Treball de Fi de Master

Master's degree in Automatic Control and Robotics

MODELLING, PLANNING AND NONLINEAR CONTROL TECHNIQUES FOR AUTONOMOUS VEHICLES

MEMÒRIA

Autor: Eugenio Alcalá Baselga
Director/s: Dr. Vicenç Puig Cayuela
Dr. Joseba Quevedo Casin
Dr. David Vázquez Bermúdez

Convocatòria: June 2016

Escola Tècnica Superior
d'Enginyeria Industrial de Barcelona

Abstract

Autonomous driving has been an important topic of research in recent years. Autonomous driving is a very challenging research topic that requires from different disciplines such as electronics, computer vision, geolocalization, control or planning. This paper tackles the problem of the vehicle control planning and performs a comparison of two nonlinear model-based control strategies for autonomous cars. These control techniques rely on the so called bicycle model and follow a reference approach. Using this approach, the error dynamics model is developed. Both controllers receive as input the longitudinal, lateral and orientation errors generating as control outputs the steering angle and the velocity of the vehicle.

The first control approach is based on a nonlinear control law that is designed by means of the Lyapunov direct approach. The second strategy is based on a sliding mode-control that defines a set of sliding surfaces over which the error trajectories will converge. The main advantage of the sliding-control technique is the robustness against non-linearities and parametric uncertainties in the model. However, the main drawback of first order sliding mode is the chattering, so it has been implemented a high-order sliding mode control.

To test and compare the proposed control strategies a quintic path planner has been implemented in order to provide the desired temporal variables to the control block. Different scenarios have been used to prove such control techniques. First both methods were proved in simulation (Matlab/Simulink and Unity[1]) and finally they were used in scenarios with a real car.

[1] https://unity3d.com/es

Acknowledgments

This joint project was developed with the Advanced Control Systems Group, Automatic Control Department of Universitat Politècnica de Catalunya (UPC) and the Computer Vision Center (CVC) in Campus UAB.

This work is supported by the Spanish MEC project TRA2014-57088-C2-1-R, by MINECO and FEDER through the project CICYT HARCRICS DPI2014-58104-R and DGT project SPIP2014-01352, by the Secretaria d'Universitats i Recerca del Departament d'Economia i Coneixement de la Generalitat de Catalunya (2014-SGR-1506). Our research is also kindly supported by NVIDIA Corporation in the form of different GPU hardware. This work was also supported by Spanish Government (Ministerio de Economía y Competitividad) and FEDER under project DPI2014-58104-R (HARCRICS).

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Autonomous driving systems have been actively researched. Many researches have shown the possibility of autonomous driving in real scenarios and several teams over the world are continuously advancing in the field. The Defense Advanced Research Projects Agency (DARPA) of the United States organises one of the most important challenges on the world: DARPA Urban Challenge. It is a competition for autonomous vehicles in which they operate in an urban environment dealing with traffic regulations and other vehicles on the course. Such a challenge incites to progress in the field.

Figure 1.1: Autonomous electric vehicle used to prove control strategies

The Computer Vision Centre (CVC) is automatizing an electric car within the context of the project *Automated and Cooperative Driving in the City (ACDC)*[1](see Figure 1.1). At this stage, environmen-

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

tal perception is mainly based on Computer Vision. In particular, while following a planned route, the obstacle-free navigable path in front of the vehicle is detected by using an on-board stereo rig. Accordingly, a short path is planned obtaining the desired set of positions and velocities. Such a set is sent to the car controller to properly execute the maneuver. Such an architecture of actions can be see in Figure 1.2.

