the state feedback gain K_ζ^∞ is computed as a linear function of the scheduling vector ζ. Afterwards, the tube is calculated as the maximum reachable values of states and inputs taking into account the previous control gain (Section 9.4.3). Finally, the MPC problem is solved where the input and state constraints are updated defining an adaptive and less conservative tube (Section 9.4.4). Hereafter, the introduced scheme will be explained in detail.

9.4.1 Local controller design

In this section, the offline design and online computation of the state feedback LPV controller is addressed. We aim to design a controller to reduce the mismatch between the states of system (9.1) and the nominal state vectors (9.4) even under the presence of disturbance. In the most recent literature, the LQR control strategy is one of the most used techniques when dealing with determining a local control structure to make the MPC a robust strategy [Gonzalez, R., Fiacchini, M., Alamo, T., Guzmán, J. L., & Rodríguez, F., 2011, Sakhdari, B., Shahrivar, E. M., & Azad, N. L., 2017, Darup, M. S., & Mönnigmann, M., 2018].

However, when dealing with systems subject to external disturbances, the LQR technique becomes less efficient against such system variations and it is when different methodologies emerge, as the case of H_∞ strategy, resulting more suitable for the application. On one hand, the H_∞ control problem allows to reduce the impact of an external disturbance to the system output. On the other hand, the obtained control gains may be large which is sometimes undesirable in applications.

9.4.1.1 Offline design

In this work, a polytopic LPV H_∞ controller is designed by means of minimizing the infinity norm of the transfer function between the disturbance signal and the control variables.

The following LPV system is considered for control design purposes

$$\begin{aligned} \dot{x} &= A_{\zeta}x + Bu + Ed \\ z &= Cx + D_1u + D_2d \end{aligned} , \tag{9.15}$$

where state matrix A is function of the scheduling vector ζ, input matrix B is a linearization of B_{ζ} in (9.4) around $\delta = 0$, E is the disturbance input matrix, d represents the exogenous disturbance vector, z represents the controlled variables vector and C, D_1 and D_2 are constant matrices of appropriate dimensions.

From the LPV system (9.15) and considering the state feedback control law $u = K_{\zeta}^{\infty}x$, we can formulate the transfer function from d to z as

$$G_{zd} = (C + D_1K_{\zeta})(sI - (A_{\zeta} + BK_{\zeta}^{\infty}))^{-1}E + D_2 \ . \tag{9.16}$$

Hence, the proposed problem consists on finding a polytopic state feedback gain K_{ζ} such that

$$\|G_{zd}\|_{\infty} \leq \gamma \ , \tag{9.17}$$

holds for the attenuation scalar γ. To find the solution, we solve the H$_{\infty}$ problem in continuous time via LMIs using the polytopic approach as suggested in Duan, Guang-Ren, and Hai-Hua Yu. [2013] given by

$$\begin{aligned} &\min_{X,W_i} \quad \gamma \\ &\text{s.t.} \\ &\qquad \begin{bmatrix} S & E & XC^T + W_i^T D_2^T \\ * & -\gamma I & D_1^T \\ * & * & -\gamma I \end{bmatrix} \leq 0 \\ &\qquad S = A_iX + BW_i + XA_i^T + W_iB^T \\ &\qquad \forall i = 1, ..., N \ , \end{aligned} \tag{9.18}$$

being the solutions $X = P^{-1}$ and $W_i = KP^{-1}$ where P represents the common Lyapunov matrix for the polytopic LPV system. Parameter N represents the number of vertexes in the polytope. Then, the resulting vertices of the new polytopic controller are obtained by $K_i = W_iX^{-1}$.

9.4.1.2 Online computation

At each control iteration, the state feedback LPV control gain K_ζ is updated based on the current value of the scheduling vector ζ. To do so, a convex combination of polytopic controller vertices, i.e. the set of K_i, is computed as

$$K_\zeta = \sum_{i=1}^{N} \mu_i(\zeta) K_i \ , \tag{9.19}$$

where $\mu_i(\zeta)$ is given by

$$\mu_i(\zeta) = \prod_{j=1}^{n_\zeta} \xi_{ij}(\eta_0^j, \eta_1^j) \ , \qquad \forall i = 1, ..., N \ , \tag{9.20}$$

with

$$\begin{aligned} \eta_0^j &= \frac{\overline{\zeta_j} - \zeta_j(k)}{\overline{\zeta_j} - \underline{\zeta_j}} \\ \eta_1^j &= 1 - \eta_0^j \ , \end{aligned} \tag{9.21}$$

where each scheduling variable ζ_j is known and varies in a defined interval $\zeta_j \in \left[\underline{\zeta_j}, \overline{\zeta_j}\right] \in \mathbb{R}$, n_ζ is the number of scheduling variables and $\xi_{ij}(\cdot)$ corresponds with the function that performs the N possible combinations. In addition, next conditions must be satisfied

$$\sum_{i=1}^{N} \mu_i(\zeta) = 1, \ \ \mu_i(\zeta) \geq 0, \ \forall \zeta \in \Theta \ . \tag{9.22}$$

9.4.2 Terminal Robust Invariant Set & Cost

A commonly used approach to guarantee asymptotic stability of deterministic MPC consists in incorporating both a terminal cost, P, and a terminal constraint set, χ_f. In this section, we propose an offline method to compute both P and χ_f. Thus, the closed-loop system convergence to the origin is ensured if

- $Q = Q^T \geq 0$, $R = R^T > 0$ and $P > 0$
- The sets X, χ_f and U are polytopes containing the origin
- The terminal cost is a Lyapunov function in χ_f
- χ_f is the minimal robust positively invariant (mRPI) set, $\chi_f \subseteq X$.

On one hand, the computation of P is carried out by solving the LMI-based H_∞ problem (9.18). Furthermore, a set of polytopic robust controllers is found. The optimal problem solutions, i.e. X and W_i, are used to calculate the controllers at the vertices of the polytope as $K_i = W_i X^{-1}$. Note that the Lyapunov function in the optimization problem is found to be equal to X^{-1} and will be use later in (9.32) as P.

On the other hand, the terminal set χ_f will be the mRPI set if and only if it is contained in any closed RPI set and is convex and unique. Then, the mRPI set for the stable and disturbed system (9.7) is computed by the following recursive procedure

$$\begin{aligned}
&\text{1. Initialization:}\\
&\quad \Omega_0 = E_{k^*}\\
&\text{2. Loop:}\\
&\quad \Omega_{k+i} = \mathcal{A}(\Omega_k) \oplus W\\
&\text{3. Termination condition:}\\
&\quad \text{stop when } \Omega_{k+1} = \Omega_k. \text{ Set } \chi_f = \Omega_{k+1}
\end{aligned} \quad , \tag{9.23}$$

where $\mathcal{A}(\cdot)$ is the set mapping defined as

$$\mathcal{A}(\Omega_k) = \mathrm{Conv}\{\bigcup_{i=1}^{N}(A_i + BK_i^{\infty})\Omega_k\}. \tag{9.24}$$

Note that, Conv$\{\cdot\}$ represents the convex hull and is used to compute the one-step reachable set for the polytopic system case. This allows to preserve the convexity of the resulting set within the recursive iterations.

However, this recursive approximation to compute the mRPI set is intractable and not realistic since we may need infinite iterations to reach the termination condition. For that reason, in [Tan, Junbo, et al., 2019], the authors propose an outer approximation method for computing the mRPI set with a given precision. This approach consists on replacing the termination condition in (9.23) by the condition of terminating when there exist a $k^{\dagger}$ iteration such that

$$\mathcal{A}^{k^{\dagger}}(\Omega_0) \subseteq \mathbb{A}_p^{n_x}(\epsilon), \tag{9.25}$$

where $\mathbb{A}_p^{n_x}(\epsilon) = \{x \in \mathbb{R}^{n_x} : \|x\|_p \leq \epsilon\}$ defines a ball of arbitrary small size. Therefore, in such an article, it is concluded that the set $\Omega_{k^{\dagger}}$ is an outer approximation of the mRPI set Ω_∞ with the given precision $\mathbb{A}_p^{n_x}(\epsilon)$ as well as an RPI set too.

In addition, the initialization condition in (9.23) is still not defined. To find E_{k^*}, which is an RPI set for the system (9.7), it is necessary to solve the following iterative algorithm where there exist a finite k^* such that the termination condition is reached

1. Loop:

$$\begin{aligned}
&\mathcal{A}(E_k) = \text{Conv}\{\bigcup_{i=1}^{N}(A_i + BK_i^\infty)E_k\} \\
&\overline{E}_{k+1} = \mathcal{A}(E_k) \oplus W \\
&E_{k+1} = \text{Conv}\{\overline{E}_{k+1} \bigcup E_k\}
\end{aligned} \quad . \tag{9.26}$$

2. Termination condition:

   stop when $E_{k^*+1} = E_{k^*}$

Furthermore, given the stabilized system (9.7), the initial convex set $E_0 \supseteq \Omega_\infty$ can be computed as

$$E_0 = \sum_{i=0}^{p^*-1} \mathcal{A}^i(B(r)) \oplus \frac{p^*\xi}{1-\xi} B(r), \tag{9.27}$$

where $\xi \in (0, 1)$, $p^* \in \mathbb{N}$ and $B(r) = \{x \in \mathbb{R}^{n_x} : \|x\|_\infty \leq r\}$ is a box containing W. Note that, we should find a proper E_0 such that $\mathcal{A}^k(B(r)) \subseteq \xi B(r)$ holds for $k \geq p^*$.

9.4.3 Online Reachable Sets

This section addresses the reachable sets calculation also known as the one-step forward-reachable set computation. These sets define the problem of finding the set of states that can be reached from a given set of states in a set of finite steps [Borrelli, F., Bemporad, A., & Morari, M., 2017] .

In this approach, the main idea of using reachability theory is to bound the maximum achievable values for the mismatch error (9.7) between the prediction model and the real measurements at every sampling time.

To this aim, the one-step robust reachable set from the set Φ is denoted as

$$Reach(\Phi, W) = \{y : \exists x \in \Phi, \exists u \in U, \exists w \in W \ s.t. \ y = (A_\zeta + B_\zeta K_\zeta^\infty)x + w\}. \tag{9.28}$$

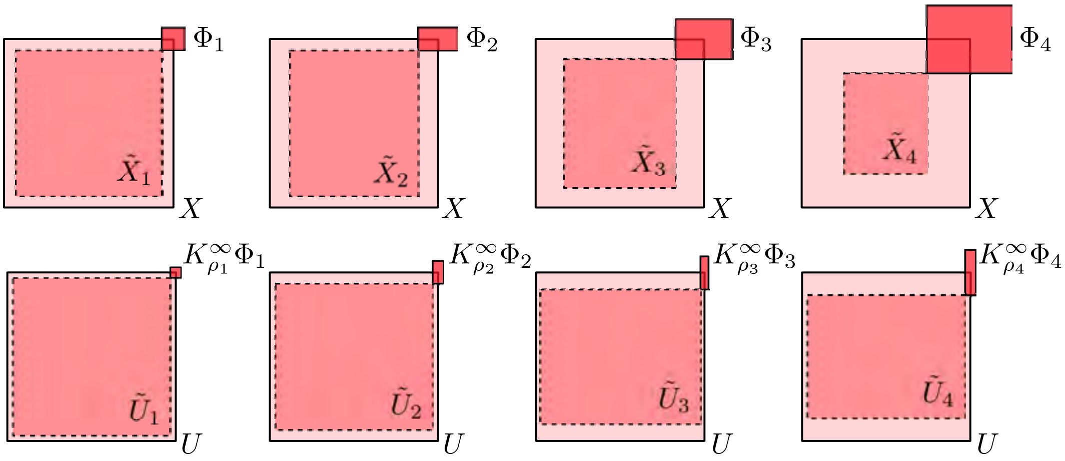


Figure 9.2: Example of reachable sets and new MPC constraints computation for a prediction horizon of four steps and considering two system states and two control inputs. State constraints ($\tilde{X}_k$) are depicted in the upper row while input constraints ($\tilde{U}_k$) calculations are shown in the lower row

Note that, by using polytopic notation, the robust reachable set Reach(Φ, W) can be compactly written as

$$Reach(\Phi, W) = \{((A_\zeta + B_\zeta K^{\infty}_{\zeta}) \circ \Phi) \oplus W\}. \tag{9.29}$$

Then, denoting the first initial reachable set as a null zonotope ($\Phi_0 = \langle 0_{n\times 1}, 0_{n\times p}\rangle$) and the disturbance set as a constant predefined zonotope ($W = \langle c_w, R_w\rangle$), at every sampling time k a group of reachable sets is computed by

$$\begin{array}{l} \Phi_{k+i+1} = (A_{\zeta_{k+i}} + B_{\zeta_{k+i}} K^{\infty}_{\zeta_{k+i}})\Phi_{k+i} \oplus W \\ \forall i = 0, ..., H_p \end{array}, \tag{9.30}$$

where H_p is the prediction horizon of the MPC strategy. Note that, each reachable set depends on its past realisation, the current scheduling vector for computing system matrices and controller and the uncertainty/disturbance set W.

Finally, these reachable sets are used for computing the concatenation of consecutive resulting state/input sets in the prediction horizon, known as tube (see Figure 9.2).

9.4.4 MPC design

Considering the previous discussions about the terminal conditions, the local controller and the reachable sets, in this section, we focus our attention on the tube-based MPC

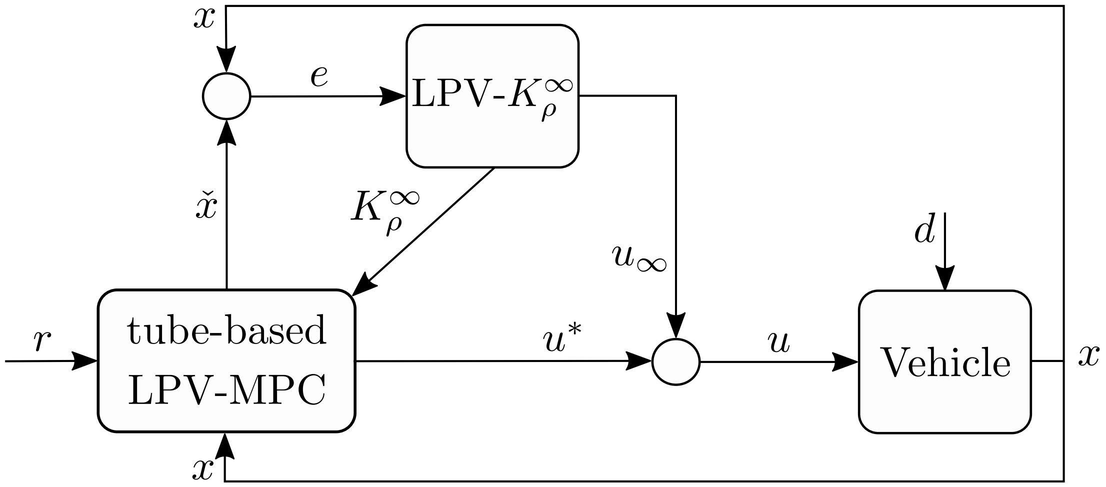


FIGURE 9.3: Robust control scheme composed of a nominal controller (tube-based LPV-MPC) and a local corrective controller (LPV-K_ζ^∞)

implementation. Figure 9.3 shows the complete scheme used in this work. Note that, the MPC strategy is in charge of controlling the nominal system while the differences between the real system and the nominal one are compensated by the local controller. Such a difference may be produced by external sources as a exogenous disturbances, unmodeled dynamics or by uncertain parameters in the nominal model. Then, in order to guarantee robustness against all these sources, the reachable sets are used to compute the input/state space where the feasibility is ensured under the presence of the maximum disturbances considered in the design.

Remark 9.2. Considering large disturbances acting over the vehicle implies bounding the differences between the real and the nominal system in a large polytope which will lead to a more conservative scenario and also to the reduction of the maximum prediction horizon in the MPC design.

The inputs and states sets are updated at every control iteration and introduced as the new input/state constraints throughout the prediction window. They are computed as

$$\begin{aligned}\tilde{X}_{k+i} &= X \ominus \Phi_{k+i} \ , \ \forall i = 0, ..., H_p, \\ \tilde{U}_{k+i} &= U \ominus K^\infty_{\zeta_{k+i}} \Phi_{k+i} \ , \ \forall i = 0, ..., H_p - 1 \ .\end{aligned} \tag{9.31}$$

Note that, these sets cannot be empty sets since the optimal problem would not have solution.

