Treball de Fi de Màster

Master's degree in
Automatic Control and Robotics

LPV Control of an Autonomous Vehicle

MEMÒRIA

Autor: Oscar Martí Rubio
Director/s: Vicenç Puig Cayuela
Convocatòria: Juny 2016

Escola Tècnica Superior
d'Enginyeria Industrial de Barcelona

Abstract

There are four main procedures that an Autonomous car-like vehicle must perform in a real-time dynamic complex environment: Perception and Modelling, Localization and Map Building, Path Planning and Decision-Making, and Motion Control. The Motion Control procedure translates the decision taken by the Path Planning and Decision-Making module to specific commands for the different actuators of the vehicle. The Motion Control issue treated in this Thesis is the Path Tracking problem.

The goal of this Thesis is to approach the problem in the simplest possible way whilst being accurate and developing an intuitive design procedure. To achieve this goal, the bicycle-like kinematic model and a switched state feedback LPV controller, which has been designed using the polytopic transformation via nonlinear embedding and bounding box approach in order to perform the control of the system in an integrated way (lateral and longitudinal control at the once), have been chosen.

Consequently, two different controllers have been synthesized, compared and proven fully adequate to confront the Path Tracking Problem. Both controllers show very good results and seem very consistent in the non-conservative simulation tests performed.

The approach proposed in this Thesis is simple, intuitive and has produced successful simulation results.

Contents

ETSEIB

List of Figures

Chapter 1

Introduction

1.1 Context, motivation and objectives

Every day, we are becoming more familiar with autonomous vehicles. Autonomous drones or cars are good examples of them and these driving systems have been a very active field of research during the last decade. In particular, the interest of research institutions and car manufacturers in the development of autonomous car-like robots has boomed since 2004, when the DARPA agency held the Grand Challenges and the Urban Challenge. There are lots of practical applications for this technology behind the interest. In the case of autonomous car-like robots, which is the subject of the Thesis, the necessity of improving the security of pedestrians and passengers, reducing the energy used or travelling with better comfort conditions exists. These types of vehicles can also be used to get access into areas where humans are not able to due to environmental conditions, or even to explore other planets.

Mainly, as a generalization, an autonomous car-like vehicle or robot must perform four fundamental procedures in a real-time dynamic complex environment [5]: Perception and Modelling, Localization and Map building, Path Planning and Decision-Making, and Motion Control. They are shown schematically in Figure 8. The Perception and Modelling procedure is responsible for describing the surroundings of the vehicle, thus obtaining a model of it. This is done thanks to the set of sensors that the vehicle has. The Localization and Map Building procedure is the one that interprets the information from the sensors and estimates the position of the vehicle on a global map. The Path Planning and Decision-Making procedure is in charge of deciding what the next desired position is, according to some behavioural rules like: goal reaching, obstacle avoidance, safety, comfortability and others. Finally, the Motion Control procedure translates the decision taken by the path planning and Decision-Making module to specific commands for the different actuators of the vehicle, for instance the steering wheel or the throttle.

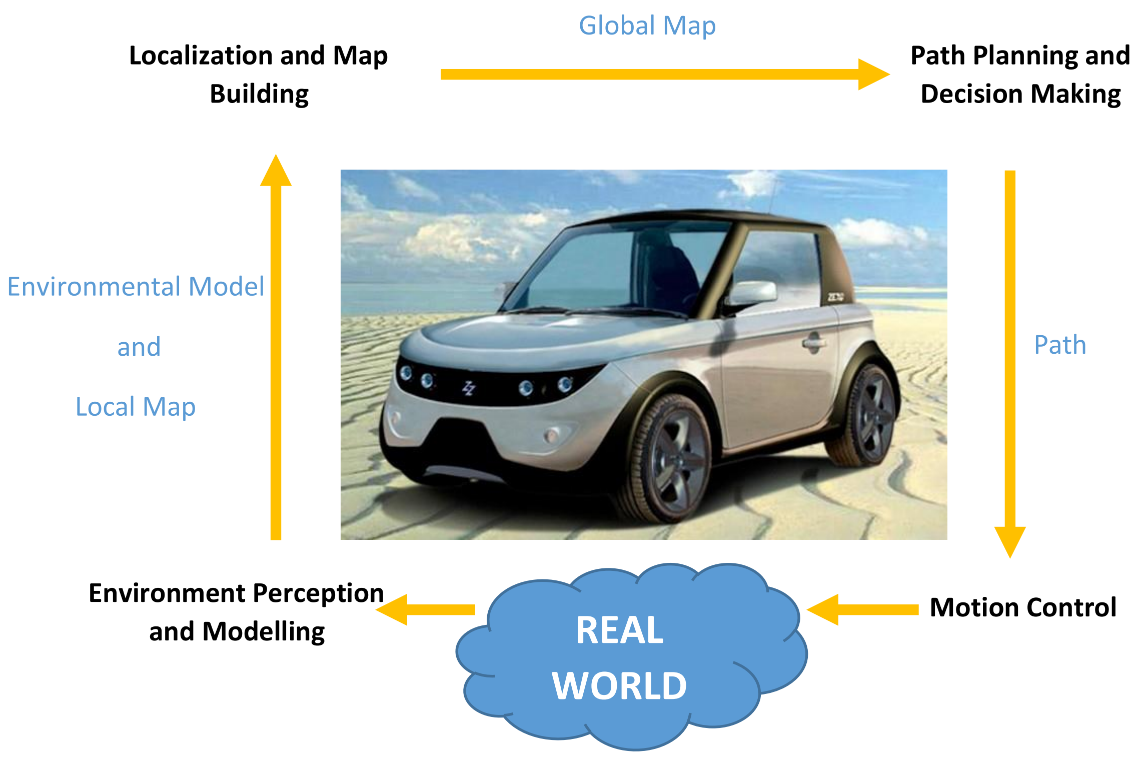
Localization and Map Building
Global Map
Path Planning and Decision Making
Environmental Model and Local Map
Path
Environment Perception and Modelling
REAL WORLD
Motion Control

Figure 1.1: Basic modules of an autonomous car-like vehicle

Each one of these modules implies a different line of research and investigation, and there are many ways of approaching them individually. This Thesis is focused on the Motion Control one. As previously stated, Motion Control is a translator. The main goals of it are: to follow the orders received from the Path Planning and Decision-Making module as well as possible, to produce accurate and smooth trajectories within a range of reaction times, and, to provide the level of required comfort. The motion orders give the next position the vehicle should be in or, equivalently, the velocity and the angle of the steering wheel in order to get to the desired next position. Therefore, two different controls are needed: a lateral motion control and a longitudinal motion control. These controls can be computed separately or integrated into a single one. Controllers that perform this kind of motion control are called path trackers. Path tracking refers to a vehicle executing a globally defined geometric path by applying appropriate steering motions or angular velocity (lateral control) and speed or linear velocity (longitudinal control) that guide the vehicle along the path [5]. This requires a feedback control of the error position, which is the difference between the reference and the real position of the vehicle. There are different parts in a feedback control system, namely: the reference path, the vehicle, sensors and the controller. The reference path is the one given by the Path Planning and Decision-Making module. The vehicle's behaviour can be reduced to a model with the aim of easing the computations. This model can be obtained basically from physical equations that describe the behaviour of the vehicle or obtaining it from experimental data. Many different models for a car are available: linear and nonlinear, kinematic and dynamic, simple ones (bicycle-like vehicle) or much more complex. The complexity of the model does not ensure better results. Sensors are needed in order to know the states and parameters of the vehicle necessary for the design of the controller. The controller is in charge of determining the actions to be performed by the vehicle actuators.

The design of the controller can be approached in many ways. It can be designed using linear techniques in a linearized model. In this case, the nonlinearities of the real behaviour of the vehicle are not taken into account, but, on the other hand, the very well-studied classic control methods can be applied. Several nonlinear methods that solve specific problems exist. The controller can also be designed by using them and, hence, taking into account a much more accurate representation of the vehicle [1], [2]. Another design possibility would be the use of linear like techniques (as e.g. the LPV approach) to address the nonlinear control problem. This allows the combination of the well-studied classic control methods with the accuracy of the representation of the nonlinear behaviour. Gain scheduling is a very popular approach that follows this last scheme. It is a collection of methods that tries to solve the problem in a divide and conquer manner [16]. It is based on the computation of a different control for each operating point, using a linearized model of the system at that point, and then, schedule the set of controllers depending on the operating point the system is at. The drawback of this approach is that it does not capture the non-local behaviour in the transition from one operating point to another. In order to overcome this problem, a Linear Parameter-Varying (LPV) approach was first introduced in [17]. The Linear Parameter-Varying computes a controller capable of automatically varying its gain, working with regions of operating points and addressing the nonlinear control problem. However, since this method follows a parametric approach, the complexity of the problem increases with the number of varying parameters. From the moment this methodology appeared, it has become an increasing field for research [9]. All in all, the number of possible approaches is very high and depends, mainly, on the chosen model and the applied technique.

In this Thesis, the goal is to approach the problem in the simplest possible way whilst being accurate and developing an intuitive design procedure. To achieve this goal, the bicycle-like kinematic model of the vehicle has been chosen because of its simplicity, and, the application of linear-like control techniques to the integrated control (lateral and longitudinal), thanks to its generality and wide range. Taking into account these two design decisions, the control method developed in this Thesis is the LPV control theory. The combined use of the bicycle-like model and the LPV technique has never been completely tried before. A very similar one was developed in [3] using the Takagi-Sugeno approach but the obtained control was an average of the controls that bound the area of operating points. This is the reason and motivation for the development of the approach proposed in this Thesis, including all the steps from the starting model, the bicycle-like model, to the control design, a switched LPV one. The proposed approach has been validated in simulation using Simulink and obtained very promising results in a wide range of reference variables.

1.2 Outline of the Thesis

This text is organized into six chapters. The outline of each chapter may be summarized as follows:

1.2.1 Chapter 2: Background Theory

This Chapter provides the sufficient background regarding the LPV control theory to be able to follow the rest of the Thesis development.

1.2.2 Chapter 3: Control-oriented model

The bicycle-like kinematic model of the vehicle and the error model are derived in this Chapter.

1.2.3 Chapter 4: Control Design

This Chapter shows the design of a switched state feedback LPV controller. The model formulation approach is the polytopic transformation via nonlinear embedding and bounding box approaches.

1.2.4 Chapter 5: Results

This Chapter presents and compares the results obtained during the simulation of the two controllers. The results are compared and analysed.

1.2.5 Chapter 6: Sustainability: environmental, social and economic impact

This Chapter discusses the various impacts that the introduction of autonomous cars could have on our lives.

1.2.6 Chapter 7: Conclusions and Future works

This Chapter states the most important outcome and the perspectives of this Thesis.

1.2.7 Chapter 8: Cost of the project

This Chapter presents the budget of the project costs.

Chapter 2

Background Theory

In this chapter, a revision of the background theory on LPV systems is presented for completeness. If the reader is familiar with the subject, this chapter could be omitted and be used only in case it is needed. The intention is to provide a sufficient background of LPV theory in order to be able to follow the rest of the Thesis.

2.1 Introduction

In practice, most of the existing systems that need to be controlled are nonlinear and they have to work in different operating points. Standard linear models, known as linear time invariants (LTI), are only valid around a given operating point. The extension to several operating points lead to the LPV systems. Focusing on the difference, a clue to the main idea of LPV systems is as follows: somehow an LPV system tries to convert nonlinear systems into linear parameter varying ones. LPV theory allows the use of a formulation which is very similar to the linear systems one, but it takes into account the nonlinearities. LPV was first introduced by [17] and it has been proven to be suitable for managing some nonlinear systems.

2.2 What is a Linear Parameter Variant system. Definition and Types.

Shamma made the distinction between LTI, LTV and LPV systems [17]. A particular type of linear time varying systems are LPV systems. In LPV systems, the time-varying elements depend on measurable parameters that can vary over time [16]. A more formal definition would be: an LPV system is a finite-dimensional LTV system whose state space matrices are fixed functions of some vector of varying, measurable parameters, assumed to be unknown a priori, but measured or estimated in real time [16].

A continuous state space LPV model is usually defined as:

$$\dot{x}(t) = A(\vartheta(t))x(t) + B(\vartheta(t))u(t) \tag{2.1}$$

$$y(t) = C(\vartheta(t))x(t) + D(\vartheta(t))u(t) \tag{2.2}$$

where $x \in R^{n_x}$, $u \in u^{n_u}$ and $y \in R^{n_x}$ are the state, the input and the output vector. The varying matrices are $A(\vartheta(t))$, $B(\vartheta(t))$, $C(\vartheta(t))$ and $D(\vartheta(t))$.

2.2.1 Types

LPV systems can be classified in two different categories: Pure LPV systems and Quasi-LPV systems. The difference between them is the nature of the signal that is used to describe the parameter variation.

Pure LPV: This is an LPV system where the varying parameters only depend on exogenous signals, such as noise or disturbances.

Quasi-LPV: This is an LPV system where the varying parameters depend on some endogenous signals, such as the states, inputs or outputs. Quasi-LPV systems have proven to be appropriate for controlling nonlinear systems by embedding the nonlinearities in the varying parameters.

2.3 What is not a Linear Parameter Variant system

The LPV control approach has some similarities with others.. It is as important to know what an LPV system is as to know what it is not.

- Adaptive Control: LPV control is similar because both share the concept of having a controller that adapts to the varying parameters of the system. However, adaptive control requires a persistent excitation of the inputs to be able to adapt the parameters of the controller while LPV does not.

- Classic Gain Scheduling: LPV control is similar because both techniques use different controllers for each operating point of the system. The main difference is that, in classic

Gain Scheduling, the control is designed for a finite number of operating points, while, in the LPV approach, the control, is designed for a set of operating points guaranteeing stability and performance.

- Robust Control: LPV control is different from robust control. A parameter in an LPV system is measurable in real time while, in robust control, parametric uncertainty in an uncertain system is not measurable. LPV can be used to design robust controllers if the system with the uncertainty can be transformed into an LPV system with unmeasurable parameters.
- Takagi-Sugeno : LPV systems and Takagi and Sugeno systems (TS) are equivalent. They are equivalent approaches with different origins. While LPV comes from the field of Control Systems, TS originates from the field of Artificial Intelligence.