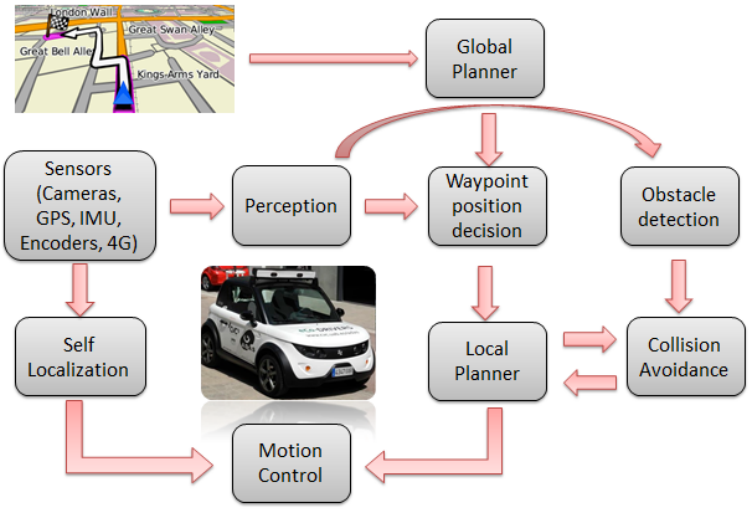


Figure 1.2: System architecture of the autonomous vehicle

This work is mainly focused on the low frame of the automatic control of the speed and the steering angle of the car following a predefined path with the best performances of stability and accuracy. A good automatic car control is the key for achieving the other challenges of the autonomous driving systems. Accordingly two strategies of non-linear automatic control are proposed, one is based on the method of Lyapunov [5, 20] and the other one is based on sliding mode [6, 9, 13, 23]. Both techniques has been tested in Matlab/Simulink over several circuits at different vehicle speeds. Moreover they have also been tested in a realistic driving simulator working over Unity 5.1.3 platform which provides a more complex scenario. Finally, one of the studied control techniques has been tested on board of a real autonomous vehicle achieving promising results. The set of software tools used in this work are showed in Figure 1.3.

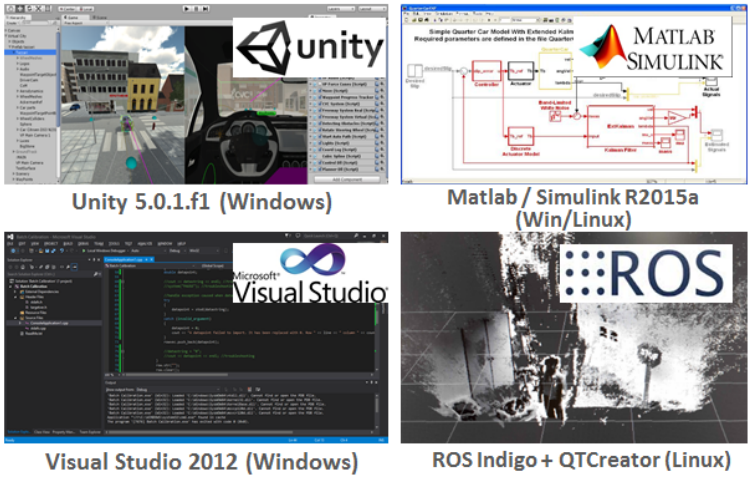


Figure 1.3: Software tools used to develop the project

1.1 Motivation

People drive their cars to work, to go shopping, to visit friends and to many other places converting the car into one of the most used modes of transport in the world. Such a vehicle may seem to be sufficiently safe but an average of 3,287 deaths a day occur using this mode of transport [1].

From the perspective of building a more robust and safer mode of transport the idea of developing autonomous vehicles emerges and that is why autonomous driving systems have been actively researched. It is composed for many engineering disciplines like mechanics, electronics, computer vision, informatics and control among others. This last one, control discipline, is the one entrusted to perform the correct vehicle motion. There exist different families of approaches inside the control theory and although linear control approaches are very suitable, nonlinear control theory has demonstrated to be a good option in autonomous driving systems being able to deal with uncertainties and lack of system information.

1.2 Thesis objectives

This work focuses on applying nonlinear control techniques in autonomous vehicles. Hence, the main objective of this thesis is to implement two non linear control laws extracted from a simple kinematic model and prove them in simulation and real scenarios. To reach the global objective some subgoals have been proposed:

- To model a 6 degrees of freedom vehicle model for simulation.

- To build a kinematic model to develop the control laws.
- To consider a dynamic model to prove the resulting control algorithms.
- To implement a trajectory planner in order to provide all desired variables to the control algorithm.
- To transfer the control laws and the trajectory planner to the realistic driving simulator implemented in Unity framework.
- To export the tested control algorithms in Matlab and Unity to Robotic Operating System (ROS)[2] environment to work within the real vehicle.

1.3 Thesis structure

This work is organised as follows:

Chapter 2

This chapter covers the current state on autonomous vehicles and the more relevant control techniques applied in the field.

Chapter 3

In this chapter, a set of vehicle models are presented. It is divided in three parts, the first one presents a complete model of the vehicle which describes almost all the dynamics of the car. The second one explains a reduced four wheels dynamic model and the last one presents the kinematic bicycle model which is very suitable for control applications due to its simplicity.

Chapter 4

In this chapter, we have studied two non-linear automatic control strategies based on the method of Lyapunov and based on sliding mode for trajectory tracking. Also a local path planner is presented which computes the path to be followed.

[2] http://www.ros.org/

Chapter 5

In this section, the results of the control methods are presented in simulation. The simulation has been developed in Matlab/Simulink and Unity 5.1.3 platforms. The performance of the control strategies are compared by means of two tests.

Chapter 6

This chapter presents a summary of the main results obtained in real situation. A discussion of the results is done and it is highlighted the advantages and drawbacks of both control methods.