Finally, the grouping of all the previous steps allow us to formulate the optimal problem as a quadratic optimization problem that is solved at each time k to determine the next sequence of control actions considering that the values of x_k and u_{k-1} are known

$$\begin{aligned}
\min_{\Delta U_k, X_k} \quad & \hat{x}_{k+H_p}^T P \hat{x}_{k+H_p} + \sum_{i=0}^{H_p-1} (r_{k+i} - \check{x}_{k+i})^T Q (r_{k+i} - \check{x}_{k+i}) \\
& + \Delta u_{k+i}^T R \Delta u_{k+i} \\
\text{s.t.} \quad & \\
& x_{k+i+1} = x_{k+i+1} + \big(A_{\zeta_{k+i}} x_{k+i} + B_{\zeta_{k+i}} u^*_{k+i}\big) T_s \\
& u^*_{k+i} = u^*_{k+i-1} + \Delta u_{k+i} \\
& u^*_{k+i} \in U \ominus (K^{\infty}_{\zeta_{k+i}} \Phi_{k+i}) \\
& \hat{x}_{k+i} \in X \ominus \Phi_{k+i} \\
& \hat{x}_k - x_k \in W \\
& x_{k+H_p} \in \chi_f \ominus \Phi_{k+Hp}
\end{aligned} \quad , \tag{9.32}$$

where $P \in \mathbb{R}^{s \times s} > 0$ represents the terminal cost computed in Section 9.4.2 and $Q = Q^T \in \mathbb{R}^{s \times s} \geq 0$ and $R = R^T \in \mathbb{R}^{n \times n} \geq 0$ are the tuning matrices for the states and the variation of the control inputs, respectively.

9.5 Results

In this section, we validate the performance of the proposed zonotope-tube-based LPV-MPC control scheme in a racing scenario through simulation in MATLAB. The principal objective of the presented scheme is to follow the proposed racing-based references ensuring asymptotic stability and the highest possible levels of robustness and performance while dealing with exogenous disturbances.

The racing references are provided by a trajectory planner and make the vehicle to perform close to its dynamic limits. The reference vector (r in Figure 9.3) is composed by two, the linear longitudinal speed and the angular velocity. Both are depicted using dashed lines in Figure 9.4. Note that, the linear speed reference belongs to a low velocity interval, i.e. between 10 and 25 km/h, however, it is the relationship between linear and angular velocities one of the measures used to determine the level of driving at the limit. Commonly, a high result in their product its an indicative of racing behaviour which consequently implies high lateral accelerations.

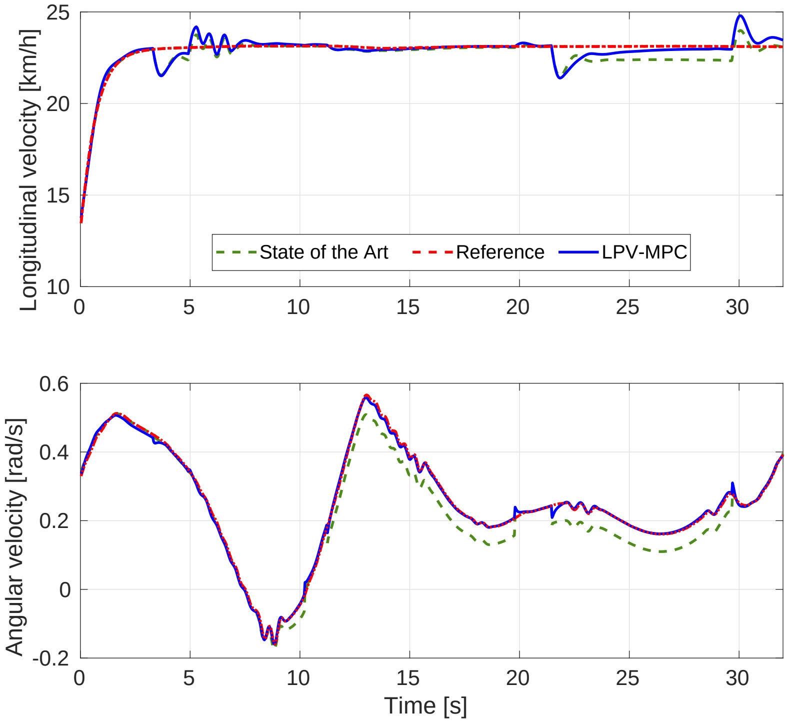


FIGURE 9.4: Dynamic reference tracking. Top: Longitudinal velocity reference and states (v_x) for both compared cases. Bottom: Angular velocity reference and states (ω)

To show the effectiveness of the robust control scheme, we compare the proposed strategy against a state of the art tube-based MPC scheme [Sakhdari, B., Shahrivar, E. M., & Azad, N. L., 2017] using the same scenario with several disturbance sources in simulation using the non-linear vehicle model. Such disturbance variables are chosen to be the road slope acting over the longitudinal vehicle dynamics and lateral wind affecting the lateral and angular vehicle dynamics (see Figure 9.5). The non-linear model used for simulation is a high-fidelity bicycle-based representation of the Driverless UPC vehicle [Driverless UPC., 2019] used in the Formula Student challenge [Formula Student., 2019] and is presented in Section 9.3.1. An identified tire model using the simplified Magic Formula [Pacejka, Hans., 2005] is used for generating accurate lateral forces from front and rear slip angles.

The LPV-MPC uses the predicted data in the past realisation to instantiate the state space matrices at every time step within the MPC prediction stage. To verify the real-time feasibility of the presented strategies, we perform the simulations on a DELL inspiron 15 (Intel core i7-8550U CPU @ 1.80GHzx8). Then, the optimal control problem

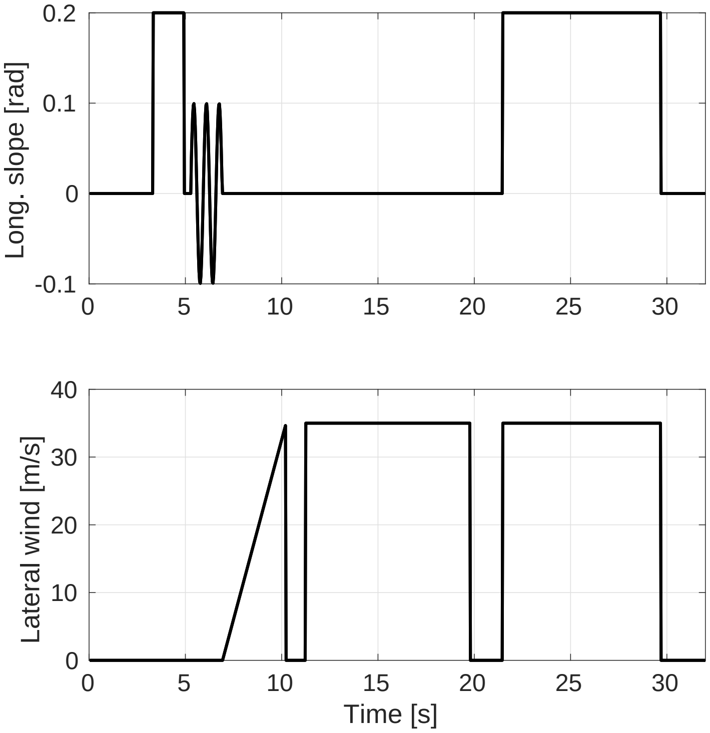


Figure 9.5: Disturbances acting on the scenario (d en Figure 9.3). Top: road slope profile composed by steps and sinusoidal parts. Bottom: Lateral wind velocity profile in the form of steps and a ramp

(9.32) is solved at a frequency of 30 Hz using the solver GUROBI [Optimization, Gurobi., 2014] through YALMIP [Lofberg, J., 2004] framework and the local controller is run at a higher frequency of 200 Hz. The tuning parameters for the robust LPV-MPC and LPV-H$_\infty$ problems are listed in Tables 9.3 and (9.33), respectively.

Table 9.3: Tube-based LPV-MPC design parameters. Q and R matrices are normalized by dividing the respective variable by its interval to the square ι^2

Parameter	Value	Parameter	Value
Q	0.8*diag($\frac{0.4}{\iota^2_{v_x}}$ 0 $\frac{0.6}{\iota^2_{\omega}}$ 0 0)	R	0.2*diag($\frac{0.4}{\iota^2_{\delta}}$ $\frac{0.4}{\iota^2_{a}}$)
$\overline{x}$	[15 1 1.4]	$\underline{x}$	[1 -1 -1.4]
$\overline{u}$	[0.267 13]	$\underline{u}$	[-0.267 -2]
$\overline{\Delta u}$	[0.05 0.5]	$\underline{\Delta u}$	[-0.05 -0.5]
T_s	30 ms	H_p	5
W	diag(0.0744 0.1895 0.1054 0 0)		

$$E = 0.3 \begin{bmatrix} 0 & 0.5 & 0 & 0 & 0 & 0 & 0 \\ 2 & 0 & 0 & 0 & 0 & 0 & 0 \\ 3 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0.0001 & 0 & 0 & 0 & 0 & 0 \\ 0.01 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \tag{9.33a}$$

$$C = 10^{-4} \begin{bmatrix} 0.2 & 0 & 0 & 0 & 0 \\ 0 & 0.2 & 0 & 0 & 0 \\ 0 & 0 & 0.2 & 0 & 0 \\ 0 & 0 & 0 & 0.2 & 0 \\ 0 & 0 & 0 & 0 & 0.1 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} \tag{9.33b}$$

$$D_1 = 10^{-4} \begin{bmatrix} 0 & 0 & -0.2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -0.2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -0.2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & -0.2 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -0.1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}, \; D_2 = 10^{-3} \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0.15 & 0 \\ 0 & 0.15 \end{bmatrix}. \tag{9.33c}$$

The reference tracking results are depicted in Figure 9.4. It can be seen the significant improvement of the presented scheme with respect to the tube-based MPC using LQR controller as the corrective error approach. Furthermore, the disturbance rejection has enhanced using a local controller whose design has been based on minimizing the infinity norm instead of the 2-norm as the case of LQR approach. However, note that using a H_∞ design may produce troubles in the closed-loop response because of the large gains that are obtained and hence, a meticulous tuning is needed.

In Figure 9.6, the errors or mismatch between the predicted state and the measured state are presented. Note that, such a vector of errors correspond with the vector entering the state feedback local controller (e in Figure 9.3). It can be appreciated the better

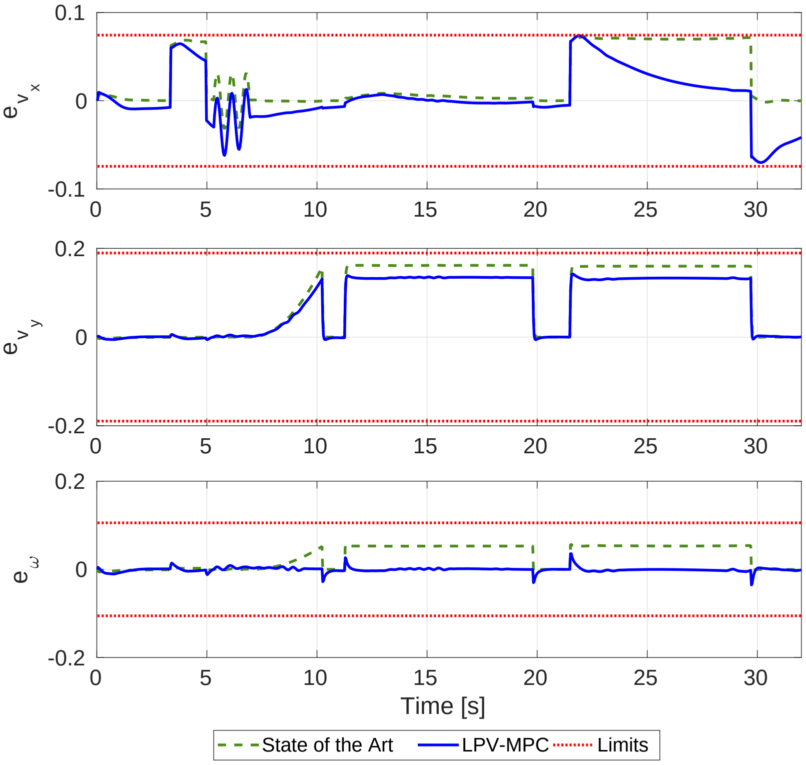


FIGURE 9.6: Mismatch between real and nominal states. e_{v_x} represents the error in the longitudinal behaviour, e_{v_y} the error in the lateral behaviour and e_ω represents the error for the angular behaviour. Dotted red lines represent the maximal bounds for each one of the errors defining then the set W

performance of the strategy presented in this work being able to reject most of the error produced by the uncertainty and the applied exogenous disturbances.

Figure 9.7 shows the control actions applied during the simulation test. Figure 9.8 shows the elapsed time per iteration of the complete tube-based LPV-MPC strategy with a prediction horizon of 6 steps. Hence, we prove the fast computation of this technique and the real-time implementation possibility on embedded systems.

Finally, a quantitative comparison is made using the normalized root mean squared error (NRMSE) as performance measurement (see Table 9.4). These results highlight the conclusive improvement of the proposed approach, improving up to thirty times the angular velocity tracking error with respect to the compared strategy.

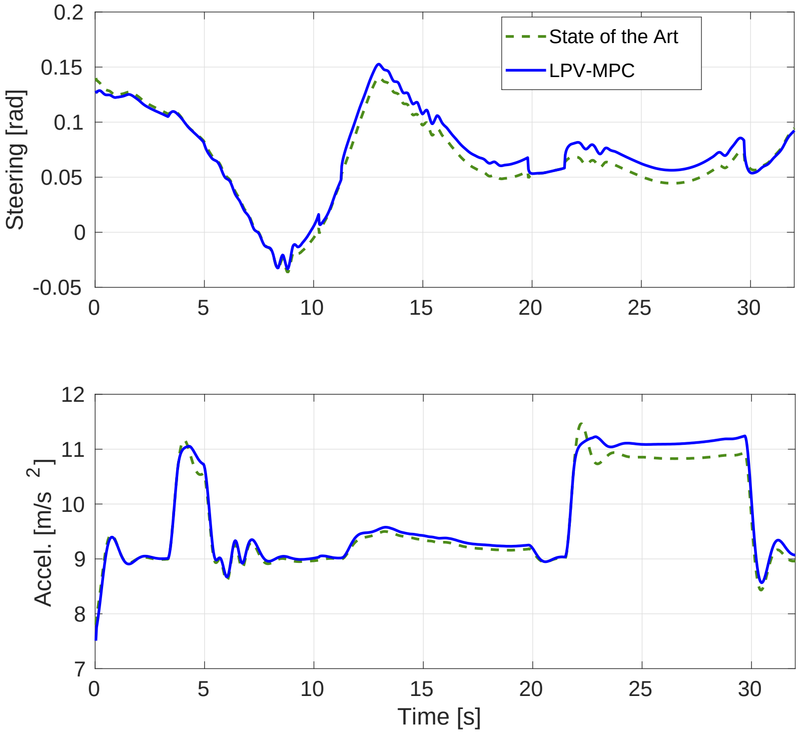


FIGURE 9.7: Control actions applied to the simulated vehicle (u in Figure 9.3)

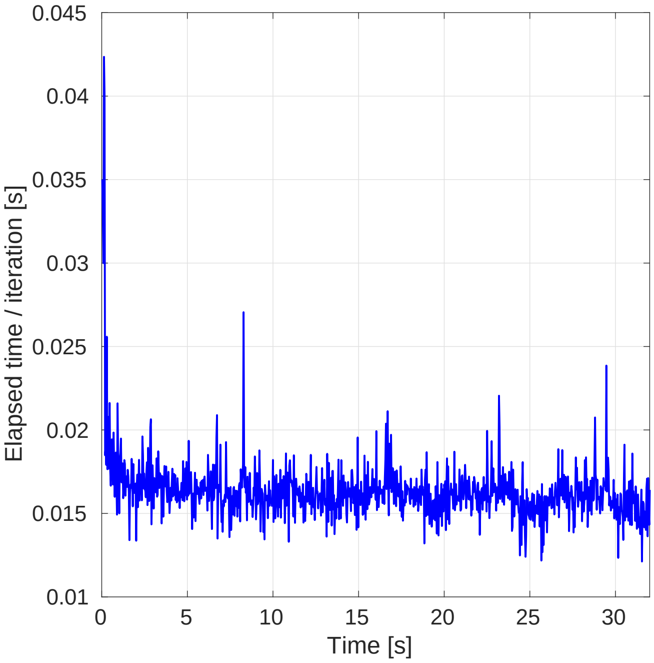


FIGURE 9.8: Elapsed time per iteration throughout the simulation. The mean time is 0.0164 s

Approach	RMSE v_x	RMSE ω
State of the art	$4.3846\ 10^{-4}$	0.0249
LPV-MPC	$3.4227\ 10^{-4}$	$8.0762\ 10^{-4}$

Table 9.4: Quantitative results for the tracking variables errors. These are the difference with respect to their respective reference (see Figure 9.4)

9.6 Conclusion

In this chapter, we have proposed a zonotope-tube-based LPV-MPC scheme for autonomous vehicles focusing on improving the computational load while preserving high levels of robustness and performance in racing scenarios.