2.4 Formulation of an LPV description

A model of the dynamical system is required in order to apply a model based technique. Two main ways of obtaining such descriptions exist: analytical and experimental. The experimental way is based on input/output data. The identification of the model is done carried out by obtaining it from the data (an overview of different methods can be found in [4]). The analytical way is based on equations which describe the behaviour of the system. The interest in obtaining LPV models has constantly increased over recent years due to the popularity of LPV [9].

The same nonlinear system can be represented by an LPV model in more than one way. Therefore, LPV representations are not unique. This is good and bad at the same time. The LPV representation can be transformed to convenience until obtaining a shape that suits for controller synthesis purposes. On the other hand, not every LPV description is appropriate for control.

2.4.1 Properties of the LPV descriptions

LPV descriptions have some properties that could be summarized as follows, [16]:

1. Parameters and states are related in a way such that this relation is equal in the LPV description and in the nonlinear system. Thus, trajectories of the nonlinear system are also trajectories of the LPV description. The LPV description includes the trajectories of the nonlinear system.

2. The relation between parameters and states depends only on measured signals. This property comes from the definition of LPV systems and ensures that the controller can depend on the parameter. The variable parameters are assumed to be measurable or depend only on measured signals.

3. The relation between parameters and states is known.

4. The LPV description includes the nonlinear system. This fact introduces some conservativeness, the more similar the LPV description and the nonlinear system are, the less conservativeness is introduced and a better performance can be obtained. LPV descriptions that are conservative can be useful when robustness is a design requirement.

2.4.2 LPV model formulation approaches

Different approaches to formulate LPV models exist. In this section, only the main ones are described, and the one used in this Thesis, nonlinear embedding plus polytopic transformation using the bounding box approach, is hereby explained in more detail.

Polytopic transformation via nonlinear embedding and bounding box approach

The idea behind this approach is to embed the nonlinearities of the system in the varying parameters. By doing this, the control of the system can be synthesized using an extension of linear techniques.

The procedure to obtain a polytopic representation of an LPV, often referred to as bounding box method, is described here:

- The nonlinearities must be embedded in $\vartheta(t)$.

- The variation range of each one of the parameters must be known or measurable. Therefore, the variation bounds are the maximum and minimum value of each parameter $\vartheta(t)$.

- The polytope is defined by the variation bounds of the parameters and describes a simple shape area, which contains the whole nonlinear system and, therefore, all its operating points. Describing the system only with the vertex points of the polytope is very useful at the time of performing different computations. Only the vertex points are computed, which signifies a huge reduction of the number of points to compute or equivalently the number of constraints. The infinite number of points of the nonlinear system have been reduced to only the vertex points of the polytope. This procedure introduces some conservativeness because the area of the bounding box contains other possible operating points other than the nonlinear system ones. The application of the procedure becomes more complicated as the number of varying parameters increases as well. The number of vertex points is two to the power of the number of parameters. Another drawback is that any singular points inside the polytope are not considered.

Formally, an LPV polytopic system is described by the following equations:

$$\dot{x}(t) = \sum_{i=1}^{N} \mu_i(\vartheta(t))(A_i x(t) + B_i u(t)) \tag{2.3}$$

$$y(t) = \sum_{i=1}^{N} \mu_i(\vartheta(t))(C_i x(t) + D_i u(t)) \tag{2.4}$$

where A_i, B_i, C_i and D_i define the vertex systems. There is a LTI model for each vertex. μ_i is the scheduling function or weights. The varying parameters and weights fulfill the following condition:

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

where ϑ is the polytope or bounding box.

Jacobian Linearization

Jacobian Linearization is the simplest methodology to formulate LPV models. It is useful for nonlinear models that can be linearized around its equilibrium points of interest [13]. These linearizations join together to form a family of linearized models. The resulting LPV model, the whole family, is a local approximation of the dynamics of the nonlinear system around its equilibrium points of interest. The approximation of the nonlinear system is computed using Taylor-series, either first-order or higher. First order Taylor expansion could lead to a model that might diverge from the behaviour of the nonlinear system [13], while higher-order Taylor expansion could lead to impractical implementations [10].

State transformation

This technique is only useful when the system has the following form:

$$\begin{bmatrix}\dot{z}(t)\\ \dot{l}(t)\end{bmatrix} = g(z(t)) + A(z(t)) \begin{bmatrix}z(t)\\ l(t)\end{bmatrix} + B(z(t))u(t) \tag{2.6}$$

where $z(t) \in R^{n_z}$ are the scheduling states, and $l(t) \in R^{n_h}$ are the non-scheduling ones with $n_z = n_u$.

In the state transformation approach, a coordinate change is searched for being able to remove the nonlinear part from some measured states. Any nonlinear term that is not dependent on the scheduling parameters is aimed at being removed [18]. This approach applies a transformation and not an approximation as in the Jacobian Linearization one. The Quasi-LPV model obtained exactly represents the original nonlinear system.

2.5 Synthesis of an LPV controller

Once the nonlinear system is described as an LPV one, the next step is to synthesize the controller. As previously mentioned, the polytopic formulation has been the one chosen in this Thesis. In summary, a polytopic representation approximates the parameter set by its vertices, whose position depends on the varying parameter bounds, and contains all the operating points of the nonlinear system.

Basically, two steps need to be performed: to compute a controller for each vertex of the polytope and to map any operating point inside the bounding box with a weight for each vertex controller. By doing this, a nonlinear controller for the whole bounding box is synthesized.

Controller for each vertex

The system is considered as an LTI in the bounding box vertices. Hence, extensions of linear techniques can be applied there. A linear controller is computed for each vertex of the bounding box taking into account all the constraints imposed by the system itself or design requirements and decisions.

In this Thesis, the resulting nonlinear error model is brought to a Quasi-LPV form suitable for designing an LPV controller by solving a system of linear matrix inequalities (LMIs), a problem for which efficient solvers are available [20], [11]. Many optimization problems in control theory, system identification and signal processing can be formulated using LMIs. Therefore, LMIs are a very powerful mathematical tool. Basically, the system to control is translated, if possible, into a set of LMIs that can be solved. The solution of an LMI can be interpreted as the intersection of several inequalities, which could represent, for instance, design requirement constraints, as placing the poles in a certain region or only allowing decay ratios higher than a value.

Formally, an LMI has the form [6]:

$$F(x) \overset{\text{def}}{=} F_0(x) + \sum_{i=1}^{m} x_i F_i > 0 \tag{2.7}$$

where $x \in R^m$ is the variable and the symmetric matrices $F_i = F_i^T \in R^{nxn}$, $i = 0, ..., m$, are given and $F(x)$ is positive definite.

Specific software that is capable of dealing with LMIs is available. YALMIP is the one used to solve the different sets of LMIs in this Thesis. YALMIP is a modelling language for advanced modelling and solution of convex and nonconvex optimization problems and it is available for free at [12].

The LMIs used in this Thesis are the appropriate ones to solve for a state feedback control with the decay ratio as a design parameter and using the polytopic transformation via nonlinear embedding and bounding box approach. The use of the decay ratio as the design parameter implies placing all the poles of the controlled system at the hyperplane defined on the left side of a desired real value in the complex plane. The LMIs are the following:

$$(A(\vartheta) + BK)P + P(A(\vartheta) + BK)^t + 2\alpha P < 0 \ , \ \ P > 0 \qquad \forall \ \vartheta P > 0 \tag{2.8}$$

where A is the state matrix, B is the input matrix, ϑ are the varying parameters of the system, α is the desired value that defines the right boundary of the hyperplane. K is the state feedback controller that is computed for each edge of the bounding box. The bounding box is defined by the boundaries of the parameter region of variation. And P is a Lyapunov function that, if it exists, ensures feasibility.

Each LMI of the set of LMIs expressed in (2.8) should be solved for P and for K simultaneously. This is not possible because the problem is nonlinear. A variable change is made in order to overcome this issue:

$$K = WP^{-1} \tag{2.9}$$

Therefore, the LMIs to solve have the following form:

$$A(\vartheta)P + BW + PA(\vartheta)^t + W^tB^t + 2\alpha P < 0 \ , \ \ P > 0 \qquad \forall\, \vartheta P > 0 \tag{2.10}$$

Mapping operating points with vertex controllers

Once the controller of each vertex has been obtained, the next step is to find a way of mapping an operating point inside the bounding box with a controller. In order to do this, membership functions are defined following an interpolation rule.

Membership functions express the degree of membership of each vertex controller with any operating point inside the bounding box. The interpolation rule ensures that the controllers obtained, as a result of applying the membership functions, are inside a bounding box of controllers defined by all vertex controllers.

Membership functions are computed using the boundaries of each varying parameter of the system. At least, there are as many functions as varying parameters. The value of a membership function must be in the range [0 1]. An example of a membership function is as follows:

$$\mu_i(\vartheta_i) = \frac{\vartheta_{i_{max}} - \vartheta_i}{\vartheta_{i_{max}} - \vartheta_{i_{min}}} \tag{2.11}$$

where i refers to the parameters and μ is the membership function. In the case of the example, μ_i is 1 if $\vartheta_i = \vartheta_{i_{min}}$.

The interpolation rule that constraints the membership functions is:

$$\sum_{i=1}^{N} \mu_i(\vartheta(t)) = 1 \tag{2.12}$$

2.6 Formulataion of the nonlinear controller

To summarize until here, the nonlinear system trajectories are contained in an LPV description. In this Thesis, this description has been obtained by using the polytopic formulation. Then, controllers for each vertex of the bounding box are computed using LMIs as a mathematical tool. Finally, all operating points are mapped to a weighted controller, using the vertex controllers and some membership function and a rule of interpolation.

The nonlinear controller is the one that commutes its gain according to the states and the parameters of the nonlinear system. The final gain of this LPV controller is computed as the addition of the gain of each control vertex, taking into account its membership degree

$$K(\vartheta(t)) = \sum_{i=1}^{N} \mu_i(\vartheta(t))K_i \tag{2.13}$$

where K is the nonlinear controller, K_i is the controller of the i-th vertex and μ_i is the scheduling rule (named weights or membership degree, as well).

2.7 Switched LPV

When there are both continuous-valued and discrete-valued varying parameters, the resulting system is referred to as a switching LPV [8], [15] .

Sometimes, singular points inside the bounding box appear when using the polytopic formulation approach to reduce the number of constraints from infinite to finite. This causes unfeasibility of the set of LMIs. In order to solve this problem, the system can be divided into subsystems that do not contain the singular point and where the subsets of LMIs are feasible. Thus, a virtual switching component does the job of changing between subsystems according to the value of the switching variable. In this case, the system is referred to as switching LPV, where there are both, continuous-valued and discrete-valued varying parameters [8].

Formally, this kind of control systems are expressed as follows:

$$\dot{x}(t) = A_\sigma(\vartheta(t))x(t) + B_\sigma(\vartheta(t))u(t) \tag{2.14}$$

where σ is the discrete switching variable.

If the controller is a state feedback one, then it can be expressed as

$$u(t) = K_\sigma(\vartheta(t))x(t) \tag{2.15}$$

Chapter 3

Control-oriented model

In this Chapter, the model of the vehicle is presented. The starting point is the kinematic model of a bicycle, widely used in the design of controllers for autonomous systems. A complete derivation of it is shown. Afterwards, its adaptation to a model that is suitable for solving the trajectory tracking problem, which is the model of the error posture, is developed.

3.1 Model of the car

Several models for a car exist [7]. The car is considered as a rigid body on wheels that operates on a horizontal plane. In the case of this Thesis, one of the goals is to have, as starting point, the simplest possible model. Therefore, the kinematic model of a bicycle has been chosen. This simplification of the model is a very common one. The kinematic model of the mechanical structure of a car describes the motion with respect to a fixed reference Cartesian frame. The causes of this motion, forces and moments, are ignored in the kinematic model.

In the kinematic model of a bicycle, it is assumed that the left and right wheels collapse into only one wheel, leaving only the front and rear wheel. Three assumptions are made here for the motion: wheels do not present lateral slip, only the front wheel is steerable and the whole motion of the vehicle happens on a plane.

The equations that describe the kinematic system are:

$$\dot{x} = v\cos(\theta) \tag{3.1}$$

$$\dot{y} = v\sin(\theta) \tag{3.2}$$

$$\dot{\theta} = \frac{v}{L}\tan(\delta) = w \tag{3.3}$$

where v is the linear velocity, x and y are the rear wheels position and θ is the the orientation of the vehicle, all in the world (or global) frame.

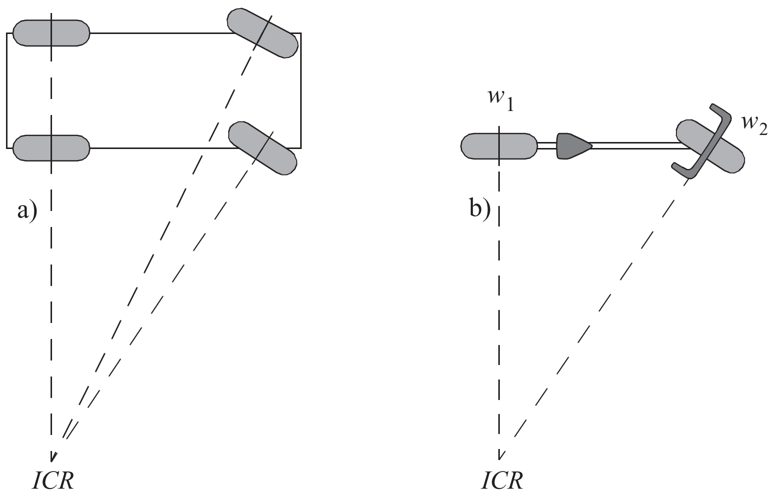

Figure 3.1: From car, a), to bicycle model b)

3.2 Derivation of the kinematic model

The equations of motion for the kinematic bicycle model are derived here for completeness.

There are some simple geometric relationships between the different parts of the bicycle model and the steering angle

$$\tan(\delta) = \frac{L}{R} \tag{3.4}$$

where δ is the steering angle, L is the distance between front and rear wheel and R is the radius of the turning curve. The geometric vehicle model is shown in the Figure 3.2.