Chapter 2

State of the art on autonomous vehicles

Nowadays there exist several research groups which have an almost complete autonomous vehicle which is able to deal with the complete environment. For instance in [15], the Junior Standford's autonomous vehicle is presented which was the second classified at the DARPA Urban Challenge. This vehicle uses a LIDAR to generate high-resolution maps of the environment which are used for localization and uses perception in order to recognise obstacles as cyclists, pedestrians, and other vehicles. Then, it computes a lot of candidate trajectories and its controller selects the optimal throttle, brake, and steering actions in terms of maximising comfort and minimizing trajectory error.

Another vehicle fully automatised is the Bertha Benz vehicle [27]. It uses vision, radar sensors and GPS/INS together with accurate maps to obtain an accurate localization and to understand the environment. This vehicle has implemented a feed-back lateral controller and a model predictive controller for speed control [11]. In 2014, it completed its first journey covering a trajectory which had a length of 103 km passing trough rural and urban roads.

Google has been testing its autonomous vehicle in actual traffic condition. In August 2012, Google announced that they have completed over 500,000km autonomous driving without any accident [25]. Another research group in Italy, the VisLab in Parma University did 13,000 km test run for autonomous vehicles from Italy to China [3]. They focus on the perception of the surrounding environment by camera sensors and fusion with other sensors. The VisLab also tested the autonomous vehicle in a real environment, together with real traffic on July 2013.

In May 2014, Google presented a new concept for their driverless car that had neither a steering wheel nor pedals [8], and unveiled a fully functioning prototype in December of that year that they planned to

test on San Francisco Bay Area roads beginning in 2015. Google plans to make these cars available to the public in 2020 [12].

The trajectory tracking control of an autonomous vehicle is one of the most difficult automation challenges due to its motion constraints: linear and angular speeds, linear and angular accelerations, etc; and there exist a constant environment interaction.

Under the control point of view several control approaches have been developed and discussed in the literature. In [16] a nested PID steering control for lane keeping in vision based autonomous vehicles is designed to perform path following in the case of roads with an uncertain curvature. In [26] a control algorithm based on adaptive PID is developed obtaining very good results in a real scenario. In [14] a control technique is developed based on the finite preview optimal control method. In [23] a sliding mode control algorithm for vehicle stability enhancement were proposed and evaluated in simulation. In [24] a design and experimental validation of a vehicle lateral controller for autonomous vehicle based on a higher-order sliding mode control is presented. In [17] an adaptive sliding mode controller for yaw stabilisation was proposed obtaining satisfactory results over a real vehicle. In [4, 10, 11] the model predictive control technique is presented for autonomous vehicles. Such a strategy is an attractive method to generate feasible trajectories and robustly track them. This technique is also very useful since it allows to generate a set of optimal control actions according with a set of constraints.

Chapter 3

Mathematical model of the vehicle

There are many approaches when one wants to model the dynamics of a vehicle. In the literature it is very common to find three models: the kinematic bicycle model, the dynamic bicycle model and the four wheels dynamic model. The majority of researchers work with dynamic models since it is sufficiently representative of vehicle dynamics. There exists many approaches, some models consider motor dynamics, brake system dynamics and roll and pitch motion and others only represent the vehicle motion dynamics. In this project, the development of a quite complete four wheels vehicle model has as objective the implementation of a simulator in which to test the control algorithms. A dynamic four wheels model is developed as a simplification of the last one. The idea of such a model is to use it to test kinematic control algorithms and as a future work to perform dynamic control algorithms. The dynamic bicycle model is not presented since it is quite similar to the four wheels one. Finally, the kinematic model is the simplest because it does not take into account some dynamics like $roll$, $pitch$ and z motion among others. It is a very used model for control applications due to its simplicity.
This chapter is divided in three parts. The first one presents a complete model of the vehicle which describes almost all the dynamics of the car. The second one explains a reduced four wheels dynamic model and the last one presents the kinematic bicycle model.

3.1 Six degrees of freedom vehicle model

In this section, the four wheels dynamic model is studied. This one considers the six DOF in the car which means that it takes in consideration all possible movements. In order to find out the set of equations that describes the complete system it is necessary to understand the vehicle motion in the plane X-Y, to study

the vehicle roll, pitch and vertical motions and to understand the drivetrain dynamics, the brake dynamics and the engine dynamics.

We assume a set of constraints to further simplify the model:

- The vehicle works at ambient temperature (25^{o}C) and ambient pressure (105 KPa).
- The road is considered as smooth dry asphalt.
- The tires are assumed like a rigid body.
- The steer angle is assumed to be equal in both front wheels.
- The vehicle is dealt as a system without spring-damper system.
- The engine is considered as a DC motor instead of a trifasic motor.
- A clutch and gear box systems are not considered since our vehicle is electric.
- It has not been considered any energy consumption and batteries capacity in the model.