While most of the strategies in the literature overcome the tube-based MPC problem using linearized models, this chapter aims to reformulate the non-linear equations to be expressed in an LPV form obtaining better results solving a quadratic optimization. In addition, the tube is computed using zonotopes theory which makes the calculation of reachable sets much faster since the propagation using the Minkowski basic set operations is much easier using zonotopes than polyhedron operations and less conservative since there are not approximations. Consequently, the prediction horizon can be increased compared to standard tube-based techniques since, due to the precision of calculation using zonotopes, the reachable set does not grow conservatively maintaining a more adjusted shape.

To reject the effect of acting disturbances maintaining robustness, a polytopic local controller has been designed solving the H_∞-based LMI problem. Furthermore, such a local controller is updated at a higher frequency than the nominal controller (LPV-MPC).

Finally, we have tested and compares the performance of the proposed strategy against a current state of the art tube-based MPC. We have shown the effectiveness of the presented approach in a disturbed racing scenario being able to perform online tube-based MPC with a high performance and reduced computational cost.

Part III

Trajectory planning

Chapter 10

LPV-MP Planning for Autonomous Racing considering Obstacles

The content of this chapter is based on the following works:

- [Alcalá, Eugenio, Puig, Vicenç and Quevedo, Joseba., 2019.B]. Alcalá, E., Puig, V. & Quevedo, J. LPV-MP Planning for Autonomous Racing Vehicles considering Obstacles. *Robotics and Autonomous Systems*, 2019.

- [Alcalá, Eugenio, et al., 2020.A]. Alcalá, E., Puig, V., Quevedo, J. & Rosolia, U. Autonomous Racing using Linear Parameter Varying - Model Predictive Control (LPV-MPC). *Control Engineering Practice*, 2019.

10.1 Introduction

The objective of racing planners is to find the optimal trajectory while maximizing the speed or minimizing the lap time. One of the main conditions of racing planning is to accurately consider the dynamics of the vehicle in the algorithm calculations in order to compute real feasible solutions. Then, determining a feasible solution for the autonomous racing planning is not a trivial task at all and there are few studies under this topic [Caporale, Danio, et al., 2018, Alrifaee, B., et al., 2018, Verschueren, Robin, et al., 2014, 2016, Liniger, A., et al., 2015]. The weaknesses that these works have in common are the use of models of reduced complexity (kinematic models basically) and/or the use of a path-based planner formulation.

In this chapter, we present a novel approach to solve the optimal trajectory planning for autonomous racing vehicles considering static obstacles throughout the track. Unlike the works mentioned above, a dynamic vehicle model is used for computing more realistic trajectories. In order to deal with track constraints, we express the road limits as the maximum values, positive and negative, of the allowed vehicle lateral error. In addition, such a way of limiting the road allows a straightforward extension to avoid obstacles. The way we propose to overcome obstacles is to modify the limit of the track at each instant of time. In this way, the limit values of the lateral error interval in the optimal problem vary as a function of the obstacle and vehicle positions. The key idea of this approach is to use a Optimal Quadratic Programming algorithm that maximizes the velocity vector within a certain horizon. This optimization-based algorithm uses an LPV model representation of the vehicle to compute the trajectory in a given prediction window.

The scheduling variables of the LPV system are obtained by taking the shifted result of the previous optimal iteration. Simulations are carried out using the Driverless UPC Car. This development platform for autonomous racing was presented in Section 9.3. The computational time of this approach is reduced drastically in comparison with its non-linear version making it very suitable for real-time implementations. Finally, the resulting trajectory is provided to the controller as the current trajectory to be followed and the process is repeated at the next sampling time.

10.2 NL-MPC as a planner in space domain

The main function of this racing planning task is to find a trajectory within the circuit that minimizes the total lap time and provides relevant information to the motion control.

Then, using the space-based vehicle model presented in Section 3.4.3 the following nonlinear optimization problem is solved using a constant sampling space (ds)

$$\begin{aligned}
&\underset{\Delta\mathcal{U},\tilde{\mathcal{X}}}{\text{minimize}} \quad \mathrm{J} = \sum_{i=0}^{H_p-1}\left(\tilde{x}_i^T Q \tilde{x}_i + \Delta u_i R \Delta u_i\right) + \tilde{x}_{H_p}^T P \tilde{x}_{H_p} \\
&\text{s. t.} \\
&\qquad \tilde{x}_{i+1} = \tilde{x}_i + f(\tilde{x}_i, u_i, \kappa_i) ds, \quad \forall i = 0, ..., H_p - 1 \\
&\qquad u_i = u_{i-1} + \Delta u_i \\
&\qquad H_p = L_{track}/ds \\
&\qquad \tilde{y}_e \in [\overline{\tilde{y}_e}, \underline{\tilde{y}_e}] \\
&\qquad \left(\frac{\tilde{v}_{x_{i+1}} - \tilde{v}_{x_i}}{\overline{\Delta v_x}}\right)^2 + \left(\frac{\tilde{v}_{y_{i+1}} - \tilde{v}_{y_i}}{\overline{\Delta v_y}}\right)^2 - 1 \leq 0 \ ,
\end{aligned} \tag{10.1}$$

where the decision vector variables are

$$\begin{aligned}
\tilde{\mathcal{X}} &= (\tilde{x}_1, \tilde{x}_2, ..., \tilde{x}_{H_p}) \\
\Delta\mathcal{U} &= (\Delta u_1, \Delta u_2, ..., \Delta u_{H_p}) \ .
\end{aligned} \tag{10.2}$$

Note that, with the aim of smoothing the performance effort and providing a null tracking error in steady state, the variation of the input model variables (Δu_i) is used as an optimization variable. State vector, control input vector and the continuous-time nonlinear model are properly defined in Section 3.4.3. L_{track} is the total length of the circuit, $Q = Q^T \in \mathbb{R}^{s \times s} \geq 0$, $R = R^T \in \mathbb{R}^{n \times n} \geq 0$ and $P = P^T \in \mathbb{R}^{s \times s} \geq 0$ are the weighting matrices. The lateral error ($\tilde{y}_e$) is limited in the interval defined by the width of the track. The last inequality constraint in (10.1) bounds the longitudinal and lateral vehicle acceleration. This constraint defines an ellipse limited by $\overline{\Delta v_x}$ and $\overline{\Delta v_y}$ which are found experimentally.

The adjustment in the cost function is a choice of the designer, however, certain guidelines must be followed to achieve the objective of minimizing the total lap-time. Furthermore, in order to achieve numerical reliability, very small weights are used for the rest of variables in the cost function. The diagonal values of the weighting matrices, found to obtain the best trade-off among the different objectives, are

$$\begin{aligned} Q &= [\ 10^{-8} \quad 10^{-8} \quad 10^{-3} \quad 10^{-5} \quad 10^{-8} \quad 10^{-8}\ ] \\ R &= [\ 0.05 \quad 0.01\ ] \\ P &= [\ 10^{-8} \quad 10^{-8} \quad 10^{-8} \quad 10^{-8} \quad 10^{-8} \quad 1\ ] \,. \end{aligned} \tag{10.3}$$

Note that using this approach just one optimization is solved. The optimal solution is the vehicle state vector for a given L_{track} distance.

Finally, before providing the obtained trajectory references to the motion control strategy, a time-based interpolation is made in order to convert the space-based variables into time-based variables. Before applying the linear interpolation, we have the space-based state vector ($\tilde{x}$) obtained from the optimization problem 10.1 and among them the time depending on the space ($\tilde{t}$). Now, we define the query time points as a finer sampling in the range of $\tilde{t}$. Then, the next interpolation is performed at those query points (t_i) to obtain the vehicle state variables as a function of time (x_i)

$$x_i = \tilde{x}_i + \frac{\tilde{x}_{i+1}(t_i - \tilde{t}_i) - \tilde{x}_i(t_i - \tilde{t}_i)}{\tilde{t}_{i+1} - \tilde{t}_i} \ , \ \forall i = 0, ..., H_p \ , \tag{10.4}$$

where x_i is the new time-dependent variable and t_i is the accumulative sampling time. Note that, this interpolation procedure is carried out for variables $\tilde{v_x}$, $\tilde{v_y}$, $\tilde{\omega}$, $\tilde{y_e}$ and $\tilde{\theta_e}$, but not for $\tilde{t_i}$.

10.3 NL-MPC as a planner in time domain (NL-MPP)

In this case, the time is not a state variable anymore and therefore a different way of minimizing time is looked for. The maximization of the vehicle speed vector is equivalent to minimizing the time variable. As presented in Section 3.2.2, the velocity vector of the vehicle in curvature-based coordinates ($\dot{s}$) can be computed by

$$V = \frac{v_x \cos\theta_e - v_y \sin\theta_e}{1 - y_e \kappa} \tag{10.5}$$

as a relation between the track curvature and the dynamic and error vehicle states.

Limiting accelerations in the CG of the vehicle is one of the best ways to limit the dynamics of the vehicle (last non-linear constraint in (10.1)). However, these limits are directly related to the adhesion of the tires and therefore to the tire force model. That is why, the limits for the front and rear slip angles as well as minimizing the difference between them in the cost function are introduced in the formulation of this section. The objective is to avoid high levels of understeer or oversteer behaviors. These depend on the front and rear wheels slip angles and are given as

- The understeer situation is given when the front wheels slip angle is greater than the rear wheels slip angle : $\alpha_f > \alpha_r$.
- The oversteer situation is given when the rear wheels slip angle is greater than the front wheels slip angle : $\alpha_r > \alpha_f$.

These two situations should be avoided for a smooth performance of the vehicle. However, in racing environments it is allowed to have particular levels of understeer and oversteer.

Hence, both objectives can properly be formulated as a weighted non-linear cost function where it is pretended

- to maximize the linear velocity vector which implies to minimize the travel time

$$maximize: \ J_V(x_{k+i}, \kappa_{k+i}) = \sum_{i=0}^{H_p} \|V_{k+i}\|_Q^2 \tag{10.6}$$

- to minimize the difference between the front and rear slip angles with the aim of avoiding high levels of understeer or oversteer behaviours

$$minimize: \ J_\alpha(x_{k+i}) = \sum_{i=0}^{H_p} \|\alpha_{f_{k+i}} - \alpha_{r_{k+i}}\|_R^2 \tag{10.7}$$

where $Q \in \mathbb{R}^{1x1} \geq 0$, $R \in \mathbb{R}^{1x1} \geq 0$ are the proper weighting scalars.

Additionally, a third component added to the cost function aims to minimize the slack variable σ introduced over the lateral error state to provide some flexibility to the optimization problem

$$minimize: \ J_\sigma(x_{k+i}) = \sum_{i=0}^{H_p} \|\sigma_{k+i}\|_P^2 \tag{10.8}$$

where $P \in \mathbb{R}^{1x1} \geq 0$. Finally, combining previous objectives, a constrained non-linear optimal control (CNLOC) problem is formulated as follows and solved at each discrete time step k to compute an optimal trajectory

$$\begin{aligned} &\min_{\Delta U_k, X_k} \Big(-J_V(x_{k+i}, \kappa_{k+i}) + J_\alpha(x_{k+i}) + J_\sigma(x_{k+i}) \Big) \\ &\text{s.t.} \\ &\qquad x_{k+i+1} = x_{k+i} + \big(f(x_{k+i}, u_{k+i}, \kappa_{k+i})\big) T_s \ , \ \forall i = 0, ..., H_p \\ &\qquad u_{k+i} = u_{k+i-1} + \Delta u_{k+i} \\ &\qquad u_{k+i} \in [\overline{u}, \underline{u}] \\ &\qquad y_{e_{k+i}} \in [\overline{y_e} + \sigma_{k+i}, \underline{y_e} - \sigma_{k+i}] \\ &\qquad \alpha^f_{r_{k+i}} \in [\overline{\alpha^f_r}, \underline{\alpha^f_r}] \end{aligned} \tag{10.9}$$

where Δu_{k+i}, also known as slew rate, represents the time variation of u_{k+i} and is used to add an integral action to the system. ΔU_k and X_k represent the optimal control input and state variables sequences, respectively. However, note that, although this CNLOC problem is able to provide a solution to the planning problem there exists a high computational load when solving it and, hence, becoming a not implementable solution for online planning problem in real embedded systems.

10.4 Space-domain vs time-domain planning

Both spaces are appropriate depending on the application, being sometimes the use of one more advantageous than another. From the point of view of a continuous model, this will be discretized using a constant sample space in the case of space-domain or a constant sample time in the case of time-domain.

On the one hand, in the domain of space, the vehicle model will evolve at constant steps of space as if it were on a grid where the size of the advanced space is given by the size of the grid. This representation is very useful when our system is for some reason based on space. Note that, the curvature-based model presented in 3.2.2 contains the curvature variable (κ) which is a function of the distance traveled (s) on the track. In this way, when we make a prediction in the MPC strategy we can know perfectly the curvature since we know what we are going to advance in that iteration (constant space).

However, this space-based formulation of the model is complex and therefore its LPV representation is not trivial.

On the other hand, the planning using the vehicle model expressed in the time domain loses precision in the solution since the curvature of the track is calculated with the variables obtained in the previous optimization which may not be global optimal. However, non-linear expressions of the vehicle model in time-domain are more suitable for LPV reformulation. Table 10.1 presents the main advantages and disadvantages.

Table 10.1: Advantages and disadvantages

	NL-MPP space-domain	NL-MPP time-domain
Advantages	curvature acts as a reference since is a space function	loss of accuracy: curvature is calculated with previous optimization variables
Disadvantages	complex LPV formulation	reachable LPV formulation

10.5 LPV-MPP formulation

In this section, we present a novel formulation for the optimal problem in time-domain (10.9) using the LPV representation of the non-linear vehicle model. The key idea of this approach relies on the use of an LPV-based modeling which provides the ability to simulate the vehicle dynamics with a low computational cost. This imply that, at every discrete time k, a set of H_p instantiations of the LPV representation (4.14) are used to compute the online trajectory. In addition, in order to formulate the optimization problem in the same form as (10.9), the non-linear speed function (10.5) has to be represented in a quadratic form to handle the problem as a constrained linear quadratic optimal control (CLQOC) problem.

10.5.1 Convexifying the objective function

At this point, the cost function of the CNLOC problem (10.9) consists on a quadratic term (J_α) and a non-linear part (J_V). In this section, the methodology for convexifying the non-linear term J_V is addressed. The objective is to find out a linear-quadratic formulation that approximates the non-linear equation of the vehicle velocity

$$V = \frac{v_x \cos\theta_e - v_y \sin\theta_e}{1 - y_e \kappa} \ , \tag{10.10}$$

such as

$$V \approx \mathrm{V}^{\mathrm{QP}} = s^T Q s + q^T s \ . \tag{10.11}$$

Since the original objective function in the optimization problem is to maximize (11), then we look for a concave formulation such that it can be introduced in the CLQOC problem as the minimization of its convex version. Studying the problem, we find that approximating the original function is a difficult task and only making some assumptions we can find a suitable result. In particular, on one hand, we observe that there is not a strong relationship with variable κ, such that $s := \begin{bmatrix} v_x & v_y & \theta_e & y_e \end{bmatrix}^T$. On the other hand, we consider the use of a diagonal Q matrix which simplifies the approximation of (10.10) using (10.11).

Least-squares techniques for fitting polynomials are limited for this purpose to provide a quadratic model, with constant and quadratic terms but avoiding the linear term which may not fit the objective.

To solve this problem, we propose the following constrained linear quadratic optimal problem

$$\begin{aligned} &\min_{Q,q} \quad \sum_{i=0}^{M} \| V_i - V_i^{QP} \|^2 \\ &\text{s.t.} \\ &\qquad V_i^{QP} = s_i^T Q s_i + q^T s_i \\ &\qquad diag(Q) < 0 \end{aligned} \tag{10.12}$$

where $Q = Q^T \in \mathbb{R}^{4\times4} < 0$ and $q \in \mathbb{R}^4$ are the optimization variables and M denotes the length of the optimization problem. Note that due to the last constraint in (10.12), the resulting V^{QP} is defined as a strictly concave function.

Finally, after formulating the LPV model and the convex objective function, we present the following CLQOC problem that is solved at each time k to determine the next sequence of states considering that the values of x_k and u_{k-1} are known

$$\min_{\Delta U_k, X_k} \quad \Big(-\sum_{i=0}^{H_p} V^{QP}(x_{k+i}) + \sum_{i=0}^{H_p} \|\alpha_{f_{k+i}} - \alpha_{r_{k+i}}\|_R^2 + \sum_{i=0}^{H_p} \|\sigma_{k+i}\|_P^2 \Big)$$

s.t.

$$\begin{aligned} & x_{k+i+1} = x_{k+i} + \big(A(\zeta_{k+i})x_{k+i} + B(\zeta_{k+i})u_{k+i}\big)T_s \ , \ \forall i = 0, ..., H_p \\ & u_{k+i} = u_{k+i-1} + \Delta u_{k+i} \\ & u_{k+i} \in [\overline{u}, \underline{u}] \\ & y_{e_{k+i}} \in [\overline{y_e}_k + \sigma_{k+i}, \underline{y_e}_k - \sigma_{k+i}] \\ & \alpha^f_{r_{k+i}} \in [\overline{\alpha^f_r}, \underline{\alpha^f_r}] \end{aligned} \quad , \tag{10.13}$$

where $R \in \mathbb{R}^{1x1} \geq 0$ and $P \in \mathbb{R}^{1x1} \geq 0$ are weighting scalars. Note that, the third component in the objective function aims to minimize the slack variable σ introduced over the lateral error state. This choice is made to provide the optimal problem some flexibility.