All wheeled vehicles are subject to kinematic constraints. These constraints reduce the mobility of the vehicle. In the bicycle model, the nonholonomic constraint equations for the front and rear wheels are:

$$\dot{x}_f \sin(\theta + \delta) - \dot{y}_f \cos(\theta + \delta) = 0 \tag{3.5}$$

$$\dot{x} \sin(\theta) - \dot{y} \cos(\theta) = 0 \tag{3.6}$$

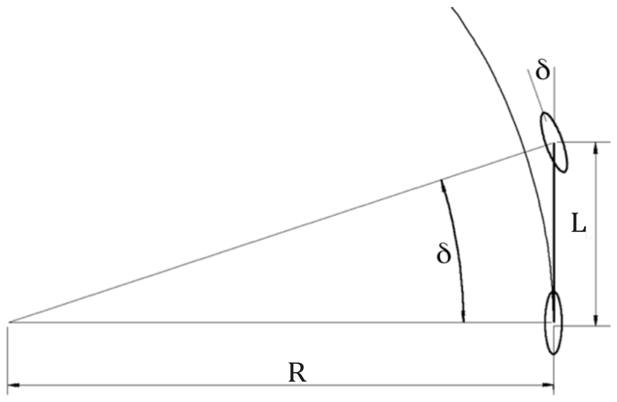

Figure 3.2: Geometric vehicle model

where (x_f, y_f) is the global coordinate of the front wheel. The kinematic model is shown in Figure 3.1. As the front wheel is located at distance L from the rear wheel along the orientation of the vehicle, (x_f, y_f) may be expressed as:

$$x_f = x + L\cos(\theta) \tag{3.7}$$

$$y_f = x + L\sin(\theta) \tag{3.8}$$

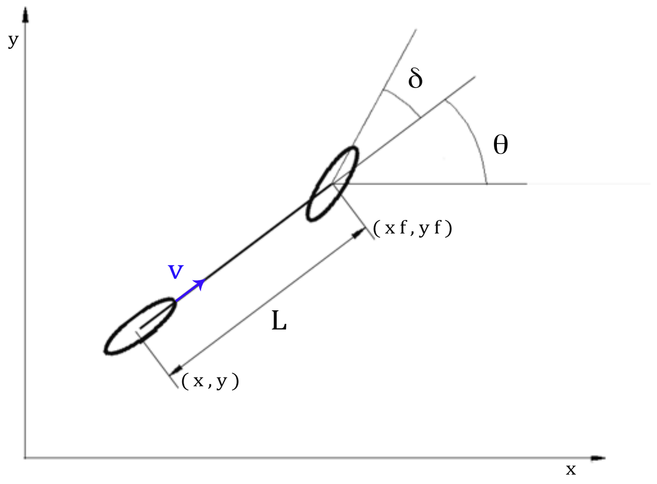

Figure 3.3: Kinematic vehicle model

3.2.1 Derivation of the Linear velocity equations

The linear velocity equations of the kinematic model, (3.1) and (3.2), have to fulfill the nonholomonic constraints. The demonstration is very straightforward. The equations are obtained just by looking at the kinematic model shown in Figure 3.3 and applying basic trigonometry. The nonholomonic constraint for the rear wheel, (3.6), is satisfied when $\dot{x}\sin(\theta) = \dot{y}\cos(\theta)$, and also any scalar multiple thereof. This scalar corresponds to the longitudinal velocity in the car frame. Therefore, (3.1) and (3.2) fulfill the imposed constraint.

3.2.2 Derivation of the Angular velocity equation

The angular velocity equation of the kinematic model is (3.3). It can be derived from the front wheel equations, (3.7) and (3.8), the nonholomonic constraint for the front wheels, (3.5), and the linear velocity equations of the kinematic model, (3.1) and (3.2).

The derivation is the following:

- x_f and y_f from (3.7) and (3.8) are replaced in (3.5):

$$\begin{aligned}
0 &= \frac{\partial(x + L\cos(\theta))}{\partial t} sin(\theta+\delta) - \frac{\partial(x + L\sin(\theta))}{\partial t} cos(\theta+\delta) \\
&= (\dot{x} - \dot{\theta}L\sin(\theta)\sin(\theta+\delta)) - (\dot{y} + \dot{\theta}L\cos(\theta)\cos(\theta+\delta)) \\
&= \dot{x}\sin(\theta+\delta) - \dot{y}\cos(\theta+\delta) - \dot{\theta}L\sin(\theta)(\sin(\theta)\cos(\delta) + \cos(\theta)\sin(\delta)) \\
&- \dot{\theta}L\cos(\theta)(\cos(\theta)\cos(\delta) - \sin(\theta)\sin(\delta)) \\
&= \dot{x}\sin(\theta+\delta) - \dot{y}\cos(\theta+\delta) - \dot{\theta}L\sin^2(\theta)\cos(\delta) - \dot{\theta}L\cos^2(\theta)\cos(\delta) \\
&- \dot{\theta}L\sin(\theta)\cos(\theta)\cos(\delta) + \dot{\theta}L\cos(\theta)\sin(\theta)\sin(\delta) \\
&= \dot{x}\sin(\theta+\delta) - \dot{y}\cos(\theta+\delta) - \dot{\theta}L(\sin^2(\theta) + \cos^2(\theta))\cos(\delta) \\
&= \dot{x}\sin(\theta+\delta) - \dot{y}\cos(\theta+\delta) - \dot{\theta}L\cos(\delta)
\end{aligned}$$

- Using the last equality, $\dot{\theta}$ is obtained.

$$\dot{\theta} = \frac{\dot{x}\sin(\theta+\delta) - \dot{y}\cos(\theta+\delta)}{L\cos(\delta)}$$

- $\dot{x}$ and $\dot{y}$ are substituted by (3.1) and (3.2). A short development, shown here, will lead to the expected result (3.3).

$$\begin{aligned}\dot{\theta} &= \frac{\dot{x}\sin(\theta+\delta) - \dot{y}\cos(\theta+\delta)}{L\cos(\delta)} = \frac{v\cos(\theta)\sin(\theta+\delta) - v\sin(\theta)\cos(\theta+\delta)}{L\cos(\delta)} \\ &= \frac{v\cos(\theta)(\sin(\theta)\cos(\delta) + \cos(\theta)\sin(\delta)}{L\cos(\delta))} - \frac{v\sin(\theta)(\cos(\theta)\cos(\delta) - \sin(\theta)\sin(\delta)}{L\cos(\delta))} \\ &= \frac{v(\cos^2(\theta) + \sin^2(\theta))\sin(\delta)}{L\cos(\delta))} = \frac{v\tan(\delta)}{L}\end{aligned}$$

The obtained result for the angular velocity is (3.3)

$$\dot{\theta} = \frac{v}{L}\tan(\delta)$$

3.3 Error model

The error model expresses the dynamics of the difference between reference states and the system model states, which is a posture error.

3.3.1 Error posture

Until here, only the equations of the model of the car have been derived. A reference model is needed in order to be able to compute a control for tracking trajectories. The idea behind this is to compare the posture of the real car with a virtual one and, then, obtain the error of the posture. Figure 3.4 shows this idea.

The vehicle, in the world reference frame, has three degrees of freedom in its positioning that represent a posture

$$posture = \begin{bmatrix} x \\ y \\ \theta \end{bmatrix}$$

The heading direction θ is taken counterclockwise from the $x - axis$. A null posture, 0^o, will happen when $(0, 0, n\pi)^t$, where n is an integer. A set of feasible positions of $x(t)$ and $y(t)$ provides a path trajectory. Therefore, if $\dot{x}$ and $\dot{y}$ exist, $\theta(t)$ is not independent anymore. This is expressed as follows

$$\theta(t) = \tan^{-1}\frac{\dot{y}(t)}{\dot{x}(t)} \tag{3.9}$$

As previously mentioned, the real vehicle and a virtual one are used to obtain the posture of the error. The equations of the real and virtual car can be expressed as follows:

- Reference or virtual car posture, $p_{ref} = (x_{ref}, y_{ref}, \theta_{ref})^t$:

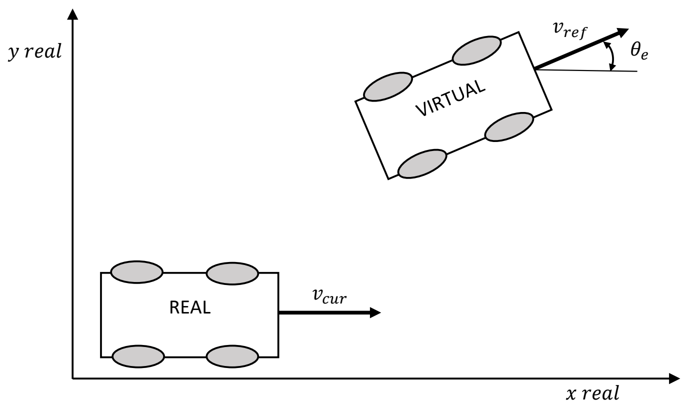

Figure 3.4: Real and virtual car

$$\dot{x}_{ref} = v_{ref}\cos(\theta_{ref}) \tag{3.10}$$

$$\dot{y}_{ref} = v_{ref}\sin(\theta_{ref}) \tag{3.11}$$

$$\dot{\theta}_{ref} = \frac{v_{ref}}{L}\tan(\delta_{ref}) = w_{ref} \tag{3.12}$$

– Real car posture, $p_{cur} = (x_{cur}, y_{cur}, \theta_{cur})^t$:

$$\dot{x}_{cur} = v_{cur}\cos(\theta_{cur}) \tag{3.13}$$

$$\dot{y}_{cur} = v_{cur}\sin(\theta_{cur}) \tag{3.14}$$

$$\dot{\theta}_{cur} = \frac{v_{cur}}{L}\tan(\delta_{cur}) = w_{cur} \tag{3.15}$$

The reference posture is the goal posture of the real vehicle. Therefore, the goal of the control will be reducing the difference between the virtual and the real vehicle to zero. This implies an error posture, $p_e = p_{ref} - p_{cur}$, and a control that will aim:

$$\lim_{t\to\infty}(p_{ref}(t) - p_{cur}(t)) = 0$$

.

It is convenient to use the model of the error with respect to the real robot frame due to the use of linear velocities with respect to the car. By doing so, subsequent computations simplify.

It is necessary to change frames, from world to car. The rotation matrix is used to do this job.

$$\begin{aligned} p_e = \begin{bmatrix} x_e \\ y_e \\ \theta_e \end{bmatrix} &= \begin{bmatrix} \cos(\theta_{cur}) & \sin(\theta_{cur}) & 0 \\ -\sin(\theta_{cur}) & \cos(\theta_{cur}) & 0 \\ 0 & 0 & 0 \end{bmatrix} (p_{ref} - p_{cur}) \\ &= \begin{bmatrix} \cos(\theta_{cur}) & \sin(\theta_{cur}) & 0 \\ -\sin(\theta_{cur}) & \cos(\theta_{cur}) & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} x_{ref} - x_{cur} \\ y_{ref} - y_{cur} \\ \theta_{ref} - \theta_{cur} \end{bmatrix} \end{aligned} \tag{3.16}$$

The error posture is obtained extending the previous matrix in equations for each state:

$$x_e = (x_{ref} - x_{cur})\cos(\theta_{cur}) + (y_{ref} - y_{cur})\sin(\theta_{cur}) \tag{3.17}$$

$$y_e = -(x_{ref} - x_{cur})\sin(\theta_{cur}) + (y_{ref} - y_{cur})\cos(\theta_{cur}) \tag{3.18}$$

$$\theta_e = \theta_{ref} - \theta_{cur} \tag{3.19}$$

3.3.2 Derivation of the error model

The dynamics of the error posture are what is needed for the trajectory tracking problem. Hence, (3.17), (3.18) and (3.19) need 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 (3.17).

$$\begin{aligned} \dot{x}_e &= (\dot{x}_{ref} - \dot{x}_{cur})\cos(\theta_{cur}) + (\dot{y}_{ref} - \dot{y}_{cur})\sin(\theta_{cur}) \\ &- (x_{ref} - x_{cur})\dot{\theta}_{cur}\sin(\theta_{cur}) + (y_{ref} - y_{cur})\dot{\theta}_{cur}\cos(\theta_{cur}) \\ &= \dot{x}_{ref}\cos(\theta_{cur}) - \dot{x}_{cur}\cos(\theta_{cur}) + \dot{y}_{ref}\sin(\theta_{cur}) - \dot{y}_{cur}\sin(\theta_{cur}) \\ &- (x_{ref} - x_{cur})\dot{\theta}_{cur}\sin(\theta_{cur}) + (y_{ref} - y_{cur})\dot{\theta}_{cur}\cos(\theta_{cur}) \end{aligned}$$

It is known that $\dot{\theta}_{cur} = w_{cur}$ from (3.15). Then

$$\begin{aligned} \dot{x}_e &= \dot{x}_{ref}\cos(\theta_{cur}) - \dot{x}_{cur}\cos(\theta_{cur}) + \dot{y}_{ref}\sin(\theta_{cur}) - \dot{y}_{cur}\sin(\theta_{cur}) \\ &- (x_{ref} - x_{cur})w_{cur}\sin(\theta_{cur}) + (y_{ref} - y_{cur})w_{cur}\cos(\theta_{cur}) \end{aligned}$$

From (3.18), we know that $y_e = -(x_{ref} - x_{cur})\sin(\theta_{cur}) + (y_{ref} - y_{cur})\cos(\theta_{cur})$ which appears in the previous equality. Hence

$$\dot{x}_e = \dot{x}_{ref}\cos(\theta_{cur}) - \dot{x}_{cur}\cos(\theta_{cur}) + \dot{y}_{ref}\sin(\theta_{cur}) - \dot{y}_{cur}\sin(\theta_{cur}) + w_{cur}y_e$$

The negative terms of the previous equality, $-\dot{x}_{cur}\cos(\theta_{cur})$ and $-\dot{y}_{cur}\sin(\theta_{cur})$, can be developed using the equations of the model of the car, (3.13), (3.14)

$$\begin{aligned}\dot{x}_{cur}\cos(\theta_{cur}) + \dot{y}_{cur}\sin(\theta_{cur}) &= v_{cur}\cos(\theta_{cur})\cos(\theta_{cur}) + v_{cur}\sin(\theta_{cur})\sin(\theta_{cur}) \\ &= v_{cur}(\sin^2(\theta_{cur}) + \cos^2(\theta_{cur})) = v_{cur}\end{aligned}$$

Now, using this result

$$\dot{x}_e = \dot{x}_{ref}\cos(\theta_{cur}) - v_{cur} + \dot{y}_{ref}\sin(\theta_{cur}) + w_{cur}y_e$$

By definition: $\theta_e = \theta_{ref} - \theta_{cur}$, then $\theta_{cur} = \theta_{ref} - \theta_e$. Replacing θ_{cur} in $\dot{x}_e$

$$\dot{x}_e = \dot{x}_{ref}\cos(\theta_{ref} - \theta_e) - v_{cur} + \dot{y}_{ref}\sin(\theta_{ref} - \theta_e) + w_{cur}y_e$$