The explanation of the model has been divided in modules. The first one explains the engine model, the second one the drivetrain model, the third the brake system, then the wheels dynamics, next the vehicle dynamics and the last one presents the tire model.

3.1.1 Engine dynamics

The first approach is to consider the electric motor as a DC motor. This kind of motor converts electric energy from the batteries into mechanical energy to the driveline. In the transformation, a set of energy losses occurs due to friction among others hence a efficiency parameters is used to compute the power balance. The best way of obtaining the output power from an input power is through the power map or power curves. All motors have its particular power curve. For our case, the motor curves are known and they are represented in Figure 3.2. Such an output power is a relation among torque and angular velocity and it can only have a unique value since it represents a point in the power curve. The equation that describes the power balance in the DC motor is presented as follows:

$$I_{in} \cdot V_{in} \cdot \eta_{loss} = \tau_m \cdot \omega_m \tag{3.1}$$

where in the left hand of the equation the η_{loss} represents the efficiency of the motor, I_{in} and V_{in} are the current and voltage at the input respectively. On the other hand τ_m and ω_m represent the torque and angular speed respectively.

Figure 3.1: Electric motor model diagram

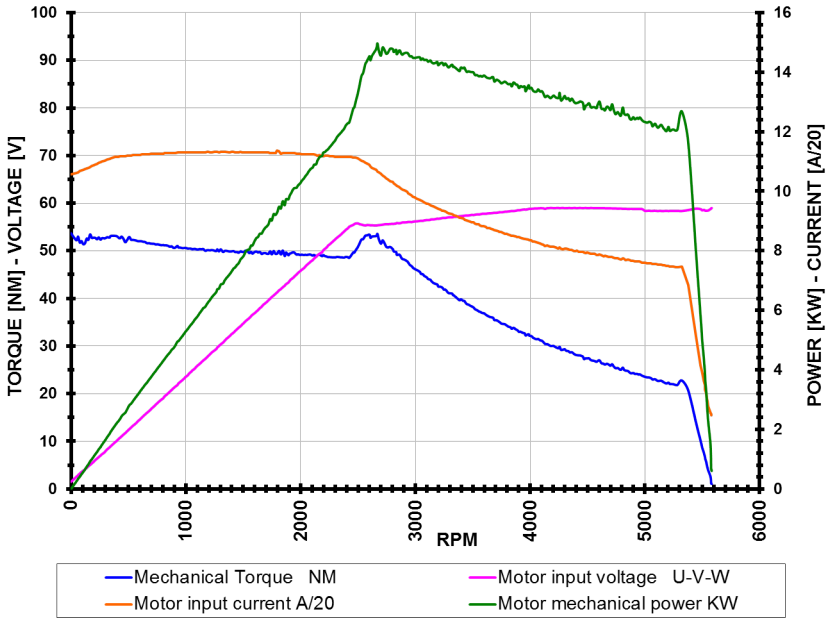


Figure 3.2: Power, torque, current and voltage curves of Tazzari electric motor

It is important to know the maximum and minimum values of torque and angular speed that the motor can offer to the driveline, and it is also interesting to know the limits of voltage and current that it can be applied to the engine.

3.1.2 Drivetrain model

This model has as input the output power of the motor. The torque at the output of the transmission is proportional to the torque produced by the engine and the efficiency also has to be taken into account since there exist friction losses. Note that due to the car is a rear-wheel driven vehicle the output drivetrain

torque is applied only to rear wheels. Such a torque is computed as:

$$\tau_{drive_i} = \frac{\tau_m \cdot \eta_{drive} \cdot R_{drive}}{2} \; ; \quad i = RL, RR \tag{3.2}$$

where η_{drive} is the transmission efficiency, R_{drive} is the gear transmission ratio and the RL and RRmean rear left and rear right, respectively.

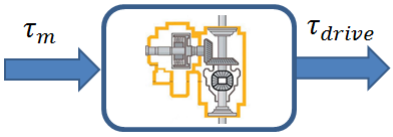


Figure 3.3: Drivetrain model diagram

3.1.3 Brake model

The vehicle has a modified hydraulic brake system which has disc brakes in the four wheels. It is important to limit the possible pressure that one can apply to the brakes, at least to the rear brakes in order that the rear wheels cannot skid.