10.6 Introducing Static Obstacles

This section addresses the static obstacle avoidance problem during the planning task. The procedure is mainly based on two steps. First, the computation of a safety polytope that contains the obstacle is done based on the information provided by the perception layer (not presented in this chapter). This polytope is chosen to be a rectangle in this chapter. Second, the computation of the new lateral bounds of the road taking into account the obstacle polytope is addressed.

The proper detection and position computation of the particular obstacle are simulated as if they were done by a higher perception layer using stereo-based cameras. This hardware provide a cloud of points on the obstacle's edge with their respective RGB data and distance to the camera. Then, projecting these cloud points to the hyperplane h^p, we are able to compute the frontal face of the polytope containing the obstacle (see Figure 10.1). Note that, such a hyperplane is always orthogonal to the road orientation. At this point, the planning stage computes the polytope that contains the obstacle. First, the hyperplane h^p is extended using a safety distance y^s. Then, the rectangle is closed by using the vehicle diagonal length, denoted by d. Note that, the obstacle is assumed

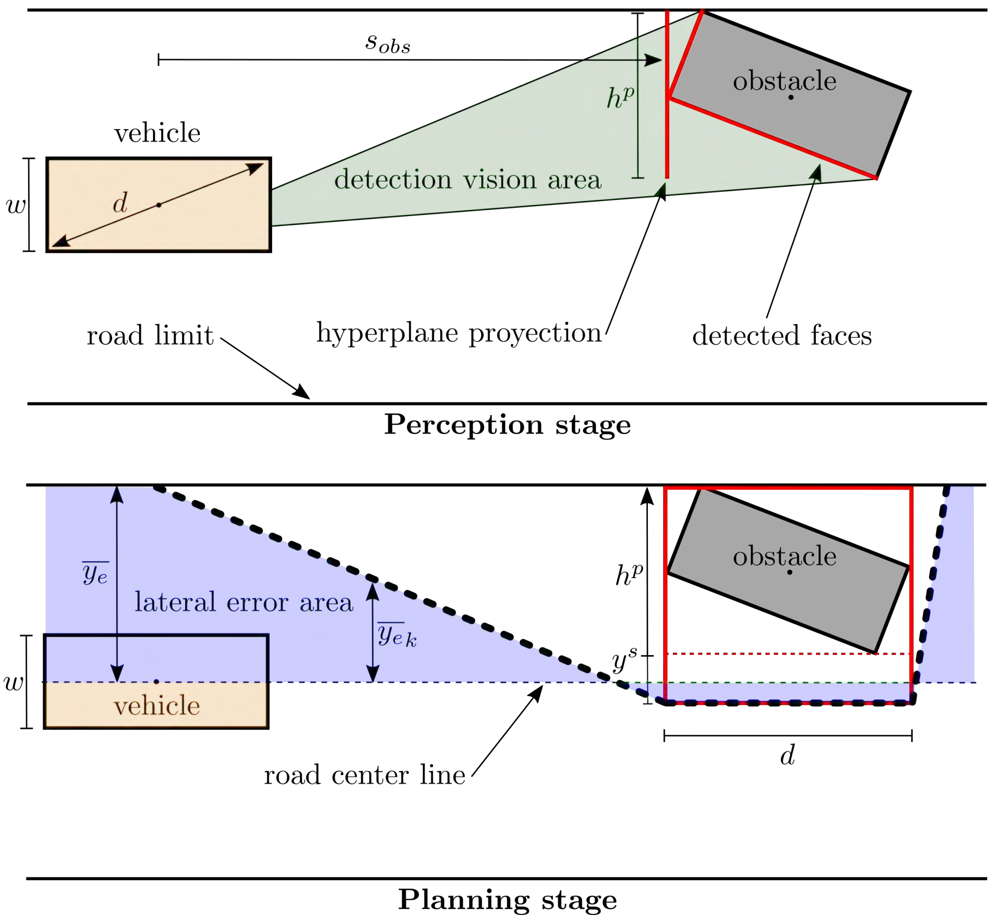


Figure 10.1: The above figure depicts how the perception layer detects the vehicle using stereo-based cameras. Below, the planning layer computes the lateral error area based on the obstacle set (red box)

to have the same width and diagonal length than the UPC Driverless vehicle (see Table 9.1). In addition, the election of y^s is made taking into account the half width of the vehicle plus an extra distance for safety reasons. Once the polytope is obtained, the new lateral bounds can be computed. To do so, an incremental variation of the limit of the lateral error ($\overline{y_e}_k$) is determined using Algorithm 1 computing then the lateral error vector as

$$\overline{Y_e} = [\overline{y_e}_0, ..., \overline{y_e}_{H_p}] \ . \tag{10.14}$$

This vector is computed at every discrete time k and introduced as an input to the CLQOC problem. Note that this approach may be conservative but very efficient computationally.

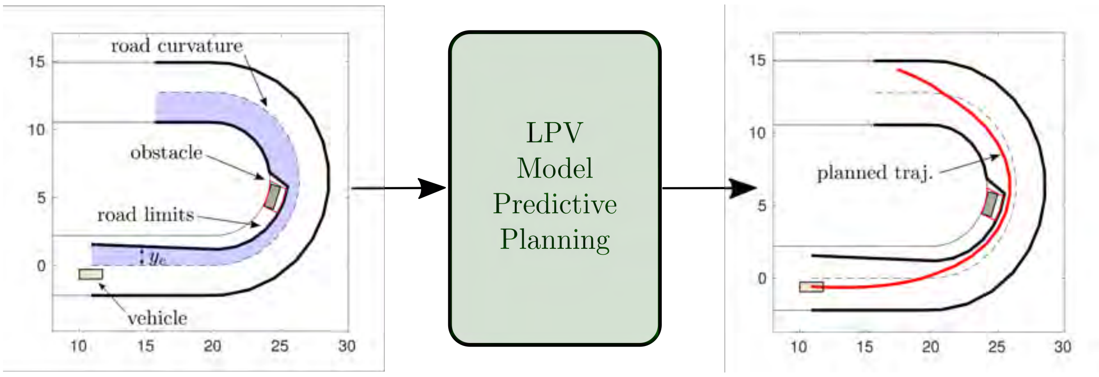


FIGURE 10.2: Simplified view of the planning strategy. The left part represents the set of inputs, i.e. vehicle position, road limits and track curvature. The right side shows the planned trajectory for a particular discrete time k

```
Input  : $s_{obs}$, $h^p$, last predicted vehicle states (4.14b)
Output: Free road limits ($\overline{Y_e}$)
integrate (10.10) to obtain $\hat{s}$ using the last predicted states and the reference $\kappa$
given the predicted vector $\hat{s}$, obtain the vector index ($i_{obs}$) at which $\hat{s} \approx s_{obs}$
compute $\Delta y_e = \frac{h^p + y^s}{H_p}$.
for $i = 1;\ i < H_p + 1;\ i = i + 1$ do
    $\sum_{Ey}$ += $\Delta y_e$;
    if $\hat{s}(i_{obs} - i) > 0$ then
        $\overline{Y_e}(i) = \sum_{Ey}$
    end
end
```

Algorithm 1: Road limits algorithm (see Figure 10.1)

10.7 Results

In this section, the performance of the proposed racing LPV-MPP approach (10.13) is evaluated. To do so, a comparison against its non-linear version (10.9) is presented as well as the performance in different scenarios. However, before entering into the details, it is important to emphasize that in racing scenarios it is necessary to have a minimum of knowledge about the evolution of the track. Thus, in this chapter, we consider the curvature of the track as a known variable to perform the racing trajectory planning. The details of the experimental set up and simulated scenarios are presented in the following.

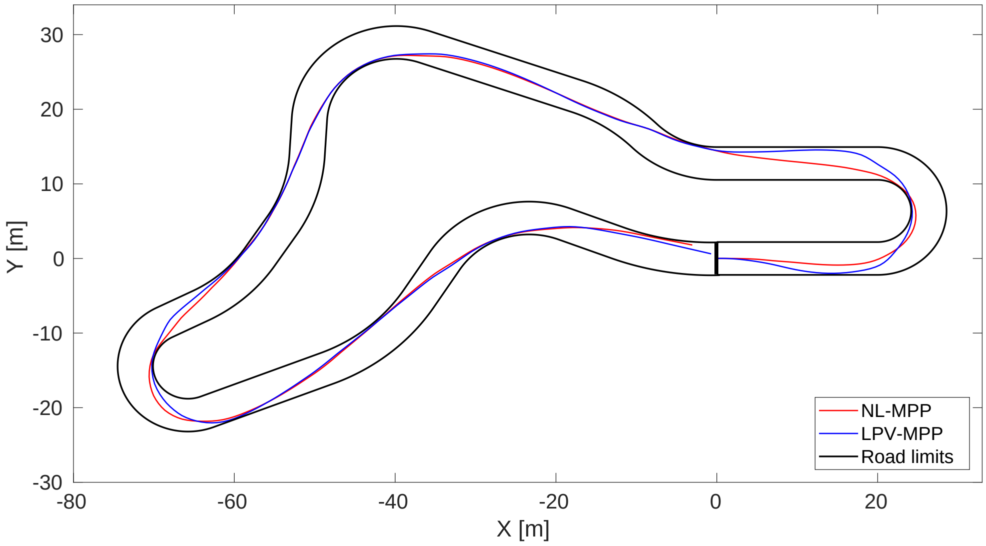


FIGURE 10.3: Resulting two laps trajectories for the proposed LPV approach and the NL approach on the free obstacle scenario

10.7.1 Simulation/Experimental Set Up

For evaluating the proposed architecture, we perform simulations using the UPC Driverless vehicle model which is described in Section 9.3 with parameters defined in Table 9.1. Note that, at the first CLQOC iteration the time evolution of the scheduling vector

```
Input  : Current vehicle states ($x_k$), past control input ($u_{k-1}$), Free road limits ($\overline{Y_e}$),
         Track curvature ($\kappa$)
Output: Vehicle states defining the predicted trajectory ($x_{k+i}$) , $\forall i = 1, ..., H_p$

if $k = 1$ then
    initialize the evolution of scheduling vector $\zeta$
else
    instantiate the scheduling vector $\zeta$ using previous predicted data
end
compute the set of $H_p$ LPV instantiations (4.14), i.e. $A_{k+i}$ and $B_{k+i}$ , $\forall i = 0, ..., H_p$
$x_{k+i} \leftarrow$ CLQOC($A(\zeta)$, $B(\zeta)$, $x_k$, $u_{k-1}$, $\overline{Y_e}$, $\kappa$) , $\forall i = 1, ..., H_p$
go to step 1

Function CLQOC($A(\zeta)$, $B(\zeta)$, $x_k$, $u_{k-1}$, $\overline{Y_e}$, $\kappa$):
    solve QP (10.13) using GUROBI, $x_{k+i}$ = CLQOC( $A(\zeta)$, $B(\zeta)$, $x_k$, $u_{k-1}$, $\overline{Y_e}$, $\kappa$ )
    interpolate the solution at the control sampling time
    return
```

Algorithm 2: LPV-MP Planning algorithm (see Figure 10.2)

is not known. At this point, we solve this problem by generating such evolution based on previous knowledge on how the states of the system evolve. Once this initialization is done, the predictions are used to instantiate the set of next H_p models of the LPV model. The sampling time used is $T_s = 300$ ms and such a prediction horizon H_p is set to 15 steps, this implies 4.5 s of future behaviour prediction, i.e. the trajectory.

Matrices $A(\zeta)$ and $B(\zeta)$ in (10.13) are instantiated online before the optimization starts which implies a set of H_p LPV model instantiations entering the optimal problem. Note that, tire stiffness coefficients in (4.14e), i.e. C_f and C_r, are also properly instantiated online using (4.7) and Table 9.1 as a function of α_f and α_r. The model used in the CLQOC problem is dependent on the curvature. This means that it is required to know the curvature of the circuit at every moment to instantiate the vehicle LPV model. In addition, the limits of the navigable space, i.e. space to which obstacles do not belong, are required by the optimization problem. These limits will vary depending on possible static obstacles throughout the track.

The solutions of problem (10.12) are obtained considering

$$\begin{aligned} Q &= diag \begin{bmatrix} -1.2\times 10^{-4} & -9.704 & 3.5\times 10^{-5} & -0.154 \end{bmatrix} \\ q &= \begin{bmatrix} 1.007 & 0.187 & 6.1\times 10^{-7} & -0.032 \end{bmatrix} . \end{aligned} \tag{10.15}$$

Note that, since the vector state in (10.13) is the one defined in (4.14b), the obtained matrices Q and q are restructured properly to be

$$\begin{aligned} Q &= diag \begin{bmatrix} -1.2\times 10^{-4} & -9.704 & 0 & 3.5\times 10^{-5} & -0.154 & 0 & 0 \end{bmatrix} \\ q &= \begin{bmatrix} 1.007 & 0.187 & 0 & 6.1\times 10^{-7} & -0.032 & 0 & 0 \end{bmatrix} . \end{aligned} \tag{10.16}$$

The control inputs bounds are set to $\overline{\delta} = -\underline{\delta} = 0.3$ rad and $\overline{a} = -\underline{a} = 12\frac{m}{s^2}$. Front and rear wheels slip angles are limited to $\overline{\alpha_r^f} = -\underline{\alpha_r^f} = 0.16$ rad. Lateral error limits are set in next sections since they will vary with the obstacles.

Both the LPV-MPP and NL algorithms are implemented in Matlab R2017a on a Dell Inspiron 15 5000 Series using a Intel core i7-8550U CPU @ 1.80GHzx8. For solving the non-linear optimization problem (5) used as baseline solution, IPOPT solver [Wächter, A, et al., 2006] is used while for solving the QP problems GUROBI solver [Optimization, Gurobi., 2014] is used, both through YALMIP [Lofberg, J., 2004] framework.

The pseudo-code for the implementation of LPV-MP Planning is shown in Algorithm 2.

10.7.2 Free Track Racing Planning

In this experiment, we compare the LPV-MPP strategy against its time-domain non-linear version (see Section 10.3) in a free obstacles track. Algorithm 2 is used to obtain the LPV-MPP results. To solve the non-linear version for comparison, we solve problem (10.9).

The simulated trajectories are depicted in Figure 10.3 for one lap. It can be seen how the non-linear version performs a bit smoother than the proposed LPV approach, even though at a higher computational cost, around 50 times slower (see Figure 10.5). The simulated track has a width of 4 m, hence, the lateral error bounds in (10.9) and (10.13) are set to be symmetric such that $\overline{y_e} = -\underline{y_e} = 2$ m.

The velocities, slip angles and lateral errors for the whole simulation are displayed in Figure 10.4. It can be seen that LPV-MPP delivers a similar solution to the NL-MPP one for the compared vehicle states. The lateral error with respect to the center line of the road is the most important state for solving this problem. We allow the vehicle to have up to 2 m of lateral error in order to find the best path. By looking into the iterations interval [60-80], it can be appreciated that the LPV-MPP y_e achieves higher values implying then the vehicle approaches more to the road limits in comparison to the NL-MPP response. However, this is still a good solution for the racing trajectory planning that allows the real time implementation.

In Table 10.2, a comparison is made in terms of mean values. The great difference is not in the solution but in the elapsed time at each iteration what makes the LPV-MPP strategy a much faster approach and therefore a suitable option to be implemented on-line in real-time on embedded systems.