The trigonometric identities for $\cos(\alpha - \beta)$ y $\sin(\alpha - \beta)$ are used in the next step

$$\begin{aligned}\dot{x}_e &= \dot{x}_{ref}(\cos(\theta_{ref})\cos(\theta_e) + \sin(\theta_{ref})\sin(\theta_e)) - v_{cur} \\ &\quad + \dot{y}_{ref}(\sin(\theta_{ref})\cos(\theta_e) - \cos(\theta_{ref})\sin(\theta_e)) + w_{cur}y_e \\ &= w_{cur}y_e - v_{cur} + (\dot{x}_{ref}\cos(\theta_{ref}) + \dot{y}_{ref}\sin(\theta_{ref}))\cos(\theta_e) \\ &\quad + (\dot{x}_{ref}\sin(\theta_{ref}) - \dot{y}_{ref}\cos(\theta_{ref}))\cos(\theta_e)\end{aligned}$$

The nonholomonic constraint for the real wheels is: $\dot{x}_{ref}\sin(\theta_{ref}) = \dot{y}_{ref}\cos(\theta_{ref})$, according to (3.6). Therefore

$$\dot{x}_e = w_{cur}y_e - v_{cur} + (\dot{x}_{ref}\cos(\theta_{ref}) + \dot{y}_{ref}\sin(\theta_{ref}))\cos(\theta_e)$$

Following the same procedure that was used before, the terms inside the parenthesis become

$$\begin{aligned}\dot{x}_{ref}\cos(\theta_{ref}) + \dot{y}_{ref}\sin(\theta_{ref}) &= v_{ref}\cos(\theta_{ref})\cos(\theta_{ref}) + v_{ref}\sin(\theta_{ref})\sin(\theta_{ref}) \\ &= v_{ref}(\sin^2(\theta_{ref}) + \cos^2(\theta_{ref})) = v_{ref}\end{aligned}$$

Finally, using the previous equality, the result for $\dot{x}_e$ is

$$\dot{x}_e = w_{cur} y_e - v_{cur} + v_{ref} \cos(\theta_e) \tag{3.20}$$

- **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 (3.18).

$$\begin{aligned}\dot{y}_e &= -(\dot{x}_{ref} - \dot{x}_{cur}) \sin(\theta_{cur}) + (\dot{y}_{ref} - \dot{y}_{cur}) \cos(\theta_{cur}) \\ &\quad - (x_{ref} - x_{cur})\dot{\theta}_{cur} \cos(\theta_{cur}) - (y_{ref} - y_{cur})\dot{\theta}_{cur} \sin(\theta_{cur})\end{aligned}$$

We had $x_e = (x_{ref} - x_{cur}) \cos(\theta_{cur}) + (y_{ref} - y_{cur}) \sin(\theta_{cur})$ from (3.17) which appears in the previous equality. We also had $\dot{\theta}_{cur} = w_{cur}$ from (3.15). Hence

$$\begin{aligned}\dot{y}_e &= -(\dot{x}_{ref} - \dot{x}_{cur}) \sin(\theta_{cur}) + (\dot{y}_{ref} - \dot{y}_{cur}) \cos(\theta_{cur}) - x_e w_{cur} \\ &= -x_e w_{cur} + \dot{x}_{cur} \sin(\theta_{cur}) - \dot{y}_{cur} \cos(\theta_{cur}) - \dot{x}_{ref} \sin(\theta_{cur}) - \dot{y}_{ref} \cos(\theta_{cur})\end{aligned}$$

The nonholonomic constraints for the rear wheels is: $\dot{x}_{cur} \sin(\theta_{cur}) = \dot{y}_{cur} \cos(\theta_{cur})$. Therefore

$$\dot{y}_e = -x_e w_{cur} - \dot{x}_{ref} \sin(\theta_{cur}) - \dot{y}_{ref} \cos(\theta_c ur)$$

The error in θ is $\theta_e = \theta_{ref} - \theta_{cur}$, then $\theta_{cur} = \theta_{ref} - \theta_e$. Replacing θ_{cur} in the previous equation

$$\dot{y}_e = -x_e w_{cur} - \dot{x}_{ref} \sin(\theta_{ref} - \theta_e) - \dot{y}_{ref} \cos(\theta_{ref} - \theta_e)$$

The trigonometric identities for $\cos(\alpha - \beta)$ y $\sin(\alpha - \beta)$ are used in the next step

$$\begin{aligned}\dot{y}_e &= -\dot{x}_e w_{cur} - \dot{x}_{ref}(\sin(\theta_{ref}) \cos(\theta_e) - \cos(\theta_{ref}) \sin(\theta_e)) \\ &\quad - \dot{y}_{ref}(\cos(\theta_{ref}) \cos(\theta_e) - \sin(\theta_{ref}) \sin(\theta_e)) \\ &= -\dot{x}_e w_{cur} + (\dot{x}_{ref} \cos(\theta_{ref}) + \dot{y}_{ref} \sin(\theta_{ref})) \sin(\theta_e) \\ &\quad + (\dot{y}_{ref} \cos(\theta_{ref}) - \dot{x}_{ref} \sin(\theta_{ref})) \cos(\theta_e)\end{aligned}$$

The same nonholonomic constraint is fulfilled for the reference car: $\dot{x}_{ref} \sin(\theta_{ref}) = \dot{y}_{ref} \cos(\theta_{ref})$.

Using this constraint in $\dot{y}_e$

$$\dot{y}_e = -x_e w_{cur} + (\dot{x}_{ref}\cos(\theta_{ref}) + \dot{y}_{ref}\sin(\theta_{ref})\sin(\theta_e)$$

Following the same procedure that was used before, the terms inside the parenthesis become

$$\begin{aligned}\dot{x}_{ref}\cos(\theta_{ref}) + \dot{y}_{ref}\sin(\theta_{ref}) &= v_{ref}\cos(\theta_{ref})\cos(\theta_{ref}) + v_{ref}\sin(\theta_{ref})\sin(\theta_{ref}) \\ &= v_{ref}(\sin^2(\theta_{ref}) + \cos^2(\theta_{ref})) = v_{ref}\end{aligned}$$

Finally, using the previous equality, the result for $\dot{y}_e$ is

$$\dot{y}_e = -x_e w_{cur} + v_{ref}\sin(\theta_e) \tag{3.21}$$

- **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: $\theta_e = \theta_{ref} - \theta_{cur}$.

The result is:

$$\dot{\theta}_e = \dot{\theta}_{ref} - \dot{\theta}_{cur} = w_{ref} - w_{cur} \tag{3.22}$$

3.3.3 Error model equations

The obtained results for the error model, equations (3.20), to (3.22), are shown here in matrix form

$$\dot{p}_e = \begin{bmatrix}\dot{x}_e \\ \dot{y}_e \\ \dot{\theta}_e\end{bmatrix} = \begin{bmatrix} w_{cur}y_e - v_{cur} + v_{ref}\cos(\theta_e) \\ -w_{cur}x_e + v_{ref}\sin(\theta_e) \\ w_{ref} - w_{cur}\end{bmatrix} \tag{3.23}$$

If it is understood that v_{cur} and w_{cur} depend on p_e and q_{ref}, where p_e y q_{ref} are the inputs of the reference model. Thus, the model can be rewritten as

$$\dot{p}_e = \begin{bmatrix}\dot{x}_e \\ \dot{y}_e \\ \dot{\theta}_e\end{bmatrix} = \begin{bmatrix} w(p_e, q_{ref})y_e - v(p_e, q_{ref}) + v_{ref}\cos(\theta_e) \\ -w(p_e, q_{ref})x_e + v_{ref}\sin(\theta_e) \\ w_{ref} - w(p_e, q_{ref})\end{bmatrix} \tag{3.24}$$

where v_{cur} and w_{cur} are the inputs of the real car.

Separating the inputs of the virtual car and the real one, the resulting matrix of the dynamics of the error is

$$\begin{aligned}\dot{p}_e = \begin{bmatrix}\dot{x}_e \\ \dot{y}_e \\ \dot{\theta}_e\end{bmatrix} &= \begin{bmatrix}\cos(\theta_e) & 0 \\ \sin(\theta_e) & 0 \\ 0 & 1\end{bmatrix}\begin{bmatrix}v_{ref} \\ w_{ref}\end{bmatrix} + \begin{bmatrix}-1 & y_e \\ 0 & -x_e \\ 0 & -1\end{bmatrix}\begin{bmatrix}v_{cur} \\ w_{cur}\end{bmatrix} \\ &= \begin{bmatrix}\cos(\theta_e) & 0 \\ \sin(\theta_e) & 0 \\ 0 & 1\end{bmatrix}\begin{bmatrix}v_{ref} \\ w_{ref}\end{bmatrix} + \begin{bmatrix}-1 & y_e \\ 0 & -x_e \\ 0 & -1\end{bmatrix}\begin{bmatrix}v(p_e, q_{ref}) \\ w(p_e, q_{ref})\end{bmatrix}\end{aligned} \tag{3.25}$$

The inputs that will be used to control the system are the ones of the real car $v(p_e, q_{ref})$ and $w(p_e, q_{ref})$.

Chapter 4

Control Design

In this Chapter, an LPV controller for the vehicle is designed. The starting point is the error model obtained in the previous Chapter. A basic scheme of the whole system is shown. Using the error model and the basic scheme of the system, equations will be shaped until they are ready to apply the LPV control design approach.. Two LPV models are developed: the first one is shaped from the starting error model, while two new states are introduced in the second model. The polytopic transformation via nonlinear embedding and bounding box approach has been applied to both models in order to obtain a state feedback control suitable for any operating point inside the bounding box.

4.1 Starting point

The error model equations obtained in Chapter 3 are the starting point for the control design:

$$\dot{p}_e = \begin{bmatrix} \dot{x}_e \\ \dot{y}_e \\ \dot{\theta}_e \end{bmatrix} = \begin{bmatrix} \cos(\theta_e) & 0 \\ \sin(\theta_e) & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v_{ref} \\ w_{ref} \end{bmatrix} + \begin{bmatrix} -1 & y_e \\ 0 & -x_e \\ 0 & -1 \end{bmatrix} \begin{bmatrix} v_{cur} \\ w_{cur} \end{bmatrix} \tag{4.1}$$

The aim of the controller is to perform a path tracking given a reference signal which changes over time. It is assumed that the vehicle can only move forward and that the path is always ahead. The control proposed in this Thesis is an LPV state feedback. Figure 4.1 shows the basic scheme of the system including the controller. In this Figure, there are two differentiated parts: feed forward (virtual vehicle) and feedback (system of the error). The feedback state is the error posture (which is the error between the reference and the real car). The input control action is the addition of two different signals: one coming from the reference, u_{ref}, and the other from the control, u_b. Linear velocities of the reference (or virtual) and real vehicle are one dimensional values. Therefore, they must be aligned to add them correctly. From Figure 3.4 of Chapter 3, now shown in this Chapter as Figure 4.2, this fact is easier to understand. Equations (4.2) and (4.3) express the control signals resulting from this reasoning.

$$v_{cur} = v_{ref}\cos(\theta_e) + v_b = v_{ref}\cos(\theta_{ref} - \theta_{cur}) + v_b \tag{4.2}$$

$$w_{cur} = w_{ref} + w_b \tag{4.3}$$

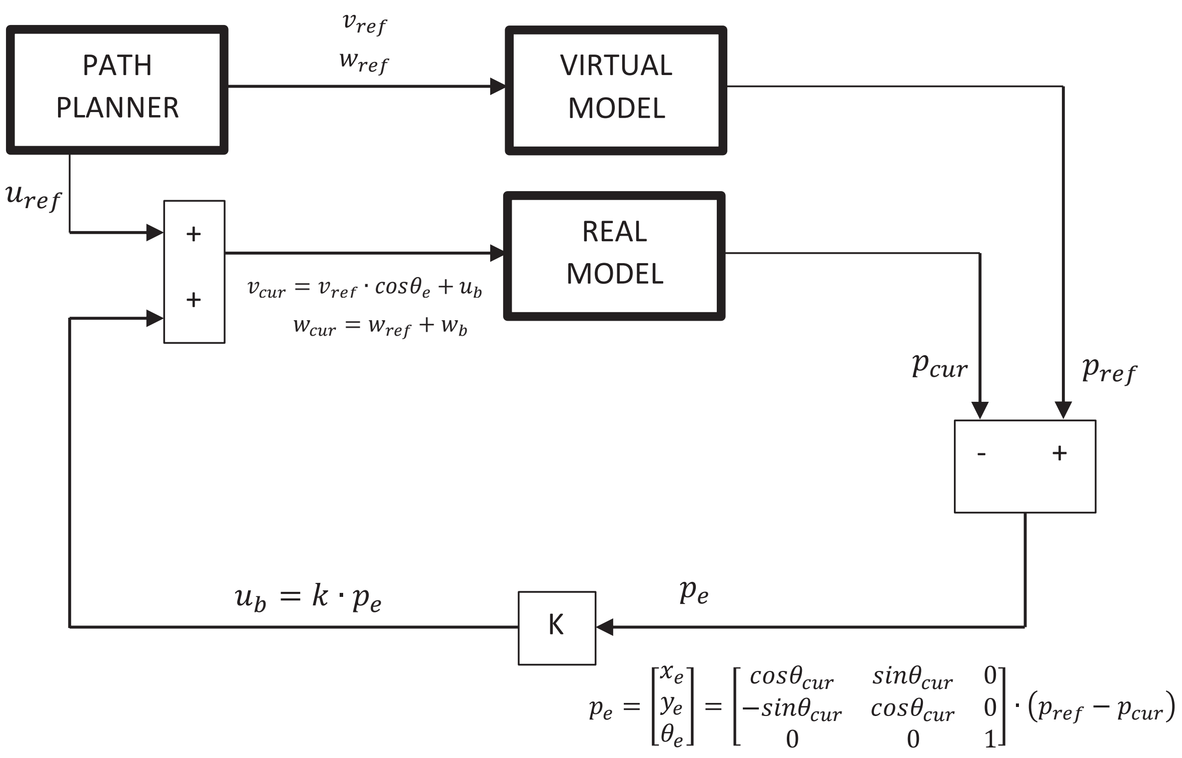
PATH PLANNER
v_{ref}
w_{ref}
VIRTUAL MODEL
u_{ref}
+
+
REAL MODEL
$v_{cur} = v_{ref} \cdot cos\theta_e + u_b$
$w_{cur} = w_{ref} + w_b$
p_{cur}
p_{ref}
-
+
$u_b = k \cdot p_e$
K
p_e
$p_e = \begin{bmatrix} x_e \\ y_e \\ \theta_e \end{bmatrix} = \begin{bmatrix} cos\theta_{cur} & sin\theta_{cur} & 0 \\ -sin\theta_{cur} & cos\theta_{cur} & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot (p_{ref} - p_{cur})$