The torque produced by the brake system in each wheel is determined by the following expression:

$$\tau_{brake_i} = \frac{\frac{\pi}{4} \cdot D_{piston}^2}{A_{pad}} \cdot 20 \cdot (0.58 \cdot D_{disc}) \cdot Press_{brake} \cdot \pi \cdot C_{pad} \cdot (D_{disc}^2 - (0.58 \cdot D_{disc})^2) \; ; \quad i = 1,2,3,4 \tag{3.3}$$

where $Press_{brake}$ is the pressure that we apply to the first valve (Master cylinder) and represent the input of the brake system. D_{piston} is the calliper piston diameter, A_{pad} is the brake pad area, C_{pad} is the friction coefficient between the disc and the pad and D_{disc} is the brake disc diameter. Hence, a simplification of (3.3) can be achieved by considering all constant factors as a unique constant factor as follows:

$$\tau_{brake_i} = K \cdot Press_{brake} \; ; \quad i = 1,2,3,4 \tag{3.4}$$

Figure 3.4 represents the input-output diagram and Figures 3.5 and 3.4 show the brake curve for rear and front wheels. Having these curves is interesting because knowing the force applied to the pedal, the output torque can be computed directly without having to deal with brake system parameters.

Figure 3.4: Brake model diagram

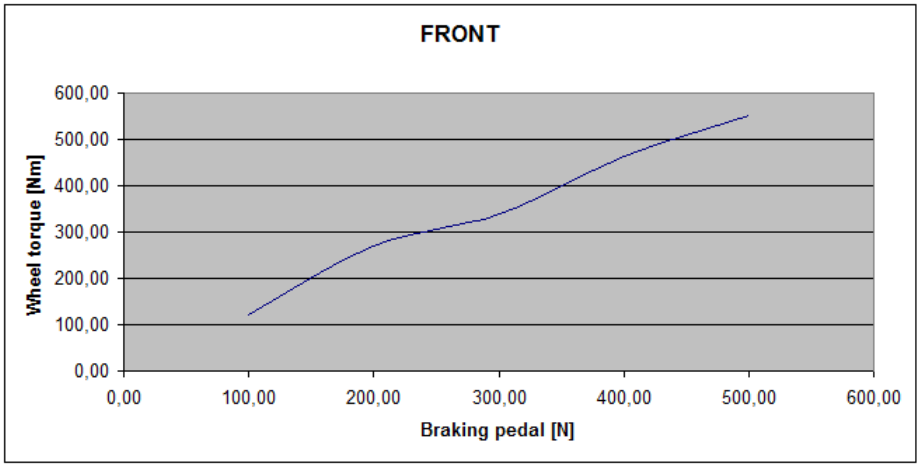


Figure 3.5: Brake curve of front wheels

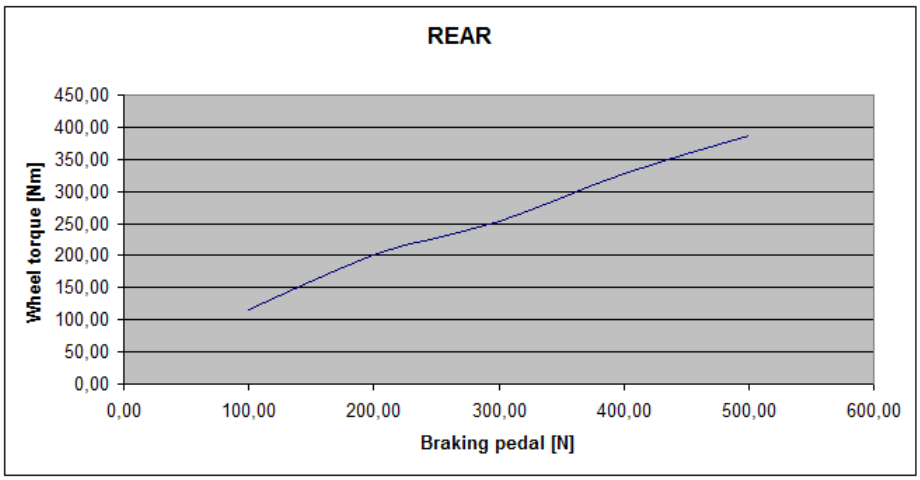


Figure 3.6: Brake curve of rear wheels

3.1.4 Wheels dynamics

The forces acting on a wheel produce a set of torques which determines the wheel rotational velocity, if these torques are not balanced it appears an angular acceleration. The equation that describes this is the

following:

$$\tau_{drive_i} - \tau_{brake_i} = I_{wheel} \cdot \alpha_{wheel_i} \; ; \quad i = 1,2 \tag{3.5}$$

where I_{wheel} represents the wheel inertia with respect to its axis of rotation and α_{wheel} denotes the angular acceleration in the wheel.