Table 10.2: Mean values of longitudinal velocity and acceleration, elapsed time and slip angles difference

	$\overline{v_x}$	$\overline{a_x}$	$\overline{t_e}$	$\overline{\alpha_r - \alpha_r}$
NL-MPP	50.1	0.615	2.961	-4.8 10^{-4}
LPV-MPP	50.6	0.616	0.057	-6.5 10^{-4}

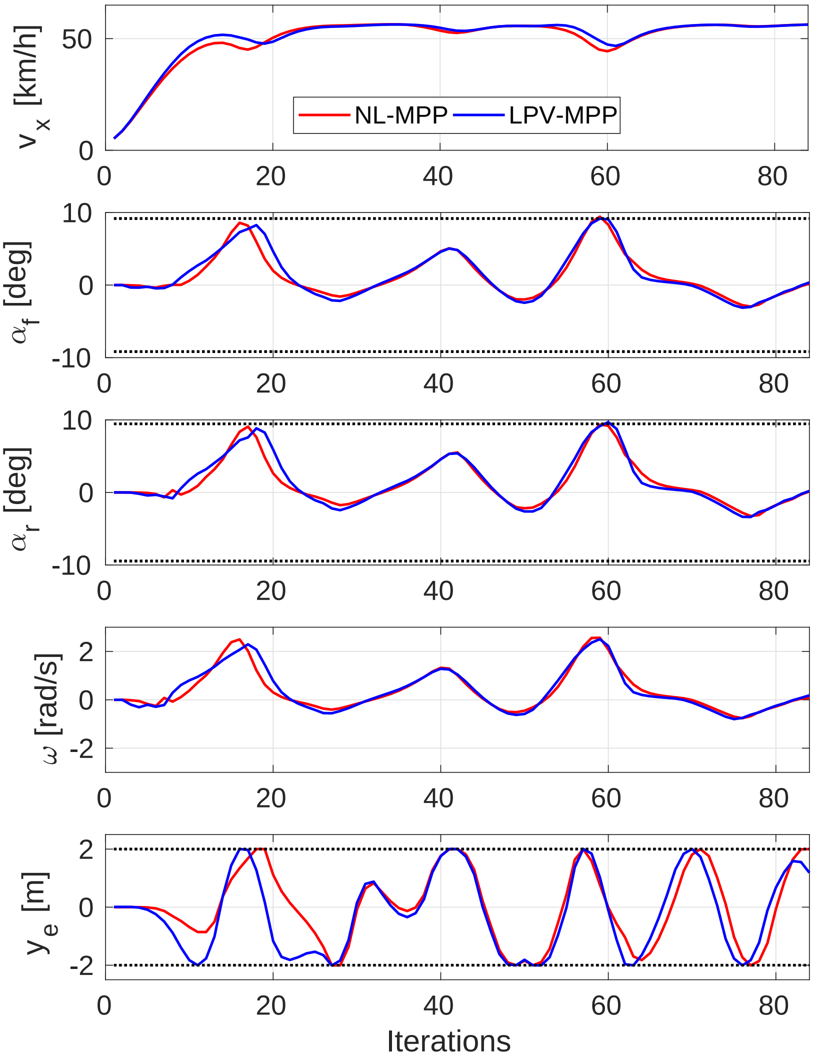


Figure 10.4: Velocities, slip angles and lateral error throughout the simulation: dotted lines represent the state limits considered in the optimal problem

10.7.3 Static Obstacles Racing Planning

In this section, the validation is made considering static obstacles within the track. Three obstacles are introduced in strategic points of the circuit. The objective of the planner is to maximize the lap velocity while avoiding the three obstacles introduced along the circuit.

The obstacle detection is assumed to be done outside this planning procedure as it was explained in Section 10.6. Then, as it is depicted in Figure 10.2, the new limits of the road are updated at every iteration taking into account the obstacle. Therefore, these limits

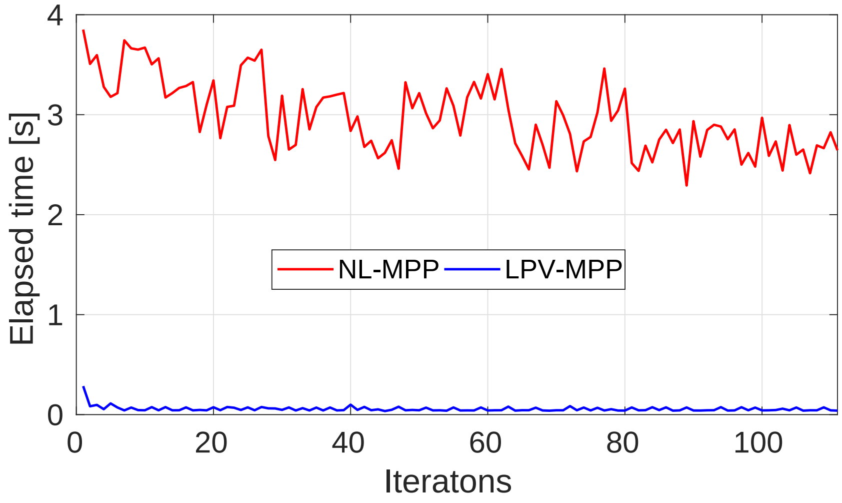


FIGURE 10.5: Computational time cost for both compared strategies: NL-MPP mean elapsed time is 2.961 s and LPV-MPP mean elapsed time is 0.0567 s

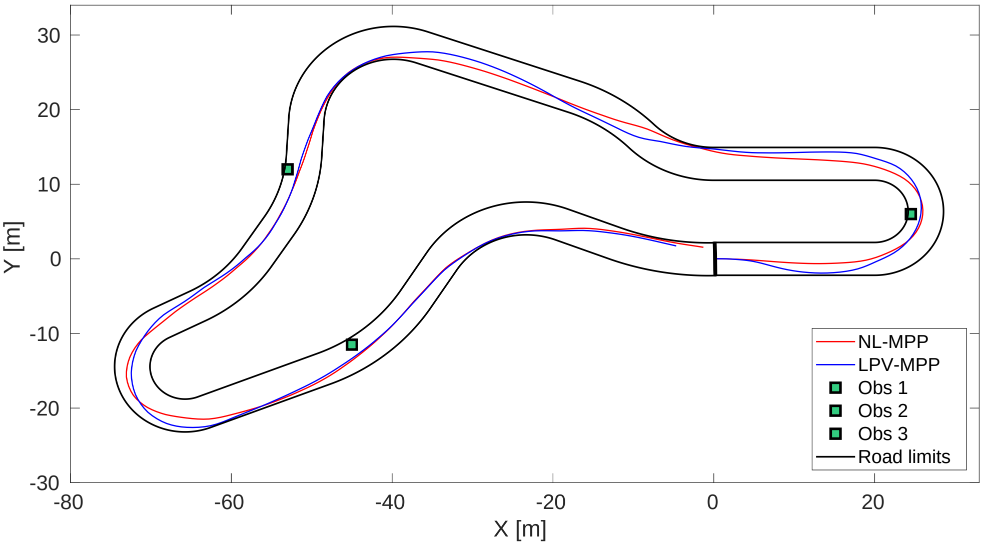


FIGURE 10.6: Resulting one lap trajectories for the proposed LPV approach and the NL approach on the obstacle scenario

are introduced as a new bounds for lateral error state $(\overline{y_e}, \underline{y_e})$ in the optimal problem (10.13).

As in Section 10.7.B, we compare the performance of the NL-MPP and LPV-MPP planning approaches. The planned trajectories are shown in Figure 10.6. At a first glance, it can be seen that the NL-MPP approach provides a smoother trajectory. The velocities, slip angles and lateral errors for the whole simulation are presented in Figure 10.7.

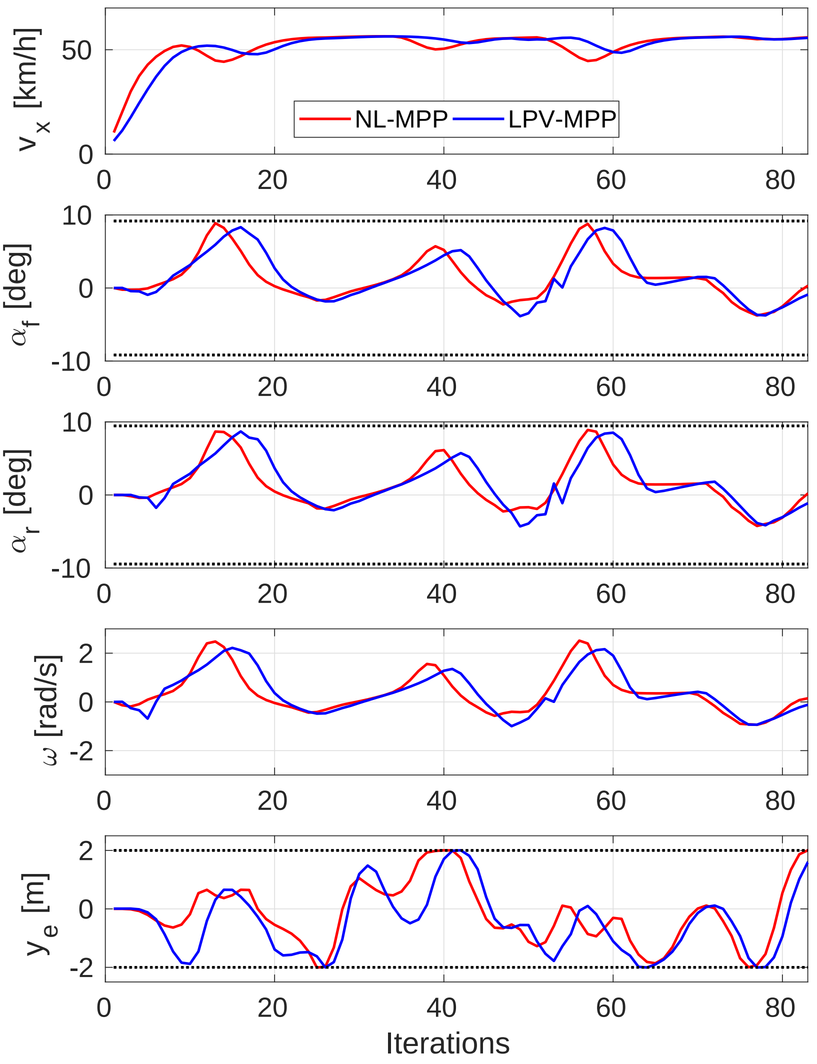


FIGURE 10.7: Velocities, slip angles and lateral error throughout the simulation: dotted lines represent the state limits considered in the optimal problem

In this graphical comparison, it is observed a greater difference between their responses. Both slip angles, which are important variables in racing, remain inside the allowed region. However, the LPV-MPP approach perform sharper solutions.

Table 10.3 illustrates the resulting comparison in terms of mean values. It can be observed that the non-linear approach is able to minimize more the difference between slip angles reducing then over and understeering behaviours. However, the LPV-MPP performs an acceptable solution and its computational time is much lower. To conclude this section,

TABLE 10.3: Mean values of longitudinal velocity and acceleration, elapsed time and slip angles difference

	$\overline{v_x}$	$\overline{a_x}$	$\overline{t_e}$	$\overline{\alpha_r - \alpha_r}$
NL-MPP	49.9	0.613	3.032	-6.4 10^{-4}
LPV-MPP	50.6	0.607	0.062	-8.7 10^{-4}

we measure the elapsed time of both performances. However, we do not observe large differences comparing to the case without obstacles so we refer to the same Figure 10.5.

10.8 Conclusion

In this chapter, an innovative solution for the online trajectory planning problem considering static obstacles and focused on racing behaviours has been presented. First, the space-domain and the time-domain representations are presented as well as their optimal formulations for trajectory planning. They are later compared.

Then, from the time-domain representation, we propose an effective online planning solution for autonomous vehicles where we focus on improving the computational load while preserving high levels of performance in racing scenarios. While most of the strategies in the literature overcome the planning problem using low complexity-based models, we aim to reformulate the non-linear vehicle equations to be expressed in an LPV form. To formulate the MPP problem, we first convexify the non-linear objective function in a linear-quadratic form. Then, we solve it using the LPV vehicle model for predicting the trajectory in a particular horizon. In addition, the algorithm has the ability of avoiding obstacles in a very simple way using exogenous track information. The limits of the lateral error vary inside the model predictive problem to take into account such static obstacles.

We test and compare the performance of the proposed strategy against its non-linear approach through simulations. We focus on the performance of our planning approach in a racing track. First, in a free track scenario and next in a scenario with static obstacles allow to show that the proposed method reduces the planning computation elapsed time while finding a suitable trajectory under the proposed constraints.

Chapter 11

Conclusions and future work

This thesis has proposed some contributions to the state estimation, automatic control and trajectory planning areas of the self driving field, with an emphasis to their application to racing situations. This chapter summarizes the work presented in this thesis, in order to review the main conclusions and explore the possibilities of further research.

11.1 Conclusions

Estimation, planning and control strategies for autonomous vehicles have been investigated throughout the last decades, and several theoretical and experimental results have been presented in the literature. Nevertheless, there is still space for further investigation. This thesis has contributed to the advance of the state of the art of this field.

- **Chapters 3 and 4** have addressed the variety of models that describe the vehicle motion from a non-linear perspective first and then their corresponding LPV representation has been formulated. It has been stated the main differences between kinematic and dynamic vehicle models as well as a comparison between Cartesian and Polar representations. It has been also shown how the non-linear embedding approach is used for building the LPV model representation. Finally, a particular space domain representation has been presented for a specific planning strategy for racing behaviours.

- **Chapter 5** has considered the problem of designing a non-linear control strategy based on Lyapunov stability criterion for controlling an autonomous ground vehicle. To adjust this controller, an iterative algorithm has been proposed to adjust the state feedback control law constants while ensuring overall system stability and certain levels of performance. To do so, using the LPV formulation of the reference-based kinematic model in closed-loop, an optimal asjustment of the controller has been made by solving an LQR-LMI based problem. Furthermore, such an adjustment has been improved by forcing the poles of the closed-loop dynamic part to be faster in comparison to the ones of the closed-loop kinematic part for achieving a decoupling between the two loops. Finally, it has been shown the performance of the vehicle in simulation obtaining satisfactory results, and it has been achieved the expected goal of moving autonomously from a starting point to a final point in a comfortable way in a real test scenario.

- In **Chapter 6**, a cascade control scheme for controlling an autonomous vehicle at normal driving behaviour has been presented. The outer layer controls the kinematics of the vehicle and the inner layer the vehicle dynamics. Both control designs have been carried out using the LQR/H_2 LMI-based formulation for polytopic kinematic LPV and dynamic LPV representations. Furthemore, an innovative LPV-UIO design has been presented to estimate dynamic vehicle states as well as the friction force as a exogenous disturbance. Then, a friction force compensation mechanism has been presented allowing the vehicle to compensate fast friction changes as well as reducing the control effort. Finally, the obtained gain scheduling LPV-LQR control approach, jointly with the LPV-UIO and a trajectory planning module, has presented suitable results in a simulated scenario. In the same way, a comparison has been shown about the friction force estimation, which shows the usefulness of this approach.

- **Chapter 7** has proposed a cascade control/observer structure for solving the autonomous racing problem. Then, a model predictive technique for controlling the kinematics of the vehicle, a GS-TS controller to deal with the vehicle dynamics and an optimal estimator have been considered. The novel kinematic control has been designed using the MPC technique with the prediction model expressed as a TS model using the non-linear embedding approach. On the other hand, the dynamic control has been addressed using the LQR strategy, with a TS modeling approach and using a discrete-time LMI formulation of the problem (TS-LMI-LQR). A comparison has been made between two methods of solving the control problem: using the NL-MPC and using the TS-MPC approach that is based on instantiating the TS model at each prediction step within the prediction stage using planning data. Such comparison has demonstrated a similar control performance but in a much faster way in the case of the TS-MPC technique. In addition, a novel estimation formulation (TS-MHE-UIO) has been introduced. This has been brought in with the aim of estimating dynamic states and exogenous disturbances acting on the vehicle.

- In **Chapter** 8, a solution to the racing control problem of autonomous vehicles has been proposed. An LPV-MPC strategy has been proposed as a novel approach to provide a realistic driving behaviour in real-time. In addition, using racing-based references provided by an external planner the controller makes the vehicle to perform in racing mode. For a good control performance, an offline identification of unknown vehicle coefficients has been carried out. The proposed strategies have been tested in simulation and in real experiments showing potential and similar results among them, thus strengthening the task of the simulator. In the real test, we have showed the contribution of the controller which have been able to solve a 20 steps prediction at 33 Hz and thus follow the racing trajectory previously established. The disadvantage found in this strategy is the system initialization due to the need to instantiate the LPV model.

- **Chapter** 9 has considered the study of introducing robust performance to the previous LPV-MPC scheme. Then, using a zonotope-tube-based LPV-MPC scheme, the focus has been on improving the computational load while preserving high levels of robustness and performance in racing scenarios. The tube has been computed using zonotopes theory which makes the Minkowski sum and difference of sets affordable from a computational point of view and less conservative since there is not approximations. Consequently, we have observed that the prediction horizon can be increased since, due to the precision of calculation using zonotopes, the reachable set does not grow conservatively maintaining a more adjusted shape. To reject the effect of acting disturbances maintaining robustness, a polytopic local controller has been designed solving the H_∞-based LMI problem. Furthermore, such a local controller is updated at a higher frequency than the nominal controller (LPV-MPC). Finally, we have tested and compared the performance of the proposed strategy against a current state of the art tube-based MPC. We have shown the effectiveness of the presented approach in a disturbed racing scenario being able to perform online tube-based MPC with a high performance and reduced computational cost.