Figure 4.1: Basic scheme of the system with the control

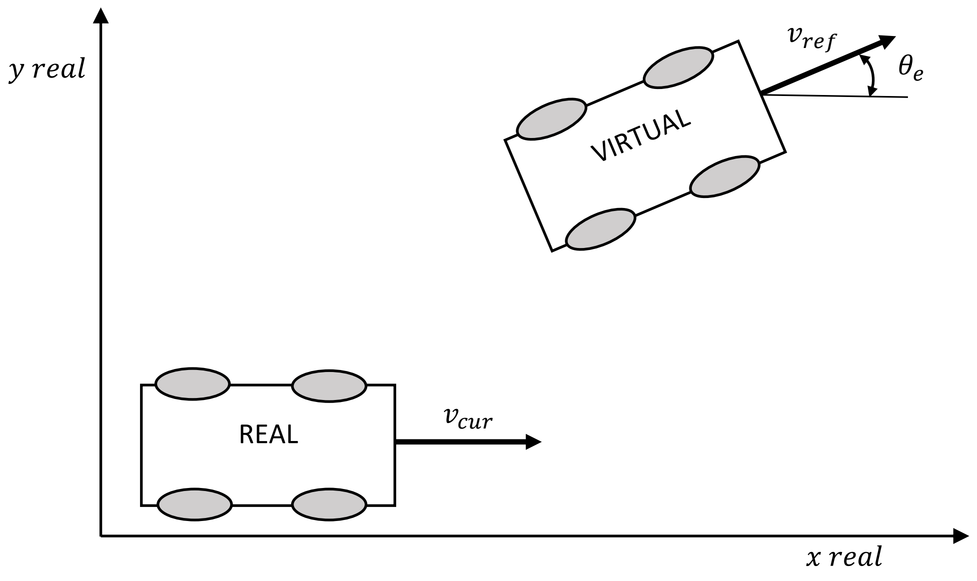

Figure 4.2: Real and virtual car

4.2 First model

4.2.1 Error model with control actions

The whole system shown in the Figure 4.1 can be expressed in equations by just introducing (4.2) and (4.3) into the dynamics of the system, (4.1) leading to:

$$\begin{aligned}\dot{p}_e = \begin{bmatrix}\dot{x}_e\\ \dot{y}_e\\ \dot{\theta}_e\end{bmatrix} &= \begin{bmatrix}\cos(\theta_e) & 0\\ \sin(\theta_e) & 0\\ 0 & 1\end{bmatrix}\begin{bmatrix}v_{ref}\\ w_{ref}\end{bmatrix} + \begin{bmatrix}-1 & y_e\\ 0 & -x_e\\ 0 & -1\end{bmatrix}\begin{bmatrix}v_{cur}\\ w_{cur}\end{bmatrix}\\ &= \begin{bmatrix}\cos(\theta_e) & 0\\ \sin(\theta_e) & 0\\ 0 & 1\end{bmatrix}\begin{bmatrix}v_{ref}\\ w_{ref}\end{bmatrix} + \begin{bmatrix}-1 & y_e\\ 0 & -x_e\\ 0 & -1\end{bmatrix}\begin{bmatrix}v_{ref}\cos(\theta_e) + v_b\\ wref + w_b\end{bmatrix}\end{aligned} \tag{4.4}$$

The equations of the error model that include the control actions are obtained by operating the matrices:

$$\dot{x}_e = v_{ref}\cos(\theta_e) - v_{ref}\cos(\theta_e) - v_b + y_e w_{ref} + y_e w_b = -v_b + y_e w_{ref} + y_e w_b \tag{4.5}$$

$$\dot{y}_e = v_{ref}\sin(\theta_e) - x_e w_{ref} - x_e w_b \tag{4.6}$$

$$\dot{\theta}_e = w_{ref} - w_{ref} - w_b = -w_b \tag{4.7}$$

where the control variables of this system are v_b and w_b and the states are x_e, y_e and θ_e. Hence, the input control signals of the system v_{cur} and w_{cur} will be controlled just by controlling v_b and w_b.

4.2.2 System in LPV form

The error model can be shaped into an LPV form as follows:

$$\begin{bmatrix} \dot{x}_e \\ \dot{y}_e \\ \dot{\theta}_e \end{bmatrix} = \begin{bmatrix} 0 & w_{ref} & 0 \\ -w_{ref} & 0 & \frac{v_{ref}\sin(\theta_e)}{\theta_e} \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} x_e \\ y_e \\ \theta_e \end{bmatrix} + \begin{bmatrix} -1 & y_e \\ 0 & -x_e \\ 0 & -1 \end{bmatrix} \begin{bmatrix} v_b \\ w_b \end{bmatrix} \tag{4.8}$$

According to [3] this error model has a problem: the transformation between the robot posture and the error model is not bijective. The same robot posture is obtained if the vehicle is rotated 360^{o} (or an integer multiple of it). This would lead to errors in the orientation and discontinuities of the angular velocity when the orientation error crosses $\pm\pi$. In [3], it is proposed a different model to solve this problem. Although the nature of the tracking problem treated in this Thesis is not affected by this issue, both models are compared in order to analyse their performance and characteristics.

4.3 Second model

4.3.1 New states

The proposed way of solving the problem of the model presented in the previous section is to increase the order of the model according to [3]. In order to do this, the variable $\theta(t)$ is substituted by two periodic variables: $s(t) = \sin(\theta(t))$ and $c(t) = \cos(\theta(t))$.

Their derivatives are

$$\dot{s}(t) = \dot{\theta}(t)\cos(\theta(t)) = c(t)w(t) \tag{4.9}$$

$$\dot{c}(t) = -\dot{\theta}(t)\sin(\theta(t)) = -s(t)w(t) \tag{4.10}$$

4.3.2 Error model

The kinematic equations of the vehicle for the second model results from introducing the new states, (4.9) (4.10), in the bicycle-like model, (3.1) to (3.3), are

$$\dot{p} = \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{s} \\ \dot{c} \end{bmatrix} = \begin{bmatrix} c & 0 \\ s & 0 \\ 0 & c \\ 0 & -s \end{bmatrix} \begin{bmatrix} v \\ w \end{bmatrix} \tag{4.11}$$

The new error states are $s_e(t) = \sin(\theta_{ref}(t) - \theta_{cur}(t))$ and $c_e(t) = \cos(\theta_{ref}(t) - \theta_{cur}(t))$. Thus, the error posture with these new states can be written as follows

$$p_e = \begin{bmatrix} x_e \\ y_e \\ s_e \\ c_e \end{bmatrix} = \begin{bmatrix} c_{cur} & s_{cur} & 0 & 0 \\ -s_{cur} & c_{cur} & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} (p_{ref} - p_{cur}) = \begin{bmatrix} c_{cur} & s_{cur} & 0 & 0 \\ -s_{cur} & c_{cur} & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_{ref} - x_{cur} \\ y_{ref} - y_{cur} \\ \sin(\theta_{ref} - \theta_{cur}) \\ \cos(\theta_{ref} - \theta_{cur}) \end{bmatrix}$$
$$= \begin{bmatrix} c(x_{ref} - x_{cur}) + s(y_{ref} - y_{cur}) \\ -s(x_{ref} - x_{cur}) + c(y_{ref} - y_{cur}) \\ \sin(\theta_{ref} - \theta_{cur}) \\ \cos(\theta_{ref} - \theta_{cur}) \end{bmatrix} = \begin{bmatrix} \cos(\theta_{cur})(x_{ref} - x_{cur}) + \sin(\theta_{cur})(y_{ref} - y_{cur}) \\ -\sin(\theta_{cur})(x_{ref} - x_{cur}) + \cos(\theta_{cur})(y_{ref} - y_{cur}) \\ \sin(\theta_{ref})\cos(\theta_{cur}) - \cos(\theta_{ref})\sin(\theta_{cur}) \\ \cos(\theta_{ref})\cos(\theta_{cur}) + \sin(\theta_{ref})\sin(\theta_{cur}) \end{bmatrix} \tag{4.12}$$

If the error posture model is differentiated and manipulated, following the same procedure used in Chapter 3 when calculating time derivatives of the error states, the error model ends up being

$$\dot{x}_e = v_{ref}c_e - v_{cur} + y_e w_{cur} \tag{4.13}$$

$$\dot{y}_e = v_{ref}s_e - x_e w_{cur} \tag{4.14}$$

$$\dot{s}_e = w_{ref}c_e - c_e w_{cur} \tag{4.15}$$

$$\dot{c}_e = -w_{ref}s_e + s_e w_{cur} \tag{4.16}$$

The matrix equivalent form is

$$\dot{p}_e = \begin{bmatrix} \dot{x}_e \\ \dot{y}_e \\ \dot{s}_e \\ \dot{c}_e \end{bmatrix} = \begin{bmatrix} c_e & 0 \\ s_e & 0 \\ 0 & c_e \\ 0 & -s_e \end{bmatrix} \begin{bmatrix} v_{ref} \\ w_{ref} \end{bmatrix} + \begin{bmatrix} -1 & y_e \\ 0 & -x_e \\ 0 & -c_e \\ 0 & s_e \end{bmatrix} \begin{bmatrix} v_{cur} \\ w_{cur} \end{bmatrix} \tag{4.17}$$

4.3.3 Error model with control actions

The control actions for the second model are the same as the ones used for the first one. Therefore, they are the following

$$v_{cur} = v_{ref}\cos(\theta_e) + v_b = v_{ref}\cos(\theta_{ref} - \theta_{cur}) + v_b = v_{ref}c_e + v_b \tag{4.18}$$

$$w_{cur} = w_{ref} + w_b \tag{4.19}$$

Introducing them in $\dot{p}_e$, (4.17), the error model with the control actions is obtained

$$\dot{x}_e = v_{ref}c_e - v_{cur} + y_e w_{cur} = -v_b + y_e w_{ref} + y_e w_b \tag{4.20}$$

$$\dot{y}_e = v_{ref}s_e - x_e w_{cur} = v_{ref}s_e - x_e w_{ref} - x_e w_b \tag{4.21}$$

$$\dot{s}_e = w_{ref}c_e - c_e w_{cur} = -c_e w_b \tag{4.22}$$

$$\dot{c}_e = -w_{ref}s_e + s_e w_{cur} = s_e w_b \tag{4.23}$$

which in matrix form is

$$\begin{aligned} \dot{p}_e = \begin{bmatrix} \dot{x}_e \\ \dot{y}_e \\ \dot{s}_e \\ \dot{c}_e \end{bmatrix} &= \begin{bmatrix} c_e & 0 \\ s_e & 0 \\ 0 & c_e \\ 0 & -s_e \end{bmatrix} \begin{bmatrix} v_{ref} \\ w_{ref} \end{bmatrix} + \begin{bmatrix} -1 & y_e \\ 0 & -x_e \\ 0 & -c_e \\ 0 & s_e \end{bmatrix} \begin{bmatrix} v_{cur} \\ w_{cur} \end{bmatrix} \\ &= \begin{bmatrix} 0 & w_{ref} & 0 & 0 \\ -w_{ref} & 0 & v_{ref} & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} x_e \\ y_e \\ s_e \\ c_e \end{bmatrix} + \begin{bmatrix} -1 & y_e \\ 0 & -x_e \\ 0 & -c_e \\ 0 & s_e \end{bmatrix} \begin{bmatrix} v_b \\ w_b \end{bmatrix} \end{aligned} \tag{4.24}$$

ETSEIB

4.3.4 System in LPV form

Looking at the (4.24), it is important to realize that it is impossible to bring these states to zero in the steady state. When s_e is zero c_e will be one and vice versa. Looking at the control signal of the linear velocity $v_{cur} = v_{ref} c_e + v_b$, when the error is zero, it means that $v_{cur} = v_{ref}$, then c_e must be one. Hence, the appropriate state variable to compute the state feedback control is s_e because it will go to zero on the steady state. Then, the second model system is split in two parts

$$\dot{p}_e = \begin{bmatrix} \dot{x}_e \\ \dot{y}_e \\ \dot{s}_e \end{bmatrix} = \begin{bmatrix} 0 & w_{ref} & 0 \\ -w_{ref} & 0 & v_{ref} \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} x_e \\ y_e \\ s_e \end{bmatrix} + \begin{bmatrix} -1 & y_e \\ 0 & -x_e \\ 0 & -c_e \end{bmatrix} \begin{bmatrix} v_b \\ w_b \end{bmatrix} \tag{4.25}$$

$$\dot{c}_e = s_e w_b \tag{4.26}$$

The first part is the one that will be used to design the control. As it can be seen, the parameters of A will never be 0 unless w_{ref} or v_{ref} are zero. Thus, the system is controllable anywhere else.

4.4 Apkarian Filter

An LPV system in general form can be expressed as follows

$$\dot{x} = A(\vartheta)x + B(\vartheta)u \tag{4.27}$$

$$y = C(\vartheta)x + D(\vartheta)u \tag{4.28}$$

The followed procedure has been applied to embed all the nonlinearities and variable parameters in matrix A. The aim is to transform the LPV system until it presents the following shape:

$$\dot{x} = A(\vartheta)x + Bu \tag{4.29}$$

$$y = Cx \tag{4.30}$$

The first model, (4.8), and the second model, (4.25), have been transformed with the goal of avoiding the dependency on varying parameters of matrix B. Here, a little trick is needed: the order of the system is increased by adding a filter [14]. This filter is expressed in state space form. Its state variables and input signals are selected consciously. The output of the new state space (the filter) is chosen in order to be equal to the input of the system we had, u_b. The state space of the filter is:

$$\dot{x}_u = A_u x_u + B_u u_{new} \tag{4.31}$$

$$y_u = u_b = C_u x_u \tag{4.32}$$

where u_{new} is the new control signal.