Figure 3.7: Wheel dynamic model diagram

Until now it has been presented the vehicle model from the actuators (electric engine and brake pedal) to the resultant angular acceleration at each wheel (see Figure 3.8). With such acceleration, the angular velocity is obtained which will be the input for the next two sections.

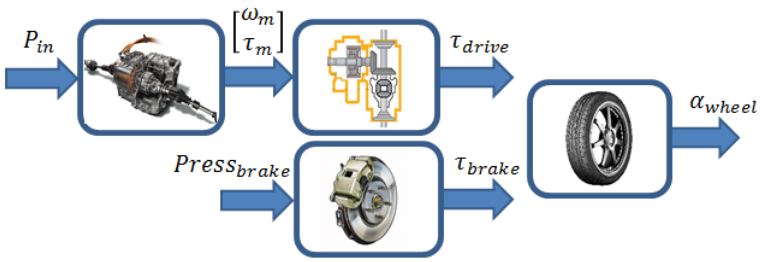


Figure 3.8: Vehicle model diagram from motor and brake actuators to wheels

3.1.5 Vehicle motion

Once one takes into account all the considerations done above, it is possible to write the set of equations which govern the vehicle's motion [7].

The way to achieve the dynamic equations is by looking for the force and torque equilibrium in every axis. We are going to denote the X axis as the longitudinal car axis, the Y axis as the lateral one and the Z axis as the perpendicular to the road. Such a XYZ frame is in the centre of gravity of the vehicle which from now will be named CoG. The equations that describe the equilibrium of longitudinal forces

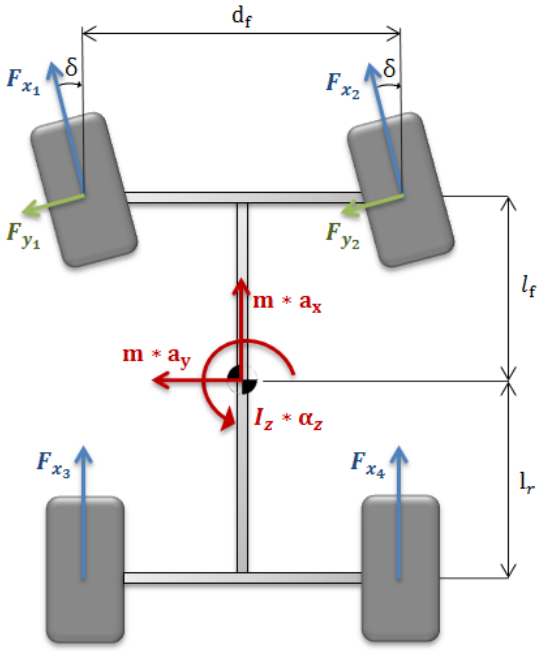


Figure 3.9: Resultant forces in the vehicle from a top view

are provided below:

$$\begin{aligned} F_{x_1} \cdot cos(\delta) - F_{y_1} \cdot sin(\delta) \\ + F_{x_2} \cdot cos(\delta) - F_{y_2} \cdot sin(\delta) \\ + F_{x_3} + F_{x_4} - F_{drag} - F_{rolling} - F_{slope} = m \cdot a_x \end{aligned} \tag{3.6}$$

where the forces F_{x_i} are the longitudinal wheel forces represented in Figure 3.9 as blue force vectors, the forces F_{y_i} are the lateral wheel forces represented as green force vectors (both wheel forces will be presented in Tire model section), the term δ_i represents the steer angle of front wheels, a_x is the inertial acceleration along the CoG X axis and the rest of terms are defined below.

The acceleration term can be computed as:

$$a_x = \dot{v}_x + v_y \cdot \omega_z \tag{3.7}$$

where $\dot{v}_x$ is the acceleration due to the motion along the X axis and the product $v_y \cdot \omega_z$ represent the centripetal acceleration produced when the car is turning.

The aerodynamic force is stated as:

$$F_{drag} = 1/2 \cdot \rho_{air} \cdot C_{drag} \cdot A_{front} \cdot v_x^2 \cdot (sign(v_x)) \tag{3.8}$$

where ρ_{air} is the density of the air (1.2 kg/m3 at normal temperature and pressure), the C_{drag} represents the drag coefficient, the A_{front} term is the frontal area of the vehicle and the last term v_x represents the flow velocity that in this case corresponds to the vehicle velocity since the wind flow velocity cannot be measured.

The rolling resistance is computed as follows:

$$F_{rolling} = f_r \cdot m \cdot g \cdot min(1, v_x) \cdot (sign(v_x)) \tag{3.9}$$

where the term f_r is the rolling resistance coefficient, m is the mass of the car and g is the gravity acceleration.