- In **Chapter 10**, an innovative solution for the online trajectory planning problem considering static obstacles and focused on racing behaviours has been presented. Two representations, the space-domain and the time-domain, have been presented as well as their optimal formulations for trajectory planning. In the time-domain representation, we have proposed an effective online planning solution for autonomous vehicles where the focus is on improving the computational load while preserving high levels of performance in racing scenarios. This algorithm has enhanced the state of the art solutions since it reformulates the non-linear vehicle equations to be expressed in an LPV form, hence, speeding up the computations while preserving similar performance levels. In addition, the algorithm has shown the ability of avoiding obstacles in a very simple way using exogenous track information. We have tested and compared the performance of the proposed strategy against its non-linear approach through simulations. We have focused on the performance of our planning approach, first, in a free track scenario and later in a scenario with static obstacles allowing to show that the proposed method reduces the planning computation elapsed time while finding a suitable trajectory under the proposed constraints.

11.2 Perspectives and future work

This section resumes the open issues that could be addressed in future work.

- The analysis and synthesis based on LMI techniques for LPV models proposed in **Chapters 5 and 6** have demonstrated to generate controllers with some degree of conservativeness. Then, future research will study other design approaches such as the grid-based technique where conservatism is reduced.

- The different vehicle models used throughout the thesis have proved to be more than enough for the different applications addressed. However, in order to improve race behavior, a 4-wheel vehicle model should be developed as future work. Longitudinal dynamics of the tire and dynamics would be introduced in the steering column, thus obtaining a model whose control actions would be torques applied to the steering column and the wheel drive axle.

- The estimation methodologies presented in **Chapters 6 and 7** have been developed for third order dynamic vehicle models. Future research will extend the MHE-UIO strategy to estimate the complete set of vehicle states, i.e. kinematic and dynamic variables, using the robust zonotope-tube-based approach applyed in **Chapter 9**.

- The trajectory planning approach developed in **Chapter 10** has been addressed under static obstacle racing environment. However, the real world is constantly moving and therefore interaction with the mobile environment is necessary. Future research will extend the proposed technique to solve the problem of trajectory planning with moving obstacles. Furthermore, the study of a robust approach to ensure safe trajectories is necessary.

- Some of the presented control-estimation techniques have demonstrated an interesting performance in real tests. However, some others have not been experimentaly tested. Then, future work will implement the advanced techiques presented throughtout the thesis in embedded platforms and interesting discussions will be extracted.

- The racing oriented control technique developed in **Chapter 8** has achieved remarkable results. However, learning-based techniques like ANFIS will be investigated to verify whether an LPV control structure that represents the NL-MPC behaviour could be successfully learned.

- The robust MPC have been studied in **Chapter 9** by bounding the predicted error in an adaptive tube considering exogenous disturbances. However, future research will extend the zonotope-tube-based LPV-MPC technique to be robust also against uncertain scheduling variables.

Appendix A

Derivation of the reference-based kinematic model

The content of this appendix is based on the following work:

- Alcalá, Eugenio, et al. [2018.A] Alcalá, E., Puig, V., Quevedo, J., Escobet, T., & Comasolivas, R. (2018). Autonomous vehicle control using a kinematic Lyapunov-based technique with LQR-LMI tuning. *Control engineering practice*, 73, 1-12.

The dynamics of the error posture are what is needed for the trajectory tracking problem. Hence, (3.3) needs to be differentiated in order to obtain the error model. The equations of motion for each state of the error model are derived here for completeness.

- **Time derivative of x_e.**

The equation that need to be derived is the one correspondent to the first row in (2) such as

$$\begin{aligned}\dot{x}_e &= (\dot{x}_d - \dot{x})\cos(\theta) + (\dot{y}_d - \dot{y})\sin(\theta)\\ &\quad - (x_d - x)\dot{\theta}\sin(\theta) + (y_d - y)\dot{\theta}\cos(\theta)\\ &= \dot{x}_d\cos(\theta) - \dot{x}\cos(\theta) + \dot{y}_d\sin(\theta) - \dot{y}\sin(\theta)\\ &\quad - (x_d - x)\dot{\theta}\sin(\theta) + (y_d - y)\dot{\theta}\cos(\theta)\ .\end{aligned}$$

Applying the change $\dot{\theta} = \omega$ the last equation can be expressed as

$$\begin{aligned}\dot{x}_e = \dot{x}_d \cos(\theta) - \dot{x}\cos(\theta) + \dot{y}_d \sin(\theta) - \dot{y}\sin(\theta) \\ - (x_d - x)\omega\sin(\theta) + (y_d - y)\omega\cos(\theta) \ .\end{aligned}$$

From the equation of y_e in (3.3), we know that $y_e = -(x_d - x)\sin(\theta) + (y_d - y)\cos(\theta)$ which appears in the previous equality. Hence

$$\dot{x}_e = \dot{x}_d \cos(\theta) - \dot{x}\cos(\theta) + \dot{y}_d \sin(\theta) - \dot{y}\sin(\theta) + \omega y_e \ .$$

The negative terms of the previous equality, i.e. $-\dot{x}\cos(\theta)$ and $-\dot{y}\sin(\theta)$, can be developed using the equations of model (3.2) considering $\alpha = 0$

$$\begin{aligned}\dot{x}\cos(\theta) + \dot{y}\sin(\theta) &= v\cos(\theta)\cos(\theta) + v\sin(\theta)\sin(\theta) \\ &= v(\sin^2(\theta) + \cos^2(\theta)) = v\end{aligned}$$

and introducing this result in the previous one ($\dot{x}_e$) the following expression is obtained

$$\dot{x}_e = \dot{x}_d \cos(\theta) - v + \dot{y}_d \sin(\theta) + \omega y_e \ .$$

By definition: $\theta_e = \theta_d - \theta$, then $\theta = \theta_d - \theta_e$. Replacing θ in $\dot{x}_e$ results in

$$\dot{x}_e = \dot{x}_d \cos(\theta_d - \theta_e) - v + \dot{y}_d \sin(\theta_d - \theta_e) + \omega y_e \ .$$

The trigonometric identities for $\cos(\alpha - \beta)$ y $\sin(\alpha - \beta)$ are used in the next step

$$\begin{aligned}\dot{x}_e &= \dot{x}_d(\cos(\theta_d)\cos(\theta_e) + \sin(\theta_d)\sin(\theta_e)) - v \\ &\quad + \dot{y}_d(\sin(\theta_d)\cos(\theta_e) - \cos(\theta_d)\sin(\theta_e)) + \omega y_e \\ &= \omega y_e - v + (\dot{x}_d\cos(\theta_d) + \dot{y}_d\sin(\theta_d))\cos(\theta_e) \\ &\quad + (\dot{x}_d\sin(\theta_d) - \dot{y}_d\cos(\theta_d))\cos(\theta_e) \ .\end{aligned}$$

The non-holonomic constraint for the real wheels is: $\dot{x}_d \sin(\theta_d) = \dot{y}_d \cos(\theta_d)$. Therefore

$$\dot{x}_e = \omega y_e - v + (\dot{x}_d\cos(\theta_d) + \dot{y}_d\sin(\theta_d))\cos(\theta_e) \ .$$

Following the same procedure that was used before, the terms inside the parenthesis become

$$\begin{aligned}\dot{x}_d \cos(\theta_d) + \dot{y}_d \sin(\theta_d) &= v_d \cos(\theta_d)\cos(\theta_d) + v_d \sin(\theta_d)\sin(\theta_d) \\ &= v_d(\sin^2(\theta_d) + \cos^2(\theta_d)) = v_d \ .\end{aligned}$$

Finally, using the previous equality, the result for $\dot{x}_e$ is

$$\dot{x}_e = \omega y_e - v + v_d \cos(\theta_e) \ . \tag{A.1}$$

- **Time derivative of y_e.**

The derivation of the $\dot{y}_e$ is similar to the one used for $\dot{x}_e$. The equation that need to be derived is the one correspondent to the second row in (3.3) such as

$$\begin{aligned}\dot{y}_e = &-(\dot{x}_d - \dot{x})\sin(\theta) + (\dot{y}_d - \dot{y})\cos(\theta) \\ &-(x_d - x)\dot{\theta}\cos(\theta) - (y_d - y)\dot{\theta}\sin(\theta) \ .\end{aligned}$$

We had $x_e = (x_d - x)\cos(\theta) + (y_d - y)\sin(\theta)$ from (2) which appears in the previous equality and we also know $\dot{\theta} = \omega$. Hence the last expression can be represented as

$$\begin{aligned}\dot{y}_e &= -(\dot{x}_d - \dot{x})\sin(\theta) + (\dot{y}_d - \dot{y})\cos(\theta) - x_e\omega \\ &= -x_e\omega + \dot{x}\sin(\theta) - \dot{y}\cos(\theta) - \dot{x}_d\sin(\theta) + \dot{y}_d\cos(\theta) \ .\end{aligned}$$

The non-holonomic constraint for the rear wheels is: $\dot{x}\sin(\theta) = \dot{y}\cos(\theta)$. Therefore

$$\dot{y}_e = -x_e\omega - \dot{x}_d\sin(\theta) + \dot{y}_d\cos(\theta_c ur) \ .$$

The error in θ is $\theta_e = \theta_d - \theta$, then $\theta = \theta_d - \theta_e$. Replacing θ in the previous equation

$$\dot{y}_e = -x_e\omega - \dot{x}_d\sin(\theta_d - \theta_e) + \dot{y}_d\cos(\theta_d - \theta_e) \ .$$

The trigonometric identities for $\cos(\alpha - \beta)$ y $\sin(\alpha - \beta)$ are used in the next step

$$\begin{aligned}
\dot{y}_e &= -x_e\omega - \dot{x}_d(\sin(\theta_d)\cos(\theta_e) - \cos(\theta_d)\sin(\theta_e)) \\
&\quad + \dot{y}_d(\cos(\theta_d)\cos(\theta_e) + \sin(\theta_d)\sin(\theta_e)) \\
&= -x_e\omega + (\dot{x}_d\cos(\theta_d) + \dot{y}_d\sin(\theta_d))\sin(\theta_e) \\
&\quad + (\dot{y}_d\cos(\theta_d) - \dot{x}_d\sin(\theta_d))\cos(\theta_e)\ .
\end{aligned}$$

The same non-holonomic constraint is fulfilled for the reference car $\dot{x}_d\sin(\theta_d) = \dot{y}_d\cos(\theta_d)$. Using this constraint in $\dot{y}_e$

$$\dot{y}_e = -x_e\omega + (\dot{x}_d\cos(\theta_d) + \dot{y}_d\sin(\theta_d)\sin(\theta_e)\ .$$

Following the same procedure that was used before, the terms inside the parenthesis become

$$\begin{aligned}
\dot{x}_d\cos(\theta_d) + \dot{y}_d\sin(\theta_d) &= v_d\cos(\theta_d)\cos(\theta_d) + v_d\sin(\theta_d)\sin(\theta_d) \\
&= v_d(\sin^2(\theta_d) + \cos^2(\theta_d)) = v_d\ .
\end{aligned}$$

Finally, using the previous equality, the result for $\dot{y}_e$ is

$$\dot{y}_e = -x_e\omega + v_d\sin(\theta_e)\ . \tag{A.2}$$

- **Time derivative of θ_e.**

The last state that need to be derivated is θ_e. This one is straightforward, the equation to differentiate is: $\theta_e = \theta_d - \theta$.

The result is

$$\dot{\theta}_e = \dot{\theta}_d - \dot{\theta} = \omega_d - \omega\ . \tag{A.3}$$

Kinematic error model equations

The obtained result for the error model, equations (A.1) to (A.3), are shown here in matrix form

$$\begin{bmatrix} \dot{x}_e \\ \dot{y}_e \\ \dot{\theta}_e \end{bmatrix} = \begin{bmatrix} \omega y_e - v + v_d \cos(\theta_e) \\ -\omega x_e + v_d \sin(\theta_e) \\ \omega_d - \omega \end{bmatrix} . \tag{A.4}$$

Bibliography

Alcalá, E., Sellart, L., Puig, V., Quevedo, J., Saludes, J., Vázquez, D., & López, A. (2016, June). Comparison of two non-linear model-based control strategies for autonomous vehicles. In *2016 24th Mediterranean Conference on Control and Automation (MED)* (pp. 846-851). IEEE.

Alcalá, E., Puig, V. & Quevedo, J. (2019). LPV-MPC Control for Autonomous Vehicles. In *IFAC-PapersOnLine*, 52(28), 106-113.

Alcalá, E., Puig, V., Quevedo, J., Escobet, T., & Comasolivas, R. (2018). Autonomous vehicle control using a kinematic Lyapunov-based technique with LQR-LMI tuning. *Control engineering practice*, 73, 1-12.

Alcalá, E., Puig, V., Quevedo, J., & Escobet, T. (2018). Gain-scheduling LPV control for autonomous vehicles including friction force estimation and compensation mechanism. *IET Control Theory & Applications*, 12(12), 1683-1693.

Alcalá, E., Cayuela, V. P., & Casin, J. Q. (2019). TS-MPC for Autonomous Vehicles including a TS-MHE-UIO estimator. *IEEE Transactions on Vehicular Technology.*

Alcalá, E., Puig, V., Quevedo, J. & Rosolia, U. (2020). Autonomous Racing using Linear Parameter Varying - Model Predictive Control (LPV-MPC). *Control Engineering Practice*, 95, 104270.

Alcalá, E., Puig, V. & Quevedo, J. (2019). LPV-MP Planning for Autonomous Racing Vehicles considering Obstacles. *Robotics and Autonomous Systems*, 103392.

Alcalá, E., Puig, V., Quevedo, J. & Sename, O. (2020). Fast Zonotope-Tube-based LPV-MPC for Autonomous Vehicles. *International Journal of Control.* (submitted).

Litman, T. (September 2019). Autonomous vehicle implementation predictions (p. 28). *Victoria, Canada: Victoria Transport Policy Institute.*

Morando, M. M., Tian, Q., Truong, L. T., & Vu, H. L. (2018). Studying the safety impact of autonomous vehicles using simulation-based surrogate safety measures. *Journal of Advanced Transportation*, 2018.

Papadoulis, A., Quddus, M., & Imprialou, M. (2019). Evaluating the safety impact of connected and autonomous vehicles on motorways. *Accident Analysis & Prevention*, 124, 12-22.

SAE international. (2016) Taxonomy and Definitions for Terms Related to Driving Automation Systems for On-Road Motor Vehicles. *SAE International*, (J3016)

Nie, L., Guan, J., Lu, C., Zheng, H., & Yin, Z. (2018). Longitudinal speed control of autonomous vehicle based on a self-adaptive PID of radial basis function neural network. *IET Intelligent Transport Systems*, 12(6), 485-494.

Hu, C., Qin, Y., Cao, H., Song, X., Jiang, K., Rath, J. J., & Wei, C. (2019). Lane keeping of autonomous vehicles based on differential steering with adaptive multivariable super-twisting control. *Mechanical Systems and Signal Processing*, 125, 330-346.

Guo, J., Luo, Y., & Li, K. (2019). Robust gain-scheduling automatic steering control of unmanned ground vehicles under velocity-varying motion. *Vehicle System Dynamics*, 57(4), 595-616.

Guo, H., Cao, D., Chen, H., Sun, Z., & Hu, Y. (2019). Model predictive path following control for autonomous cars considering a measurable disturbance: Implementation, testing, and verification. *Mechanical Systems and Signal Processing*, 118, 41-60.

Guo, J., Luo, Y., & Li, K. (2018). Adaptive non-linear trajectory tracking control for lane change of autonomous four-wheel independently drive electric vehicles. *IET Intelligent Transport Systems*, 12(7), 712-720.

Zhang, C., Hu, J., Qiu, J., Yang, W., Sun, H., & Chen, Q. (2018). A novel fuzzy observer-based steering control approach for path tracking in autonomous vehicles. *IEEE Transactions on Fuzzy Systems*, 27(2), 278-290.

Li, H. M., Wang, X. B., Song, S. B., & Li, H. (2016). Vehicle control strategies analysis based on PID and fuzzy logic control. *Procedia engineering*, 137, 234-243.

Wang, X., Fu, M., Ma, H., & Yang, Y. (2015). Lateral control of autonomous vehicles based on fuzzy logic. *Control Engineering Practice*, 34, 1-17.

Li, S., Li, K., Rajamani, R., & Wang, J. (2010). Model predictive multi-objective vehicular adaptive cruise control. *IEEE Transactions on Control Systems Technology*, 19(3), 556-566.

Marino, R., Scalzi, S., Orlando, G., & Netto, M. (2009, June). A nested PID steering control for lane keeping in vision based autonomous vehicles. In *2009 American Control Conference* (pp. 2885-2890). IEEE.

Zhao, P., Chen, J., Song, Y., Tao, X., Xu, T., & Mei, T. (2012). Design of a control system for an autonomous vehicle based on adaptive-pid. *International Journal of Advanced Robotic Systems*, 9(2), 44.