Thus, the augmented state space model including the filter states can be expressed as follows:

$$\dot{p}_e = A(\varepsilon)p_e + B(\varepsilon)u_b = A(\varepsilon)p_e + B(\varepsilon)C_u x_u \tag{4.33}$$

$$\dot{x}_u = A_u x_u + B_u u_{new} \tag{4.34}$$

$$y = u_b = Cx \tag{4.35}$$

where ε are the varying parameters of the error model.

The augmented system in matrix form is the following:

$$\begin{bmatrix} \dot{p}_e \\ \dot{x}_u \end{bmatrix} = \begin{bmatrix} A(\varepsilon) & B(\varepsilon)C_u \\ 0_{2x3} & A_u \end{bmatrix} \begin{bmatrix} p_e \\ x_u \end{bmatrix} + \begin{bmatrix} 0 \\ B_u \end{bmatrix} u_{new} \tag{4.36}$$

$$y = u_b = Cx \tag{4.37}$$

where in the case of the first model the system matrices are given by:

$$A(\varepsilon) = \begin{bmatrix} 0 & w_{ref} & 0 \\ -w_{ref} & 0 & \frac{v_{ref} sin(\theta_e)}{\theta_e} \\ 0 & 0 & 0 \end{bmatrix} \tag{4.38}$$

$$B(\varepsilon) = \begin{bmatrix} -1 & y_e \\ 0 & -x_e \\ 0 & -1 \end{bmatrix} \tag{4.39}$$

while for the second model:

$$A(\varepsilon) = \begin{bmatrix} 0 & w_{ref} & 0 \\ -w_{ref} & 0 & v_{ref} \\ 0 & 0 & 0 \end{bmatrix} \tag{4.40}$$

$$B(\varepsilon) = \begin{bmatrix} -1 & y_e \\ 0 & -x_e \\ 0 & -c_e \end{bmatrix} \tag{4.41}$$

and C_u is the identity matrix to ease the computations. A_u is chosen to present negligible poles that will not affect the results, $A_u = \begin{bmatrix} -100 & 0 \\ 0 & -100 \end{bmatrix}$, $u_{new} = \begin{bmatrix} u_{new_1} \\ u_{new_2} \end{bmatrix}$ because the dimensions must agree and $B_u = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 1 & 0 \\ 0 & 1 \end{bmatrix}$ because the control variables of the augmented control are v_b and w_b. Then, the augmented model for the two considered models can be written as follows:

– Augmented first model

$$\begin{bmatrix} \dot{p}_e \\ \dot{x}_u \end{bmatrix} = \begin{bmatrix} A(\varepsilon) & B(\varepsilon)C_u \\ 0_{2x3} & A_u \end{bmatrix} \begin{bmatrix} p_e \\ x_u \end{bmatrix} + \begin{bmatrix} 0 \\ B_u \end{bmatrix} u_{new}$$
$$= \begin{bmatrix} 0 & w_{ref} & 0 & -1 & y_e \\ -w_{ref} & 0 & \frac{v_{ref} sin(\theta_e)}{\theta_e} & 0 & -x_e \\ 0 & 0 & 0 & 0 & -1 \\ 0 & 0 & 0 & -100 & 0 \\ 0 & 0 & 0 & 0 & -100 \end{bmatrix} \begin{bmatrix} x_e \\ y_e \\ \theta_e \\ x_{u_1} \\ x_{u_2} \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} u_{new_1} \\ u_{new_2} \end{bmatrix} \tag{4.42}$$

– Augmented second model

$$\begin{bmatrix} \dot{p}_e \\ \dot{x}_u \end{bmatrix} = \begin{bmatrix} A(\varepsilon) & B(\varepsilon)C_u \\ 0_{2x3} & A_u \end{bmatrix} \begin{bmatrix} p_e \\ x_u \end{bmatrix} + \begin{bmatrix} 0 \\ B_u \end{bmatrix} u_{new}$$
$$= \begin{bmatrix} 0 & w_{ref} & 0 & -1 & y_e \\ -w_{ref} & 0 & v_{ref} & 0 & -x_e \\ 0 & 0 & 0 & 0 & -c_e \\ 0 & 0 & 0 & -100 & 0 \\ 0 & 0 & 0 & 0 & -100 \end{bmatrix} \begin{bmatrix} x_e \\ y_e \\ s_e \\ x_{u_1} \\ x_{u_2} \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} u_{new_1} \\ u_{new_2} \end{bmatrix} \tag{4.43}$$

In both models, the output of the augmented system are the states x_{u_1}, x_{u_2}, which are related with the inputs as $x_{u_1} = v_b$ and $x_{u_2} = w_b$. Thus, matrix C of the augmented system is

$$\begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \tag{4.44}$$

With these transformation, the goal of having all the varying parameters inside matrix A of the system has been achieved. Therefore, the augmented system is in the desired shape

$$\dot{x} = A(\vartheta)x + Bu \tag{4.45}$$

$$y = Cx \tag{4.46}$$

4.5 LPV parameters

The bounding box of each model is obtained by taking all possible combinations of the varying parameters in the augmented system into account. The varying parameters, ϑ, are scheduled considering their scheduling variables.

- First model

The augmented system of the first model, (4.42), has the following varying parameters:

$$\vartheta_1 = w_{ref}$$

$$\vartheta_2 = -w_{ref} = -\vartheta_1$$

$$\vartheta_3 = \frac{v_{ref} sin(\theta_e)}{\theta_e}$$

$$\vartheta_4 = y_e$$

$$\vartheta_5 = x_e$$

hence, x_e, y_e, θ_e, v_{ref} and w_{ref} are the scheduling variables.

- Second model

The augmented system of the second model, (4.43), has the following varying parameters:

$$\vartheta_1 = w_{ref}$$

$$\vartheta_2 = -w_{ref} = -\vartheta_1$$

$$\vartheta_3 = v_{ref}$$

$$\vartheta_4 = y_e$$

$$\vartheta_5 = x_e$$

$$\vartheta_6 = -c_e$$

hence, x_e, y_e, c_e, v_{ref} and w_{ref} are the scheduling variables.

4.6 LPV Controller

The designed controller is a state feedback LPV one ($u = K(\vartheta)x$) that considers the decay ratio as design parameter. This implies placing all the poles of the controlled system at the hyperplane defined on the left side of a desired real value on the complex plane. The LMIs used for this kind of design, taking into account the shape of the augmented system, are:

$$(A(\vartheta) + BK)P + P(A(\vartheta) + BK)^t + 2\alpha P < 0 \ , \quad P > 0 \qquad \forall\ \vartheta P > 0 \qquad (4.47)$$

where α is the desired value that defines the right boundary of the hyperplane. K is a proportional controller that is computed for each vertex of the bounding box. The bounding box is defined by the boundaries of the parameters of the augmented system. And P is a Lyapunov function that, if it exists, ensures stability with the pre-specified decay rate.

4.6.1 Boundaries

Each one of the models has five scheduling variables to define the varying parameters. These variables give a total of 2^5 possible combinations. The combinations define the varying parameters and therefore, the vertex of the bounding box that contains all possible operating points.

The boundaries of these variables have been chosen taking into account the specifications of the real car considered as case of study in this Thesis (Tazzari Zero, model Classic, [19]) and are the following:

- $v_{ref} \in [0.1, 50]$, the minimum value is chosen different to zero because the car would not move otherwise.

- $w_{ref} \in [-11.25, 11.25]$, these values are obtained from the limitation imposed by the steering angle (maximum 24°) and computing the equation of the bicycle model $\dot{\theta}_{ref} = \frac{v_{ref}}{L}\tan(\varphi_{ref}) = w_{ref}$.

- $x_e \in [-0.5, 0.5]$ and $y_e \in [-0.5, 0.5]$ are chosen with the intention of limiting the error between those boundaries.

- **Only for the first model:** $\theta_e \in [-1.5567, 1.5567]$, these values are chosen considering the assumption that the car will always move forward, hence the angle will always be inside the range $-90^o < \theta < 90^o$.

- **Only for the second model:** $c_e \in [0.1, 1]$, these values are chosen following the same assumption. Taking into account this assumption the values of $cos(\theta_e)$ will vary between 90^o and -90^o, which is equivalent to saying that the values of c_e will be between $[0, 1]$. The 0 point has been avoided and changed by 0.1.

4.6.2 Control

Angular velocities of the system can be either positive or negative. This fact produces infeasibility at the time of computing the LMIs for a bounding box which includes all positive and negative values. In order to make the set of LMIs feasible, the system has been split in two groups, each one with different bounds of angular velocity. One for positive values of $w_{ref} = \begin{bmatrix} 0.1 & 11.25 \end{bmatrix}$ and the other for $w_{ref} = \begin{bmatrix} -11.25 & -0.1 \end{bmatrix}$. This procedure leads to the necessity of using a switch to swap between the subsystems.

Switched LPV Control

When solving the path tracking problem treated in this Thesis using the LPV approach, a singular point inside the bounding box, $w_{ref} = 0$, appears preventing from finding a solution. Alternatively, the polytopic LPV model is decomposed in two different regions to avoid this singularity formulating a switched LPV model. This model includes both continuous-valued and discrete-valued varying parameters.

The LMIs which need to be solved include the switched LPV model as follows

$$(A_\sigma(\vartheta) + BK_\sigma)P + P(A_\sigma(\vartheta) + BK_\sigma)^t + 2\alpha P < 0 \ , \quad P > 0 \qquad \forall\, \vartheta P > 0 \tag{4.48}$$

where σ is the variable that allows the switching between subsystems.

Each LMI of the set of LMIs expressed in (4.48) should be solved for P and for K simultaneously. This is not possible because the problem is nonlinear. A variable change is made in order to overcome this issue

$$K = WP^{-1} \tag{4.49}$$

leading to the following LMI

$$A_\sigma(\vartheta)P + BW_\sigma + PA_\sigma(\vartheta)^t + W_\sigma^t B^t + 2\alpha P < 0 \ , \quad P > 0 \qquad \forall\, \vartheta P > 0 \tag{4.50}$$

The total set of LMIs to be solved are 2^5 for each subsystem, obtaining 2^5 different W's. Then, 2^5 K's for each subsystem are computed from the W's. These controllers must be weighted according to all operating points inside the bounding box.

Weighted controls

An LPV polytopic system is formally described by the equations

$$\dot{x}(t) = \sum_{i=1}^{N} \mu_i(\vartheta(t))(A_i x(t) + B_i u(t)) \tag{4.51}$$

$$y(t) = \sum_{i=1}^{N} \mu_i(\vartheta(t))(C_i x(t) + D_i u(t)) \tag{4.52}$$

where A_i, B_i, C_i and D_i define the vertex systems. μ_i is the scheduling function, or wheights. The varying parameters and weights fulfill the following condition:

$$\sum_{i=1}^{N} \mu_i(\vartheta(t)) = 1, \quad \mu_i(\vartheta(t)) \geq 0, \quad \forall i = 1, ..., N, \quad \forall \vartheta \in \Theta \tag{4.53}$$

where Θ is the polytope, or bounding box.

The states of the system are the ones that define the operating point and, therefore the controller to use. In order to have a varying control, a set of functions are needed. They allow us to map any operating point inside the bounding box with a weight of the control of each vertex. These functions, which depend on the parameters, express the degree of membership that each vertex control will have in the final control action.

Each vertex controller K_i will have a weight depending on the operating point. This weight is obtained with the aforementioned functions $\mu_i(\vartheta)$. Some examples of these functions for the second model are shown here

$$\begin{aligned}
&\mu_1(\vartheta_1,\vartheta_3,\vartheta_4,\vartheta_5,\vartheta_6) = M1_{min}(w_{ref})M3_{min}(v_{ref})M4_{min}(y_e)M5_{min}(x_e)M6_{min}(c_e)\\
&\mu_2(\vartheta_1,\vartheta_3,\vartheta_4,\vartheta_5,\vartheta_6) = M1_{min}(w_{ref})M3_{min}(v_{ref})M4_{min}(y_e)M5_{min}(x_e)M6_{max}(c_e)\\
&\dots\\
&\mu_{31}(\vartheta_1,\vartheta_3,\vartheta_4,\vartheta_5,\vartheta_6) = M1_{max}(w_{ref})M3_{max}(v_{ref})M4_{max}(y_e)M5_{max}(x_e)M6_{min}(c_e)\\
&\mu_{32}(\vartheta_1,\vartheta_3,\vartheta_4,\vartheta_5,\vartheta_6) = M1_{max}(w_{ref})M3_{max}(v_{ref})M4_{max}(y_e)M5_{max}(x_e)M6_{max}(c_e)
\end{aligned} \tag{4.54}$$

Linear interpolation rules are used to obtain the weights μ_i. The following expressions have been used

$$Mi_{min} = \frac{\vartheta_i(t) - \vartheta_{i_{min}}}{\vartheta_{i_{max}} - \vartheta_{i_{min}}} \tag{4.55}$$

which is zero when the parameter $\vartheta_i = \vartheta_{i_{max}}$, and $M_{i_{max}} = 1 - M_{i_{min}}$.

In total there are 2^5 rules of interpolation and mapping functions for each subsystem. Each one of the rules of interpolation must have values between zero and one because the resulting controller must be inside the bounding box of controllers. To do so, the value of a state is assigned to be its maximum when its current value is higher than it. In the lower case, the minimum is assigned.

The unified final gain of the controller at any instance of time is obtained by adding all 32 weighted controllers of the active subsystem

$$K(\vartheta(t)) = \sum_{i=1}^{32} \mu_i(\vartheta(t))K_i \tag{4.56}$$

Control law

State feedback control law is applied once the gain of the unified control is obtained

$$u_{new} = K(\vartheta)x = (\sum_{i=1}^{32} \mu_i(\vartheta)K_i)x \tag{4.57}$$

These control inputs in the augmented state space are u_{new_1} and u_{new_2}. These inputs are transformed into v_b and w_b unfiltering with the same Apkarian filter ((4.31) and (4.32)) used to move the varying parameters in B to the A matrix. As it has been stated in Section 4.4, the dynamics of the filter are fast enough to not influence the rest of the system. Therefore, inputs before being filtered and after being unfiltered are the same. Hence, the control actions of the real vehicle are the following:

$$v_{cur} = v_{ref} c_e + v_b \tag{4.58}$$

$$w_{cur} = w_{ref} + w_b \tag{4.59}$$

Chapter 5

Results

This Chapter shows the different results obtained in simulation from the application of the proposed LPV controller for the autonomous vehicle. A description of the Simulink scheme used for simulation is given. Results are presented in an order to help the reader to understand the design procedure for the first and the second model, which are compared in each step.