When the vehicle is going uphill or downhill a new force appears due to the road slope:

$$F_{slope} = m \cdot g \cdot sin(\beta_x) \cdot (sign(v_x)) \tag{3.10}$$

where the β_x means the road tilt.

Regarding the equations that defines the lateral dynamics they are similar to the longitudinal ones. The equilibrium of lateral forces is given by:

$$\begin{aligned} F_{x_1} \cdot sin(\delta) - F_{y_1} \cdot cos(\delta) \\ + F_{x_2} \cdot sin(\delta) - F_{y_2} \cdot cos(\delta) \\ - F_{tilt} = m \cdot a_y \end{aligned} \tag{3.11}$$

where the forces F_{x_i} are the longitudinal wheel forces represented in Figure 3.9 as blue force vectors, the forces F_{y_i} are the lateral wheel forces represented as green force vectors and the term δ represents the steer angle of front wheels. The rest of the terms are explained down below.

The lateral acceleration can be written as:

$$a_y = \dot{v}_y + v_x \cdot \omega_z \tag{3.12}$$

The last term (F_{tilt}) is only taken into account when the road is tilted a long the lateral axis, it means when the roll angle is not zero.

$$F_{tilt} = m \cdot g \cdot sin(\beta_y) \tag{3.13}$$

where β_y is the tilt angle of the road.

Until now we have described the linear movements in X and Y axis. Now the angular motion over Z axis is going to be explained. Such a motion is also known as yaw motion.

Thus, the equation that describes the yaw movement is the following equilibrium of torques:

$$\begin{aligned} &(F_{x_1} \cdot sin(\delta) + F_{y_1} \cdot cos(\delta)) \cdot l_f \\ +&(F_{x_2} \cdot sin(\delta) + F_{y_2} \cdot cos(\delta)) \cdot l_f \\ &+(F_{y_3} + F_{y_4}) \cdot l_r = I_z \cdot \alpha_z \end{aligned} \tag{3.14}$$

where variables still not defined are: l_f which represents the distance along X axis from the front wheels coordinate frame to the CoG, l_r is the distance along the X axis from the rear wheels coordinate frame to the CoG, I_z represents the inertia which is the resistance of the car to the yaw motion and α_z is the angular acceleration of the car over the Z axis.

Note that the longitudinal rear forces have been neglected in the (3.14). This is due to the fact that both rear wheels produce more or less the same force and therefore the torque in the CoG is null.

The notation used for the orientation, velocity and acceleration over the Z axis and its relation is given by:

$$\alpha_z = \dot{\omega}_z \tag{3.15}$$

$$\omega_z = \dot{\theta}_z \tag{3.16}$$

The roll motion depends on the lateral car accelerations. Thus, when the car takes a curve it rolls around the roll axis and it is compensated by two suspension systems (two springs and two dampers). It is very important to notice that the forces generated in the vehicle are over the car's CoG. However, the roll and pitch axes are separated from such a CoG. Therefore there exist two distances between both called d_{roll} and d_{pitch}. This is due to the CoG position depends on the mass distribution of the car while roll and pith frame position depends on the vehicle geometry. The following equation represents the balance over the

roll axis:

$$(I_x + m \cdot d_{roll}^2) \cdot \alpha_x + m \cdot d_{roll} \cdot \alpha_y + (K_{roll_f} + K_{roll_r} - m \cdot g \cdot d_{roll}) \cdot \theta_{roll} + (D_{roll_f} + D_{roll_r}) \cdot \omega_x = 0 \; ; \quad (3.17)$$

$$\omega_x = \dot{\theta}_{roll} \quad (3.18)$$

$$\alpha_x = \dot{\omega}_x \quad (3.19)$$

where I_x is the moment of inertia of the car over the CoG X axis, K_{roll_f} and K_{roll_r} are the front and rear torsion spring stiffness coefficients, θ_{roll} is the road tilt angle, D_{roll_f} and D_{roll_r} are the front and rear suspension shock absorber damping coefficients, a_y represents the linear acceleration in the CoG Y axis and α_x and ω_x are the angular acceleration and the angular velocity over X axis respectively.

Applying the same idea one can write a similar torque equilibrium equation for pitch motion:

$$(I_y + m \cdot d_{pitch}^2) \cdot \alpha_y - m \cdot d_{pitch} \cdot a_x + (K_{pitch} + m \cdot g \cdot d_{pitch}) \cdot \theta_{pitch} + D_{pitch} \cdot \omega_y = 0 \; ; \quad (3.20)$$

$$\omega_y = \dot{\theta}_{pitch} \quad (3.21)$$

$$\alpha_y = \dot{\omega}_y \quad (3.22)$$

where in this case the variables and coefficients refer to rotation over Y axis.