Tagne, G., Talj, R., & Charara, A. (2013, June). Higher-order sliding mode control for lateral dynamics of autonomous vehicles, with experimental validation. In 2013 IEEE *Intelligent Vehicles Symposium (IV)* (pp. 678-683). IEEE.

Nam, K., Fujimoto, H., & Hori, Y. (2015). Design of an adaptive sliding mode controller for robust yaw stabilisation of in-wheel-motor-driven electric vehicles. *International Journal of Vehicle Design*, 67(1), 98-113.

Besselmann, T. (2010). Constrained optimal control: piecewise affine and linear parameter-varying systems. *ETH Zurich.*

Kang, J., Hindiyeh, R. Y., Moon, S. W., Gerdes, J. C., & Yi, K. (2008). Design and testing of a controller for autonomous vehicle path tracking using GPS/INS sensors. *IFAC Proceedings Volumes*, 41(2), 2093-2098.

Zheng, Y., Li, S. E., Li, K., Borrelli, F., & Hedrick, J. K. (2016). Distributed model predictive control for heterogeneous vehicle platoons under unidirectional topologies. *IEEE Transactions on Control Systems Technology*, 25(3), 899-910.

Keviczky, T., Falcone, P., Borrelli, F., Asgari, J., & Hrovat, D. (2006, June). Predictive control approach to autonomous vehicle steering. In *2006 American control conference* (pp. 6-pp). IEEE.

Gray, A., Gao, Y., Hedrick, J. K., & Borrelli, F. (2013, June). Robust predictive control for semi-autonomous vehicles with an uncertain driver model. In *2013 IEEE Intelligent Vehicles Symposium (IV)* (pp. 208-213). IEEE.

Shamma, J. S. (1988). Analysis and design of gain scheduled control systems (Doctoral dissertation, Massachusetts Institute of Technology).

Shamma, J. S. (2012). An overview of LPV systems. In *Control of linear parameter varying systems with applications* (pp. 3-26). Springer, Boston, MA.

Takagi, T., & Sugeno, M. (1985). Fuzzy identification of systems and its applications to modeling and control. *IEEE transactions on systems, man, and cybernetics*, (1), 116-132.

Cao, Y. Y., & Frank, P. M. (2001). Stability analysis and synthesis of nonlinear time-delay systems via linear Takagi–Sugeno fuzzy models. *Fuzzy sets and systems*, 124(2), 213-229.

Elbanhawi, M., Simic, M., & Jazar, R. N. (2014, June). Continuous-Curvature Bounded Trajectory Planning Using Parametric Splines. In *IDT/IIMSS/STET* (pp. 513-522).

LaValle, S. M. (1998). Rapidly-exploring random trees: A new tool for path planning.

Kuwata, Y., Teo, J., Fiore, G., Karaman, S., Frazzoli, E., & How, J. P. (2009). Real-time motion planning with applications to autonomous urban driving. *IEEE Transactions on Control Systems Technology*, 17(5), 1105-1118.

Karaman, S., & Frazzoli, E. (2010, December). Optimal kinodynamic motion planning using incremental sampling-based methods. In *49th IEEE conference on decision and control (CDC)* (pp. 7681-7687). IEEE.

Karaman, S., & Frazzoli, E. (2011). Sampling-based algorithms for optimal motion planning. *The international journal of robotics research*, 30(7), 846-894.

Janson, L., & Pavone, M. (2016). Fast Marching Trees: A fast marching sampling-based method for optimal motion planning in many dimensions. In *Robotics Research* (pp. 667-684). Springer, Cham.

Schmerling, E., Janson, L., & Pavone, M. (2015, May). Optimal sampling-based motion planning under differential constraints: the driftless case. In *2015 IEEE International Conference on Robotics and Automation (ICRA)* (pp. 2368-2375). IEEE.

Jang, J. S. (1993). ANFIS: adaptive-network-based fuzzy inference system. *IEEE transactions on systems, man, and cybernetics*, 23(3), 665-685.

Kong, J., Pfeiffer, M., Schildbach, G., & Borrelli, F. (2015, June). Kinematic and dynamic vehicle models for autonomous driving control design. In *2015 IEEE Intelligent Vehicles Symposium (IV)* (pp. 1094-1099). IEEE.

Pacejka, H. (2005). Tire and vehicle dynamics. *Elsevier.*

Broggi, A., Medici, P., Zani, P., Coati, A., & Panciroli, M. (2012). Autonomous vehicles control in the VisLab intercontinental autonomous challenge. *Annual Reviews in Control*, 36(1), 161-171.

Tzafestas, S. G. (2013). Introduction to mobile robot control. *Elsevier.*

Brunner, M., Rosolia, U., Gonzales, J., & Borrelli, F. (2017, December). Repetitive learning model predictive control: An autonomous racing example. In *2017 IEEE 56th Annual Conference on Decision and Control (CDC)* (pp. 2545-2550). IEEE.

Caporale, D., Fagiolini, A., Pallottino, L., Settimi, A., Biondo, A., Amerotti, F., ... & Venturini, L. (2018, September). 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)* (pp. 1-6). IEEE.

Kazantzis, N., Chong, K. T., Park, J. H., & Parlos, A. G. (2005). Control-relevant discretization of nonlinear systems with time-delay using Taylor-Lie series. *Journal of dynamic systems, Measurement, and Control*, 127(1), 153-159.

Ali, M., Abbas, H., & Werner, H. (2010, December). Controller synthesis for input-output LPV models. In *49th IEEE Conference on Decision and Control (CDC)* (pp. 7694-7699). IEEE.

Sename, O., Gaspar, P., & Bokor, J. (Eds.). (2013). Robust control and linear parameter varying approaches: application to vehicle dynamics (Vol. 437). *Springer.*

Toth, R. (2010). Modeling and identification of linear parameter-varying systems (Vol. 403). *Springer.*

Toth, R., Heuberger, P. S., & Van den Hof, P. M. (2010). Discretisation of linear parameter-varying state-space representations. *IET control theory & applications*, 4(10), 2082-2096.

Bachnas, A. A., Tóth, R., Ludlage, J. H. A., & Mesbah, A. (2014). A review on data-driven linear parameter-varying modeling approaches: A high-purity distillation column case study. *Journal of Process Control*, 24(4), 272-285.

Mejari, M. D. (2018). Towards automated data-driven modeling of linear parameter-varying systems.

Rizvi, S. Z., Velni, J. M., Abbasi, F., Toth, R., & Meskin, N. (2018). State-space LPV model identification using kernelized machine learning. *Automatica*, 88, 38-47.

Hoffmann, C., & Werner, H. (2014). A survey of linear parameter-varying control applications validated by experiments or high-fidelity simulations. *IEEE Transactions on Control Systems Technology*, 23(2), 416-433.

Kwiatkowski, A., Boll, M. T., & Werner, H. (2006, December). Automated generation and assessment of affine LPV models. In *Proceedings of the 45th IEEE Conference on Decision and Control* (pp. 6690-6695). IEEE.

Rotondo, D., Puig, V., Nejjari, F., & Witczak, M. (2015). Automated generation and comparison of Takagi–Sugeno and polytopic quasi-LPV models. *Fuzzy Sets and Systems*, 277, 44-64.

Rajamani, R. (2011). Vehicle dynamics and control. *Springer Science & Business Media.*

Hahn, S., Zindler, K., & Jumar, U. (2016). Two-degrees-of-freedom lateral vehicle control using nonlinear model based disturbance compensation. *IFAC-PapersOnLine*, 49(11), 182-189.

Du, X., & Tan, K. K. (2016, June). Autonomous vehicle velocity and steering control through nonlinear model predictive control scheme. In *2016 IEEE Transportation Electrification Conference and Expo, Asia-Pacific* (ITEC Asia-Pacific) (pp. 001-006). IEEE.

Németh, B., Gáspár, P., & Bokor, J. (2016, July). LPV-based integrated vehicle control design considering the nonlinear characteristics of the tire. In *2016 American Control Conference (ACC)* (pp. 6893-6898). IEEE.

Jacobs, L., Hilhorst, G., Swevers, J., & Pipeleers, G. (2017). Linear parameter-varying control of an industrial AGV using a novel practical design interface. In *Benelux Meeting on Systems and Control*, Date: 2017/03/28-2017/03/30, Location: Spa, Belgium.

Zhang, H., & Wang, J. (2015). Vehicle lateral dynamics control through AFS/DYC and robust gain-scheduling approach. *IEEE Transactions on Vehicular Technology*, 65(1), 489-494.

Soualmi, B., Sentouh, C., Popieul, J. C., & Debernard, S. (2014). Automation-driver cooperative driving in presence of undetected obstacles. *Control engineering practice*, 24, 106-119.

Nguyen, A. T., Sentouh, C., & Popieul, J. C. (2016). Takagi-sugeno model-based steering control for autonomous vehicles with actuator saturation. *IFAC-PapersOnLine*, 49(5), 206-211.

Menhour, L., d'Andréa-Novel, B., Fliess, M., & Mounier, H. (2014). Coupled nonlinear vehicle control: Flatness-based setting with algebraic estimation techniques. *Control Engineering Practice*, 22, 135-146.

Attia, R., Orjuela, R., & Basset, M. (2014). Combined longitudinal and lateral control for automated vehicle guidance. *Vehicle System Dynamics*, 52(2), 261-279.

Menhour, L., D'Andréa-Novel, B., Fliess, M., & Mounier, H. (2013, December). Multivariable decoupled longitudinal and lateral vehicle control: A model-free design. In *52nd IEEE Conference on Decision and Control* (pp. 2834-2839). IEEE.

Blažič, S. (2010). Takagi-sugeno vs.lyapunov-based tracking control for a wheeled mobile robot. *WSEAS Transactions on Systems and Control*, 5(8), 667-676.

Freeman, R., & Kokotovic, P. V. (2008). Robust nonlinear control design: state-space and Lyapunov techniques. *Springer Science & Business Media.*

Dixon, W. E., Dawson, D. M., Zergeroglu, E., & Behal, A. (2001). Nonlinear control of wheeled mobile robots (Vol. 175). London: *Springer.*

Duan, G. R., & Yu, H. H. (2013). LMIs in control systems: analysis, design and applications. *CRC press.*

Indiveri, G. (1999, December). Kinematic time-invariant control of a 2D nonholonomic vehicle. In *Proceedings of the 38th IEEE Conference on Decision and Control* (Cat. No. 99CH36304) (Vol. 3, pp. 2112-2117). IEEE.

Marino, R., Scalzi, S., Orlando, G., & Netto, M. (2009, June). A nested PID steering control for lane keeping in vision based autonomous vehicles. In *2009 American Control Conference* (pp. 2885-2890). IEEE.

Shamma, J. S. (2012). An overview of LPV systems. In Control of linear parameter varying systems with applications (pp. 3-26). *Springer*, Boston, MA.

Garcia, A. (2010). Edy's Vehicle Physics. http://www.edy.es/dev/vehicle-physics/

El Ghaoui, L., & Balakrishnan, V. (1994, December). Synthesis of fixed-structure controllers via numerical optimization. In *Proceedings of 1994 33rd IEEE Conference on Decision and Control* (Vol. 3, pp. 2678-2683). IEEE.

Farag, A., & Werner, H. (2004, June). A Riccati-genetic algorithms approach to fixed-structure controller synthesis. In *Proceedings of the 2004 American Control Conference* (Vol. 3, pp. 2799-2804). IEEE.

Solea, R., & Nunes, U. (2007). Trajectory planning and sliding-mode control based trajectory-tracking for cybercars. *Integrated Computer-Aided Engineering*, 14(1), 33-47.

Bianco, C. L., Piazzi, A., & Romano, M. (2004, June). Velocity planning for autonomous vehicles. In *IEEE Intelligent Vehicles Symposium*, 2004 (pp. 413-418). IEEE.

International Organization for Standarization. (1997). International Standard ISO 2631-1: Mechanical Vibration and Shock-evaluation of Human Exposure to Wholre-body Vibration. *International Organization for Standardization.*

Aicardi, M., Casalino, G., Bicchi, A., & Balestrino, A. (1995). Closed loop steering of unicycle like vehicles via Lyapunov techniques. *IEEE Robotics & Automation Magazine*, 2(1), 27-35.

Dixon, W. E., Dawson, D. M., Zergeroglu, E., & Behal, A. (2001). Nonlinear control of wheeled mobile robots (Vol. 175). London: *Springer.*

Samson, C., & Ait-Abderrahim, K. (1990). Mobile robot control. part 1: Feedback control of nonholonomic wheeled cart in cartesian space.

Khalil, H. K. (2002). Nonlinear systems. *Upper Saddle River.*

Hung, J. Y., Gao, W., & Hung, J. C. (1993). Variable structure control: A survey. *IEEE transactions on industrial electronics*, 40(1), 2-22.

Gao, Y., Gray, A., Tseng, H. E., & Borrelli, F. (2014). A tube-based robust nonlinear predictive control approach to semiautonomous ground vehicles. *Vehicle System Dynamics*, 52(6), 802-823.

Carvalho, A., Lefévre, S., Schildbach, G., Kong, J., & Borrelli, F. (2015). Automated driving: The role of forecasts and uncertainty—A control perspective. *European Journal of Control*, 24, 14-32.

Geiger, A., Lauer, M., Moosmann, F., Ranft, B., Rapp, H., Stiller, C., & Ziegler, J. (2012). Team annieway's entry to the 2011 grand cooperative driving challenge. *IEEE Transactions on Intelligent Transportation Systems*, 13(3), 1008-1017.

Association for Safe International Road Travel (2016). http://asirt.org/initiatives/informing-road-users/road-safety-facts/road-crash-statistics

Urmson, C., Anhalt, J., Bagnell, D., Baker, C., Bittner, R., Clark, M. N., ... & Gittleman, M. (2009). Autonomous driving in urban environments: Boss and the urban challenge. In *The DARPA Urban Challenge* (pp. 1-59). Springer, Berlin, Heidelberg.

Carvalho, A., Lefévre, S., Schildbach, G., Kong, J., & Borrelli, F. (2015). Automated driving: The role of forecasts and uncertainty—A control perspective. *European Journal of Control*, 24, 14-32.

Urmson, C. (2012). The self-driving car logs more miles on new wheels. *Google official blog*, 7.

Broggi, A., Medici, P., Zani, P., Coati, A., & Panciroli, M. (2012). Autonomous vehicles control in the VisLab intercontinental autonomous challenge. *Annual Reviews in Control*, 36(1), 161-171.

Gannes, L. (2014). Google Introduces New Self Driving Car at the Code Conference-Re/code. Re/code.

Halleck, T. (2015). Google Inc. says self-driving car will be ready by 2020. *International Business Times*.

Levinson, J., Askeland, J., Becker, J., Dolson, J., Held, D., Kammel, S., ... & Sokolsky, M. (2011, June). Towards fully autonomous driving: Systems and algorithms. In *2011 IEEE Intelligent Vehicles Symposium (IV)* (pp. 163-168). IEEE.

Ziegler, J., Bender, P., Schreiber, M., Lategahn, H., Strauss, T., Stiller, C., ... & Kaus, E. (2014). Making Bertha drive—An autonomous journey on a historic route. *IEEE Intelligent transportation systems magazine*, 6(2), 8-20.

Ros, G., Sellart, L., Materzynska, J., Vazquez, D., & Lopez, A. M. (2016). The synthia dataset: A large collection of synthetic images for semantic segmentation of urban scenes. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 3234-3243).

Tazzari electric vehicle. http://www.tazzari-zero.com/es/

Haklay, M., & Weber, P. (2008). Openstreetmap: User-generated street maps. *IEEE Pervasive Computing*, 7(4), 12-18.

Chen, H., Guo, L., Qu, T., Gao, B., & Wang, F. (2017). Optimal control methods in intelligent vehicles. *Journal of Control and Decision*, 4(1), 32-56.

Paden, B., Čáp, M., Yong, S. Z., Yershov, D., & Frazzoli, E. (2016). A survey of motion planning and control techniques for self-driving urban vehicles. *IEEE Transactions on intelligent vehicles*, 1(1), 33-55.

Franklin, G. F., Powell, J. D., & Workman, M. L. (1998). Digital control of dynamic systems (Vol. 3). Menlo Park, CA: Addison-wesley.

Indiveri, G. (1999, December). Kinematic time-invariant control of a 2D nonholonomic vehicle. In *Proceedings of the 38th IEEE Conference on Decision and Control* (Cat. No. 99CH36304) (Vol. 3, pp. 2112-2117). IEEE.

Apkarian, P., Gahinet, P., & Becker, G. (1995). Self-scheduled H$_{infty}$ control of linear parameter-varying systems: a design example. *Automatica*, 31(9), 1251-1261.

Nawash, N. (2005). H-infinity control of an autonomous mobile robot (Doctoral dissertation, Cleveland State University).

Farrokhsiar, M., Pavlik, G., & Najjaran, H. (2013). An integrated robust probing motion planning and control scheme: A tube-based MPC approach. *Robotics and Autonomous Systems*, 61(12), 1379-1391.