5.1 Introduction

Results of both models are shown as the Chapter progresses and the main differences between them are pointed out. The order the results are presented follows the design procedure chronologically. First of all, the controllers described in Chapter 4 have been separately checked for each group of LMIs, $w_{ref} > 0$ and $w_{ref} < 0$, with random trajectories as the input reference. Once both subsystems work, the switched LPV controller is applied to the whole system. Finally, two different reference trajectories: a line and a circle are tested. Reference inputs, trajectories of the virtual and real vehicle, states, control inputs and errors between the virtual and real vehicle's output are the results of interest shown.

5.2 Description of the Simulink scheme of the system

Figure 5.1 shows the Simulink schematic used to simulate the system.

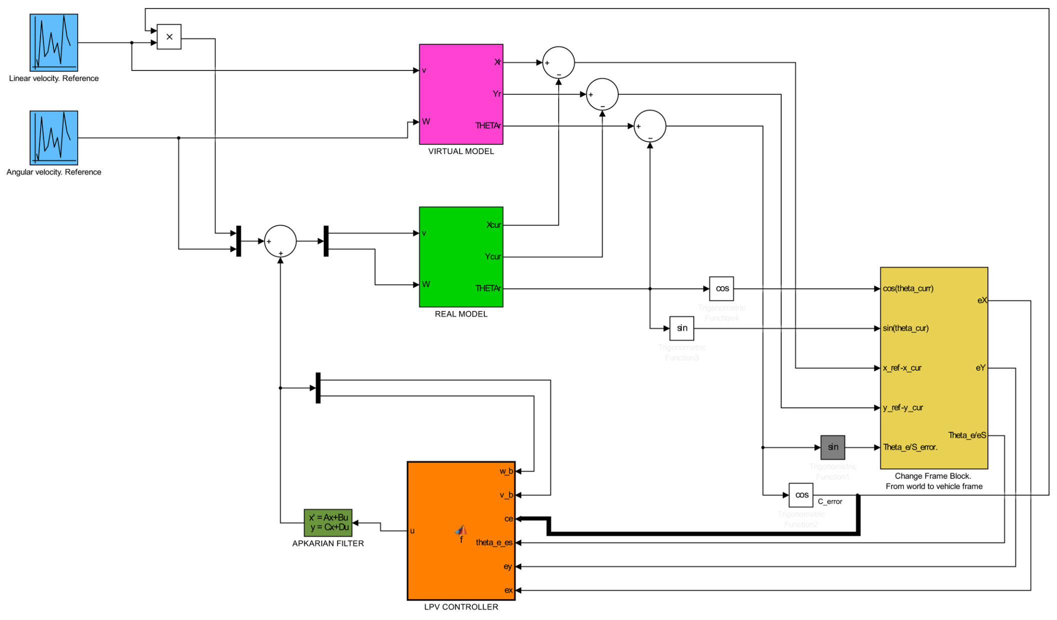

Figure 5.1: Simulink scheme.

A brief description of the main blocks is given before starting showing the simulation results.

- Blue blocks: They define the reference trajectory using velocities. The one above is the linear velocity reference and the one below is the angular velocity reference. The Simulink block used is the Uniform random number, in both cases. The parameters of this block are: minimum value, maximum value, mean value and sample time. The minimum and maximum value have been chosen according to the boundaries described in Section 4.6.1. Their values are between 0.1 and 50 in the case of linear velocities and between 11.25 and -11.25 for angular velocities. The obtained signals are a chain of pulses. An example of both reference signals is shown in Figure 5.2 and the trajectory resulting from applying them to the virtual vehicle is presented in Figure 5.3. This approach to the reference trajectory is not at all conservative, car velocities can change between any value inside the boundaries in an instant of time. A real car will never behave like this, changes in linear and angular velocities will be smoother. Therefore, if the system works in these conditions, it will work with less constrained ones as well.

- Purple and light green block: They are the models (bicycle model) of both vehicles, the virtual and the real car. Inside the blocks, the only difference between them is that the real car has got both inputs, the linear and angular velocity, saturated according to the limits of the real vehicle.

- Yellow block: This block is the one that changes from the world to the vehicle reference frame. Its outputs are the states of the system. For the first model: x_e, y_e, θ_e and for the second model: x_e, y_e, s_e
- Orange block: This is the switched LPV controller. It is computed using the amplified system as previously shown in Section 4.4. Therefore, it is different for the first and second model.
- Dark green block: This is an state space block of the Apkarian Filter applied (for a more detailed description see Section 4.4).
- Grey block: This is a block only used in the second model and is the one that implements the variable change explained in Section 4.3.1.
- Thicker line: This connection between the $cos(\theta_e)$ and the switched LPV controller and is only needed in the second model, because c_e is the one varying its parameters.

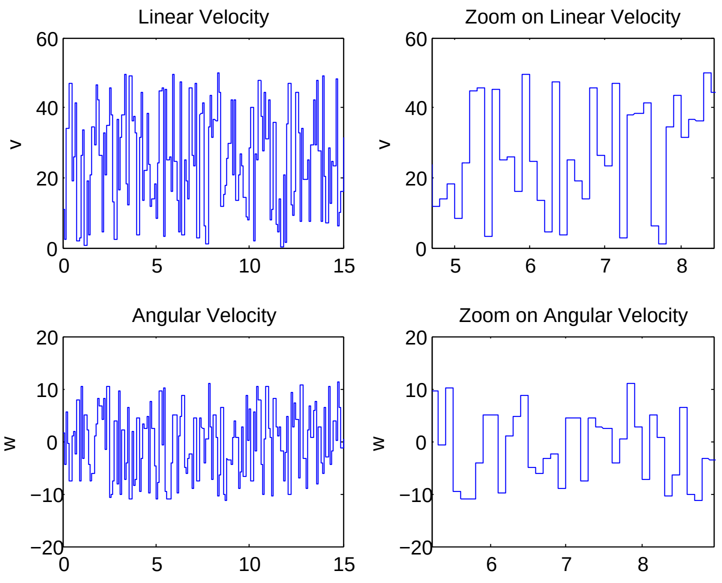

Figure 5.2: Inputs from Path Planner to generate the trajectory

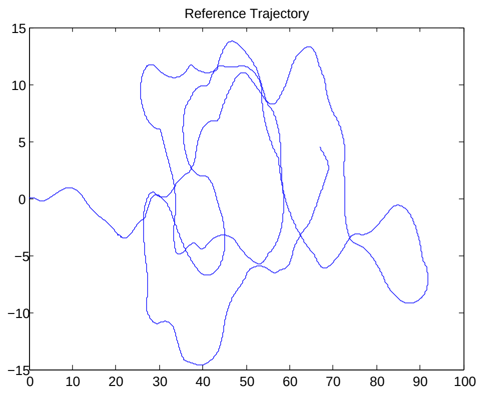

Figure 5.3: Example of trajectory

5.3 Results

5.3.1 Checking subsystems

The first step is to check that both families of controllers, $w_{ref} > 0$ and $w_{ref} < 0$ work. The input reference signal of angular velocity is either positive or negative but not both. The controller is the one computed taking the angular velocity sign into account. Figure 5.4 shows the performed tracking of the real vehicle in both cases. Here, it is only shown for the second model and it can be seen that it works in both families.

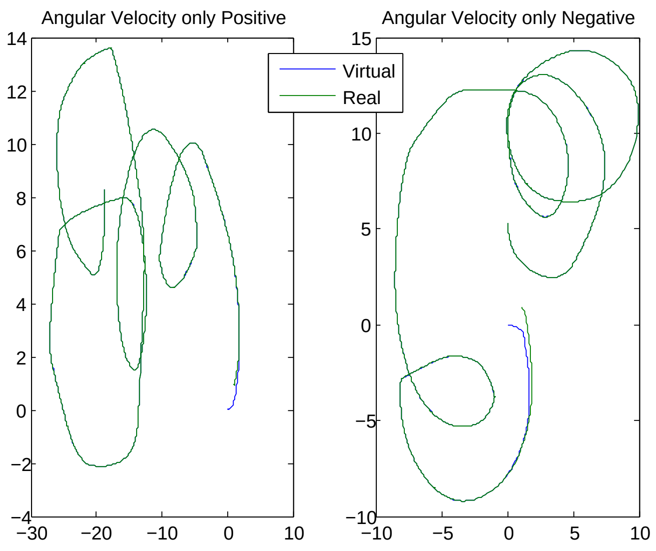

Figure 5.4: Trajectories generated using either positive or negative anglular velocities

5.3.2 Switched LPV controller

The switched LPV controller is tested once it has been checked that both families of controllers work. Figure 5.5 shows the resulting trajectories of the real vehicle compared to the reference one. The first model reaches the reference trajectory faster than the second model but both succeed in tracking the reference trajectory.

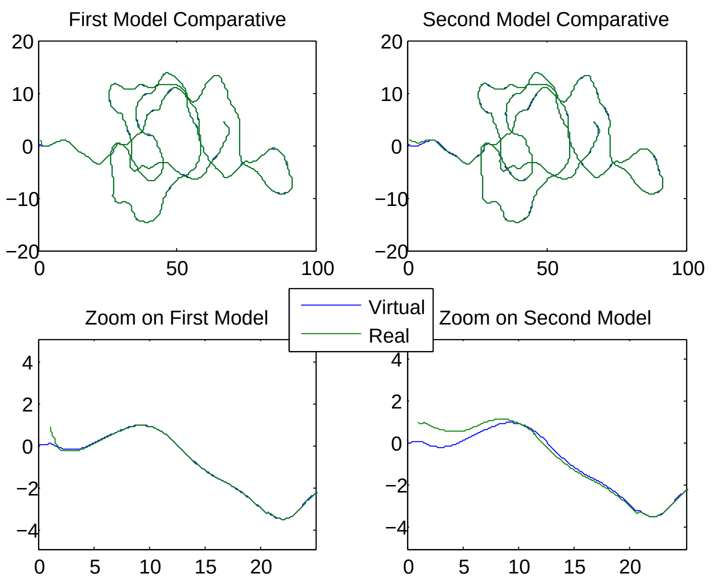

Figure 5.5: Trajectories of the two models compared to the reference trajectory

The states x_e and y_e are inside the desired design values (± 0.5) as shown in Figure 5.6 and 5.7 (Section 4.6.1). Although, in Figure 5.6, when the path is not yet reached, there are values outside the boundaries due to the fact that the initial conditions are outside the designed parameters. Both models are capable of tracking the trajectory despite having these initial conditions, which could be a hint of the robustness of the whole system.

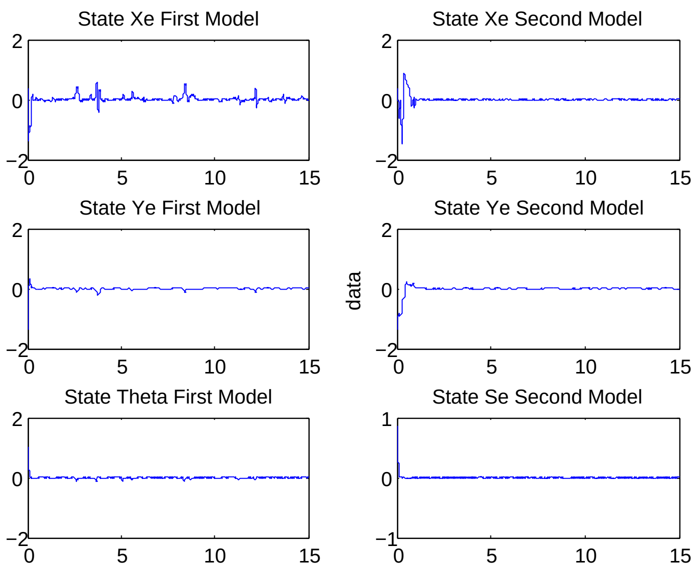

Figure 5.6: First and second model states

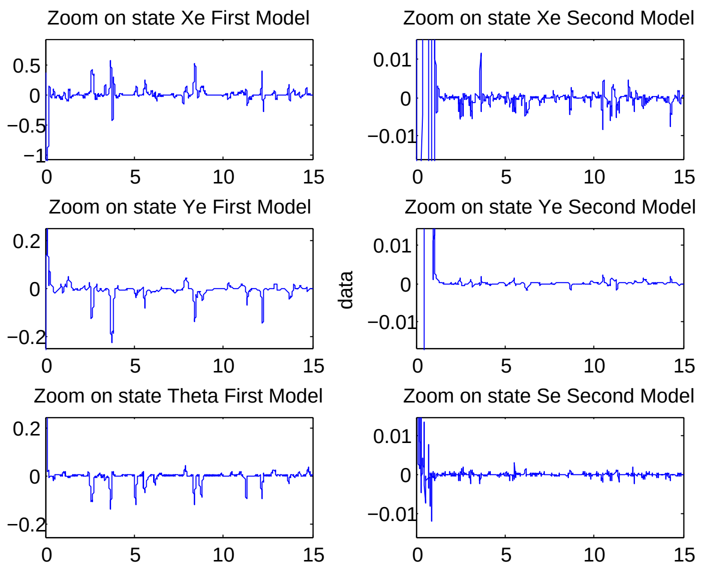

Figure 5.7: Zoom on first and second model states

The control actions that have generated the studied trajectories for both models are shown in Figure 5.8. It can be seen that the first model signals are saturated more often compared to the second model ones. This explains why the first model is rougher and the the fact that it struggles more than the second model when keeping the error low.

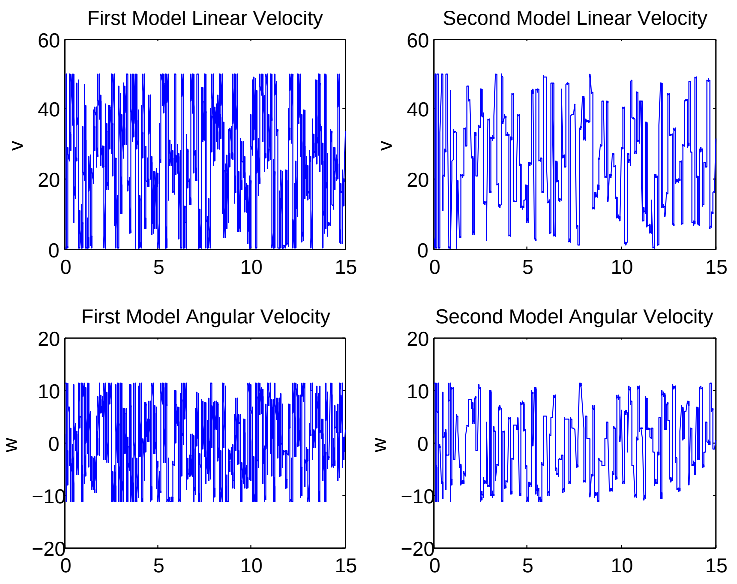

Figure 5.8: Control actions that generate the trajectories

Figure 5.9 compares the errors between the virtual and real car for both models and it confirms that the second model is smoother, with lower error values and shorter time reaching the reference trajectory.