González, R., Fiacchini, M., Guzmán, J. L., Álamo, T., & Rodríguez, F. (2011). Robust tube-based predictive control for mobile robots in off-road conditions. *Robotics and Autonomous Systems*, 59(10), 711-726.

Olsson, C. (2015). Model complexity and coupling of longitudinal and lateral control in autonomous vehicles using model predictive control.

Schwarting, W., Alonso-Mora, J., Pauli, L., Karaman, S., & Rus, D. (2017, May). Parallel autonomy in automated vehicles: Safe motion generation with minimal intervention. In *2017 IEEE International Conference on Robotics and Automation (ICRA)* (pp. 1928-1935). IEEE.

Fergani, S., Sename, O., & Dugard, L. (2015). An LPV/H_∞ Integrated Vehicle Dynamic Controller. *IEEE Transactions on Vehicular Technology*, 65(4), 1880-1889.

Polack, P., d'Andréa-Novel, B., Fliess, M., de La Fortelle, A., & Menhour, L. (2017). Finite-time stabilization of longitudinal control for autonomous vehicles via a model-free approach. *IFAC-PapersOnLine*, 50(1), 12533-12538.

Nahidi, A., Kasaiezadeh, A., Khosravani, S., Khajepour, A., Chen, S. K., & Litkouhi, B. (2017). Modular integrated longitudinal and lateral vehicle stability control for electric vehicles. *Mechatronics*, 44, 60-70.

Bascetta, L., Ferretti, G., Matteucci, M., & Bossi, M. (2016). LFT-based MPC control of an autonomous vehicle. *IFAC-PapersOnLine*, 49(15), 7-12.

Rotondo, D. (2017). Advances in gain-scheduling and fault tolerant control techniques. *Springer.*

Ostertag, E. (2011). Mono-and multivariable control and estimation: linear, quadratic and LMI methods. *Springer Science & Business Media.*

Gáspár, P., Szabó, Z., Bokor, J., & Nemeth, B. (2016). Robust control design for active driver assistance systems. *Springer*, DOI, 10, 978-3.

Desa, U. N. (2014). World urbanization prospects, the 2011 revision. *Population Division, Department of Economic and Social Affairs, United Nations Secretariat.*

Van Woensel, L., Archer, G., Panades-Estruch, L., & Vrscaj, D. (2015). Ten technologies which could change our lives. *Scientific Foresight Unit, European Parliament Research Services.* 28p. DOI, 10, 610145.

Rotondo, D., Cristofaro, A., Johansen, T. A., Nejjari, F., & Puig, V. (2018). Diagnosis of icing and actuator faults in UAVs using LPV unknown input observers. *Journal of Intelligent & Robotic Systems*, 91(3-4), 651-665.

Yi, K., Hedrick, K., & Lee, S. C. (1999). Estimation of tire-road friction using observer based identifiers. *Vehicle System Dynamics*, 31(4), 233-261.

Svendenius, J. (2007). Tire modeling and friction estimation (Doctoral dissertation, Department of Automatic Control, Lund University).

Dakhlallah, J., Glaser, S., Mammar, S., & Sebsadji, Y. (2008, June). Tire-road forces estimation using extended Kalman filter and sideslip angle evaluation. In *2008 American control conference* (pp. 4597-4602). IEEE.

Pletschen, N., & Diepold, K. J. (2017). Nonlinear state estimation for suspension control applications: a Takagi-Sugeno Kalman filtering approach. *Control Engineering Practice*, 61, 292-306.

Tanaka, K., & Wang, H. O. (2004). Fuzzy control systems design and analysis: a linear matrix inequality approach. *John Wiley & Sons.*

Cho, W., Yoon, J., Yim, S., Koo, B., & Yi, K. (2009). Estimation of tire forces for application to vehicle stability control. *IEEE Transactions on Vehicular Technology*, 59(2), 638-649.

Wang, Y., Bevly, D. M., & Chen, S. K. (2013, September). Longitudinal tire force estimation with unknown input observer. In *ASME 2012 5th Annual Dynamic Systems and Control Conference joint with the JSME 2012 11th Motion and Vibration Conference* (pp. 523-530). American Society of Mechanical Engineers Digital Collection.

Grip, H. F., Imsland, L., Johansen, T. A., Fossen, T. I., Kalkkuhl, J. C., & Suissa, A. (2008). Nonlinear vehicle side-slip estimation with friction adaptation. *Automatica*, 44(3), 611-622.

Rajamani, R., Phanomchoeng, G., Piyabongkarn, D., & Lew, J. Y. (2011). Algorithms for real-time estimation of individual wheel tire-road friction coefficients. *IEEE/ASME Transactions on Mechatronics*, 17(6), 1183-1195.

Khaleghian, S., Emami, A., & Taheri, S. (2017). A technical survey on tire-road friction estimation. *Friction*, 5(2), 123-146.

Cisneros, P. S., Voss, S., & Werner, H. (2016, December). Efficient nonlinear model predictive control via quasi-lpv representation. In *2016 IEEE 55th Conference on Decision and Control (CDC)* (pp. 3216-3221). IEEE.

Keller, J. Y., & Darouach, M. (1999). Two-stage Kalman estimator with unknown exogenous inputs. *Automatica*, 35(2), 339-342.

Optimization, Gurobi. (2014). Gurobi optimizer reference manual. *Google Scholar*

Löfberg, J. (2004, September). YALMIP: A toolbox for modeling and optimization in MATLAB. In *Proceedings of the CACSD Conference* (Vol. 3).

Gonzales, J., Zhang, F., Li, K., & Borrelli, F. (2016, September). Autonomous drifting with onboard sensors. In *Advanced Vehicle Control: Proceedings of the 13th International Symposium on Advanced Vehicle Control (AVEC'16)*, September 13-16, 2016, Munich, Germany (p. 133).

O'Kane, J. M. (2014). A gentle introduction to ROS.

Wächter, A., & Biegler, L. T. (2006). On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming. *Mathematical programming*, 106(1), 25-57.

Ni, J., & Hu, J. (2017). Dynamics control of autonomous vehicle at driving limits and experiment on an autonomous formula racing car. *Mechanical Systems and Signal Processing*, 90, 154-174.

Rosolia, U., Carvalho, A., & Borrelli, F. (2017, May). Autonomous racing using learning model predictive control. In *2017 American Control Conference (ACC)* (pp. 5115-5120). IEEE.

Laurense, V. A., Goh, J. Y., & Gerdes, J. C. (2017, May). Path-tracking for autonomous vehicles at the limit of friction. In *2017 American Control Conference (ACC)* (pp. 5586-5591). IEEE.

Li, X., Sun, Z., Cao, D., Liu, D., & He, H. (2017). Development of a new integrated local trajectory planning and tracking control framework for autonomous ground vehicles. *Mechanical Systems and Signal Processing*, 87, 118-137.

Brunner, M., Rosolia, U., Gonzales, J., & Borrelli, F. (2017, December). Repetitive learning model predictive control: An autonomous racing example. In *2017 IEEE 56th Annual Conference on Decision and Control (CDC)* (pp. 2545-2550). IEEE.

Law, C. K., Dalal, D., & Shearrow, S. (2018). Robust Model Predictive Control for Autonomous Vehicles/Self Driving Cars. arXiv preprint arXiv:1805.08551.

Stellato, B., Banjac, G., Goulart, P., Bemporad, A., & Boyd, S. (2018, September). OSQP: An operator splitting solver for quadratic programs. In *2018 UKACC 12th International Conference on Control (CONTROL)* (pp. 339-339). IEEE.

Bujarbaruah, M., Zhang, X., Tseng, H. E., & Borrelli, F. (2018). Adaptive MPC for Autonomous Lane Keeping. arXiv preprint arXiv:1806.04335.

Schramm, D., Hiller, M., & Bardini, R. (2014). Vehicle dynamics. Modeling and Simulation. Berlin, Heidelberg, 151.

Brown, M., Funke, J., Erlien, S., & Gerdes, J. C. (2017). Safe driving envelopes for path tracking in autonomous vehicles. *Control Engineering Practice*, 61, 307-316.

Li, C., Li, X., Li, J., Zhu, Q., & Dai, B. (2017, August). Trajectory Planning for Autonomous Ground Vehicles Driving in Structured Environments. In *2017 9th International Conference on Intelligent Human-Machine Systems and Cybernetics (IHMSC)* (Vol. 2, pp. 41-46). IEEE.

Gao, Y., Gray, A., Frasch, J. V., Lin, T., Tseng, E., Hedrick, J. K., & Borrelli, F. (2012, September). Spatial predictive control for agile semi-autonomous ground vehicles. In *Proceedings of the 11th international symposium on advanced vehicle control* (No. 2, pp. 1-6).

Liu, H., Lai, X., & Wu, W. (2013). Time-optimal and jerk-continuous trajectory planning for robot manipulators with kinematic constraints. *Robotics and Computer-Integrated Manufacturing*, 29(2), 309-317.

Rosolia, U., & Borrelli, F. (2019). Learning How to Autonomously Race a Car: a Predictive Control Approach. arXiv preprint arXiv:1901.08184.

Rosolia, U., Zhang, X., & Borrelli, F. (2019, July). Simple policy evaluation for data-rich iterative tasks. In *2019 American Control Conference (ACC)* (pp. 2855-2860). IEEE.

Jungers, M., Oliveira, R. C., & Peres, P. L. (2011). MPC for LPV systems with bounded parameter variations. *International Journal of Control*, 84(1), 24-36.

Xu, Z., Zhao, J., Qian, J., & Zhu, Y. (2009). Nonlinear MPC using an identified LPV model. *Industrial & Engineering Chemistry Research*, 48(6), 3043-3051.

Besselmann, T., & Morari, M. (2009, August). Autonomous vehicle steering using explicit LPV-MPC. In *2009 European Control Conference (ECC)* (pp. 2628-2633). IEEE.

Besselmann, T., Lofberg, J., & Morari, M. (2012). Explicit MPC for LPV systems: Stability and optimality. *IEEE Transactions on Automatic Control*, 57(9), 2322-2332.

Weinmann, A. (2012). Uncertain models and robust control. *Springer Science & Business Media.*

Kouvaritakis, B., & Cannon, M. (2016). Model predictive control. *Switzerland: Springer International Publishing.*

Mayne, D. (2016). Robust and stochastic model predictive control: Are we going in the right direction?. *Annual Reviews in Control*, *41*(184-192).

Liu, C., Li, H., Gao, J., & Xu, D. (2018). Robust self-triggered min–max model predictive control for discrete-time nonlinear systems. *Automatica*, 89(333-339).

Brunner, F. D., Heemels, M., & Allgöwer, F.(2016). Robust self-triggered MPC for constrained linear systems: A tube-based approach. *Automatica*, 72(73-83).

Sun, T., Pan, Y., Zhang, J., & Yu, H. (2018). Robust model predictive control for constrained continuous-time nonlinear systems. *International Journal of Control*, 91(2)(359-368).

Gonzalez, R., Fiacchini, M., Alamo, T., Guzmán, J. L., & Rodríguez, F. (2011). Online robust tube-based MPC for time-varying systems: A practical approach. *International Journal of Control*, 84(6)(1157-1170).

Sakhdari, B., Shahrivar, E. M., & Azad, N. L.(2017, October). Robust tube-based mpc for automotive adaptive cruise control design. *IEEE 20th International Conference on Intelligent Transportation Systems (ITSC)*,(1-6).

Darup, M. S., & Mönnigmann, M. (2018). Optimization-free robust MPC around the terminal region. *Automatica*, 95(229-235).

Kim, Y., Zhang, X., Guanetti, J., & Borrelli, F. (2018). Robust Model Predictive Control with Adjustable Uncertainty Sets. *2018 IEEE Conference on Decision and Control (CDC)*, (5176-5181).

McMullen, P. (1971). On zonotopes. *Transactions of the American Mathematical Society*, 15 (91-109).

Driverless UPC. (2019). *Driverless UPC Team (Spain, Barcelona).* Web: http://www.apa.org/monitor/

Formula Student. (2019). Formula Student Germany. Web: https://www.formulastudent.de/

Borrelli, F., Bemporad, A., & Morari, M. (2017). Predictive control for linear and hybrid systems. *Cambridge University Press.*

Tan, J., Olaru, S., Roman, M., Xu, F., & Liang, B. (2019). Invariant set-based analysis of minimal detectable fault for discrete-time LPV systems with bounded uncertainties. *IEEE Access*, 7, 152564-152575.

Girard, A. (2005). Reachability of uncertain linear systems using zonotopes. In *International Workshop on Hybrid Systems: Computation and Control* (pp. 291-305). Springer, Berlin, Heidelberg.

Girard, A., Le Guernic, C., & Maler, O. (2006, March). Efficient computation of reachable sets of linear time-invariant systems with inputs. In *International Workshop on Hybrid Systems: Computation and Control* (pp. 257-271). Springer, Berlin, Heidelberg.

Roborace. URL: https://roborace.com/.

Katrakazas, C., Quddus, M., Chen, W. H., & Deka, L. (2015). Real-time motion planning methods for autonomous on-road driving: State-of-the-art and future research directions. *Transportation Research Part C: Emerging Technologies*, 60, 416-442.

Paden, B., Čáp, M., Yong, S. Z., Yershov, D., & Frazzoli, E. (2016). A survey of motion planning and control techniques for self-driving urban vehicles. *IEEE Transactions on intelligent vehicles*, 1(1), 33-55.

Zhang, H., & Yang, S. (2018). Smooth path and velocity planning under 3D path constraints for car-like vehicles. *Robotics and Autonomous Systems*, 107, 87-99.

Talamino, J. P., & Sanfeliu, A. (2019). Anticipatory kinodynamic motion planner for computing the best path and velocity trajectory in autonomous driving. *Robotics and Autonomous Systems*, 114, 93-105.

González, D., Pérez, J., Milanés, V., & Nashashibi, F. (2015). A review of motion planning techniques for automated vehicles. *IEEE Transactions on Intelligent Transportation Systems*, 17(4), 1135-1145.

Liu, C., Lee, S., Varnhagen, S., & Tseng, H. E. (2017, June). Path planning for autonomous vehicles using model predictive control. In *Intelligent Vehicles Symposium (IV)*, 2017 IEEE (pp. 174-179).

Hegedüs, F., Bécsi, T., Aradi, S., & Gápár, P. (2017). Model based trajectory planning for highly automated road vehicles. *IFAC-PapersOnLine*, 50(1), 6958-6964.

Ahmadi Mousavi, M., Moshiri, B., & Heshmati, Z. (2018). Linear Time Varying MPC Based Path Planning of an Autonomous Vehicle via Convex Optimization. *Journal of Iranian Association of Electrical and Electronics Engineers*, 14(4), 79-88.

rifaee, B., & Maczijewski, J. (2018, June). Real-time Trajectory optimization for Autonomous Vehicle Racing using Sequential Linearization. In *2018 IEEE Intelligent Vehicles Symposium (IV)* (pp. 476-483). IEEE.

Verschueren, R., De Bruyne, S., Zanon, M., Frasch, J. V., & Diehl, M. (2014, December). Towards time-optimal race car driving using nonlinear MPC in real-time. In *53rd IEEE conference on decision and control* (pp. 2505-2510). IEEE.

Verschueren, R., Zanon, M., Quirynen, R., & Diehl, M. (2016, June). Time-optimal race car driving using an online exact hessian based nonlinear MPC algorithm. In *2016 European Control Conference (ECC)* (pp. 141-147). IEEE.

Plessen, M. G., Lima, P. F., Martensson, J., Bemporad, A., & Wahlberg, B. (2017, October). Trajectory planning under vehicle dimension constraints using sequential linear programming. In *2017 IEEE 20th International Conference on Intelligent Transportation Systems (ITSC)* (pp. 1-6). IEEE.

Liniger, A., Domahidi, A., & Morari, M. (2015). Optimization-based autonomous racing of 1: 43 scale RC cars. *Optimal Control Applications and Methods*, 36(5), 628-647.

Kwiatkowski, A., Boll, M. T., & Werner, H. (2006, December). Automated generation and assessment of affine LPV models. In *Decision and Control, 45th IEEE Conference* (pp. 6690-6695).

Wächter, Andreas, & Lorenz T. Biegler. (2006). On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming. *Mathematical programming* 106.1 (2006): 25-57.

Lofberg, Johan. YALMIP: A toolbox for modeling and optimization in MATLAB. *Computer Aided Control Systems Design, 2004 IEEE International Symposium* on (2004): IEEE 284-289.

Kissai, M., Monsuez, B., Tapus, A., & Martinez, D. (2018) Control Allocation of Active Rear Steering and Vehicle Dynamics Control Using a New Tire Model. *International Journal of Mechanical Engineering and Robotics Research.*

Liniger, A. (2018). Path planning and control for autonomous racing. ETH Zurich.