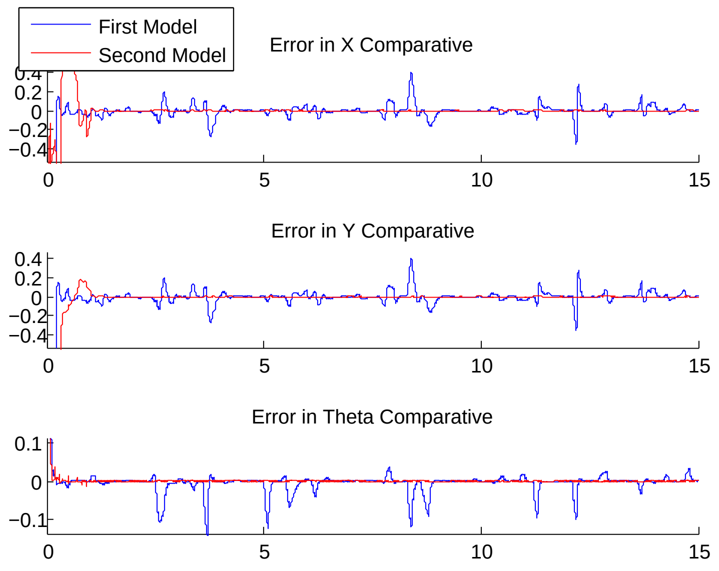

Figure 5.9: Errors comparison of the two models along the trajectory

5.3.3 Other trajectories

Two additional trajectories have been checked: straight line and a pure circle.

Straight Line

A straight line is obtained when the reference angular velocity is 0. This is the singular point of the system (view Section 4.6.2). It confirms that the first model is faster and rougher looking at Figure 5.10. In both models the error is very close to zero in the steady state.

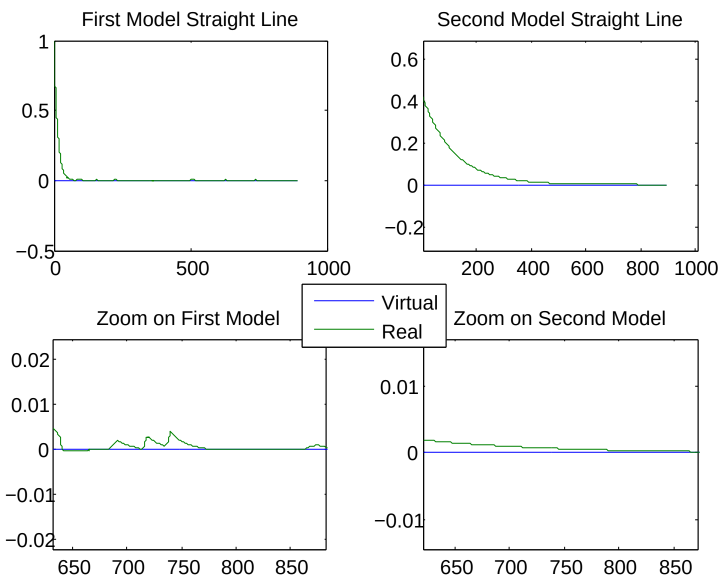

Figure 5.10: Trajectories of first and second model following a straight line

Circle

In this case, the differences in the time to reach the reference trajectory between both models are confirmed again. The first model is faster.

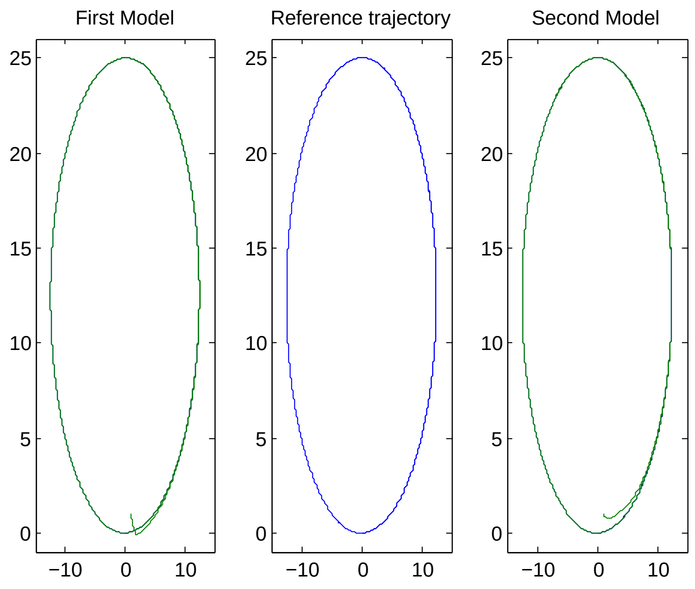

Figure 5.11: Trajectories of Model 1 and 2 following a circular trajectory

Chapter 6

Sustainability: environmental, social and economic impact

Nowadays, mobility and transportation play a very important role in our society. Autonomous vehicles could be the new paradigm for it. The way it will affect us in the long run is not determined but some clues exist. It will have an impact on the environment, our society and the economy.

Environmental impact

One of the main targets of the autonomous car developers is to reduce the energy used for travelling and enhance the comfort conditions. Fuel economy will benefit from it due to the ability of optimizing the fuel efficiency and avoiding inadequate driving style. It is expected that autonomous cars would make the traffic flow better, for instance reducing congestion in urban areas.

Social impact

In my opinion, the main social benefit of automated driving is the possible effect that it can have on the number of deaths and injuries on road. This is a fact in our society that will hopefully change drastically as autonomous cars become widespread. The majority of traffic accidents are caused by human errors. This is the reason why the majority of the developing projects in transportation are related to safety issues.

Autonomous vehicles will open new possibilities of using and saving travelling time, instead of driving the focus can be on other matters and also the car could go to the car park by itself, for example. A driverless vehicle offers the possibility of “driving” to users that are not able to drive at the present time, for instance blind people.

There is a cultural frame around cars, freedom or status are good examples of it. This culture will evolve and change as ideas of approaching transportation vary.

Economic impact

Every single concept that has already been mentioned will have an economic impact. The average fuel economy and the better treatment of vehicles will improve and therefore, their energy and maintenance expenses. Every life lost and injuries as a result of a traffic accident has a social but also a financial cost. The possibility of saving and having more free time due to transportation can be used, for instance, to enhance people and businesses’ productivity.

Companies in the transportation sector have the opportunity of cutting costs by reducing the number of drivers. This would surely cause a social impact due to job losses. Another sector that should adapt to the new paradigm is the insurance one. Insurances will no longer be about writing individual insurance policies. Some investments in infrastructures might be necessary to adapt the roads to the new way of travelling. This could be costly but in the long term more efficient, hence it is difficult to predict the economic impact because the solution that will be adopted is unknown at the moment. There might be an impact on the government accounts since there will be more active and alive people than without autonomous cars. A new legislation will be developed and it will affect the collection of fines, for instance.

On the whole, there will be deep but probably gradual changes in many aspects of our everyday lives that will produce environmental, social and economic impacts. Although, lot of people are still sceptical, autonomous applications are being introduced more often in our cars. It seems that the tendency towards fully autonomous vehicles is almost unstoppable.

Chapter 7

Conclusions and Future work

This Thesis has presented a very simple way of approaching the Motion Control procedure to solve the Path Tracking Problem for an autonomous car-like vehicle. The most basic model to represent the nonlinear behaviour of the vehicle, the bicycle model, has been adapted to a Quasi-LPV form. Only five parameters and a very simple and intuitive control law have been enough to compute a switched state feedback LPV controller, which has been designed using the polytopic transformation via nonlinear embedding and bounding box approach in order to perform the control of the system in an integrated way (lateral and longitudinal). As a result, two different synthesized controllers have been obtained, compared and proved suitable to tackle the Path Tracking Problem. Both controllers show very good results and seem very consistent in the non-conservative simulation tests performed. These promising results encourage the continuation on this path and the testing of these controllers in depth.

As a generalization, the steps to follow until the controllers are ready to be applied to a real car would be: first, transform the continuous system treated in this Thesis into a discrete one; second, test the controllers in a more accurate simulator where the environment takes the dynamics of the real car into account; and finally, implement the controllers in the real car. Nowadays, there is a project called Elektra in which an autonomous car is being developed by CVC (Computer Vision Center) - UAB (Universitat Autònoma de Barcelona) – UPC (Universitat Politècnica de Catalunya). They already have a good simulator (developed in Unity) and different controllers have been tested. This could be a good platform to continue developing the controllers proposed in this Thesis.

In addition to this, there are plenty of possibilities for further enhancement of the controllers. Some developing lines for further improvements are given here: different tuning could be applied, other or more design parameters taken into account, an observer could be added, the way they deal with uncertainty (robustness) studied, or even protect the system with a fault diagnosis and fault tolerant approach.

All in all and in my opinion, the approach followed in this Thesis to solve the Motion Control Path tracking problem for autonomous vehicles should be taken in consideration, due to its simplicity, intuitiveness and the fine simulating results obtained.

Chapter 8

Cost of the project

This budget considers the time invested, the used assets and miscellaneous expenses that have been necessary for the development of this research project.

Criteria

Time invested

The cost of the time invested has been calculated taking into account an annual base salary of 40.000€ plus 30% of it that companies pay to the Social Security in Spain. The amount of working hours in 2016 are around 1780. Therefore, the cost per hour is 29.21€ . This value is the cost, it is not what should be invoiced.

Assets

The main assets used in this project have been a computer and Matlab and Simulink software. The cost of the computer is 600€, the Matlab individual licensing is 2.000€ and the Simulink one is 3.000€. In order to compute the associated cost, the amortization time considered has been five years. The final cost is computed from the cost of the asset, the hours dedicated to the project and the amortization time.

Other costs

This section includes miscellaneous expenses like: paper, ink, energy used, travel expenses and meals when needed. This has been approximately computed due to my work base during the project being my home which is not a rented office space.

Cost breakdown

Staff costs

	Hours	€ per hour	Total €
Formation	110,00	29,21	3.213,10
Project development	420,00	29,21	12.268,20
Memory	170,00	29,21	4.965,70
			20.447,00

Assets costs

		% of a year	
	Price €	worked	Total €
Computer	600,00	39,33%	47,19
Matlab licensing	2.000,00	39,33%	157,30
Simulink licensing	3.000,00	39,33%	235,96
			440,45

Other costs

	Total €
Miscellaneous expenses	250,00

Total Cost .... 21.137.45€

Bibliography

[1] Aicardi, M., Casalino, G., Bicchi, A., and Balestrino, A. Closed loop steering of unicycle-like vehicles via lyapunov techiques. *IEEE Robotics and Automation Magazine, pp. 27-35* (1995).

[2] Alcalá, E., Sellart, L., Vicenç Puig, J. Q., Saludes, J., Vázquez, D., and López, A. Comparison of two non-linear model-based control strategies for autonomous vehicles. (2016).

[3] Blažič, S. Takagi-sugeno vs. lyapunov-based tracking control for a wheeled mobile robot. *WSEAS Transactions on Systems and Control* (2010).

[4] Casella, F., and Lovera, M. Lpv/lft modelling and identification: overview, synergies and a case study. *IEEE Multi-conference on Systems and Control* (2008).

[5] Cheng, H. *Autonomous Intelligent Vehicles.* Springer, (2011). ISBN: 978-1-4471-2279-1.

[6] Duan, G.-R., and Yu, H.-H. *LMIs in Control Systems. Analysis, Design and Applications.* CRC Press, (2013). ISBN: 978-1-4665-8300-9.

[7] García, L., and Tornero, J. Kinematic control system for car-like vehicles. *Lecture Notes in Computer Science* (2002).

[8] He, X., Dimirovski, G. M., and Zhao, J. Control of switched lpv systems using common lyapunov function method and f-16 aircraft application. *IEEE International Conference on Systems Man and Cybernetics* (2010).

[9] Hoffmann, C., and Werner, H. A survey of linear parameter-varying control applications validated by experiments or high-fidelity simulations. *IEEE Transactions on control systems technology, Vol. 23, Num. 2* (2015).

[10] Leith, D., and Leithead, W. Survey of gain-scheduling analysis and design. *International Journal of Control, 73(11), 1001–1025* (2000).

[11] Löfberg, J. A toolbox for modeling and optimization in matlab. *IEEE International Symposium on Computer Aided Control Systems Design Taipei* (2004).

[12] Löfberg, J. Yalmip website. *http://users.isy.liu.se/johanl/yalmip/*, (2015).

[13] Marcos, A., and Balas, G. J. Development of linear-parameter-varying models for aircraft. *Journal of Guidance, Control, and Dynamics, Vol. 27, No. 2, pp. 218-228* (2004).

[14] Pierre Apkarian, P. G., and Becker, G. Self-scheduled H_{inf} control of linear parameter-varying systems: A design example. *Automatica, 31(9):1251 – 1261* (1995).

[15] Rotondo, D., Puig, V., Nejjari, F., and Romera, J. A fault-hiding approach for the switching quasi-lpv fault-tolerant control of a four-wheeled omnidirectional mobile robot. *IEEE Transactions on Industrial Electronics, Vol. 62, Num. 6* (2010).

[16] Sename, O., Gáspár, P., , and Bokor, J. *Robust Control and Linear Parameter Varying Approaches.* Springer, (2013). ISBN: 978-3-642-36109-8.

[17] Shamma, J. S., and Athans, M. *Analysis and design of Gain Scheduled control systems.* PhD thesis, Massachusetts Institute of Technology, (1988).

[18] Shamma, J. S., and Cloutier, J. R. A linear parameter varying approach to gain scheduled missile autopilot design. *IEEE, American Control Conference, pp.1317-1321* (1992).

[19] Studio Freeway. *Actuation system for a vehicle controlled by CVC*, (2014).

[20] Sturm, J. F. Using sedumi 1.02, a matlab toolbox for optimization over symmetric cones. *Optimization Methods and Software, vol. 11, pp. 625-653* (1999).