The vertical movement is based on knowing the vertical displacement of the car's CoG when the car takes a curve. The following equation shows the relationship between the position of the COG in Z and the vehicle angles pitch and roll:

$$z = COG_z + d_{pitch} \cdot (cos(\theta_{pitch}) - 1) + d_{roll} \cdot (cos(\theta_{roll}) - 1) \quad (3.23)$$

$$v_z = \dot{z} \quad (3.24)$$

$$a_z = \dot{v}_z \quad (3.25)$$

Before introducing the tire model, it is suitable to present and define some important concepts. The first

one is the angle between the velocity vector and the longitudinal axis of the car defined as:

$$\begin{aligned} \theta_{v_f} &= \frac{\dot{y}+l_f \cdot \omega_z}{v_x} \\ \theta_{v_r} &= \frac{\dot{y}-l_r \cdot \omega_z}{v_x} \end{aligned} \tag{3.26}$$

where θ_{v_f} refers to the front tire, θ_{v_r} refers to the rear tire, l_f and l_f are the distances from the CoG to the front and rear axes respectively, ω_z represents the yaw velocity, v_x is the longitudinal axis velocity and $\dot{y}$ is the lateral acceleration.

The second one is the longitudinal and lateral velocities at each tire. Let us consider an scenario including a right curve in a road. When the vehicle turns in order to follow the road, the set of left wheels are rotating faster than the right set. Thus, a new geometric based velocities need to be calculated for each wheel as follows:

$$\begin{aligned} v_{x_1} &= v_x - \omega_z \cdot \frac{d_f}{2} \\ v_{x_2} &= v_x + \omega_z \cdot \frac{d_f}{2} \\ v_{x_3} &= v_x - \omega_z \cdot \frac{d_r}{2} \\ v_{x_4} &= v_x + \omega_z \cdot \frac{d_r}{2} \end{aligned} \tag{3.27}$$

where v_x is the longitudinal velocity of the vehicle, ω_z is the angular speed of the vehicle and w_{front} and w_{rear} are the distance between front wheels and rear wheels respectively.

3.1.6 Tire model

The tire model is one of the most important parts in vehicle modelling, since they are the unique component keeping the vehicle in contact with the road. This model computes the resultant forces between the road and the wheels that are used by Newton's laws (Equations 3.6, 3.11 and 3.14) to perform the correct vehicle motion. Resultant lateral forces will be described first, finally longitudinal forces will be presented [19].

Experimental results show that the lateral tire force is proportional to the slip angle for small slip angles. The tire slip angle is known as the difference among the wheel angle and the velocity vector angle such as it can be seen in Figure 3.10.

The front and rear tire slip angles can be computed as:

$$\begin{aligned} \alpha_f &= \delta - \theta_{v_f} \\ \alpha_r &= -\theta_{v_r} \end{aligned} \tag{3.28}$$

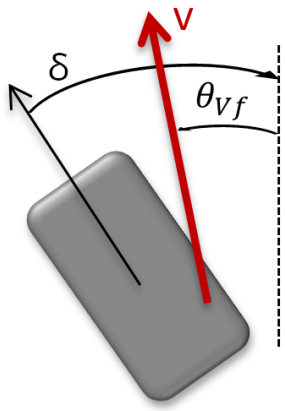


Figure 3.10: Tire slip angle where the dashed line represents the longitudinal axis of the vehicle

where δ represents the wheel angle whit respect to the longitudinal axis of the vehicle, as well known as steering angle and θ_{v_f} and θ_{v_r} are the velocity vector orientation defined in (3.26).
Thus, the lateral tire force in front and rear wheels can be expressed as:

$$\begin{aligned} F_{y_f} &= C_{\alpha f} \cdot \alpha_f \qquad f = 1,2 \\ F_{y_r} &= C_{\alpha r} \cdot \alpha_r \qquad r = 3,4 \end{aligned} \tag{3.29}$$

where the term $C_{\alpha f}$ is known as cornering stiffness which is function of tire pressure, vertical tire load and size and shape of the tire.

Such a cornering stiffness is typically less than 0.1 on dry surface, but it is important to take into account that it is a normalised value. In [21], an estimation of the cornering stiffness approach is explained.

Regarding the longitudinal tire forces, they mainly depend on the slip ratio and the friction coefficient. Before to present the equations of longitudinal tire forces it is convenient to define the effective radius of the wheel as follows:

$$r_{eff} = \frac{sin(\beta)}{\beta} \cdot r_{nom} \tag{3.30}$$

$$\beta = cos^{-1}(\frac{r_{st}}{r_{nom}}) \tag{3.31}$$

where the variables r_{st} and r_{nom} are the radius when the car is static and the nominal radius respectively, as represented in Figure 3.11.

The longitudinal tire slip is the difference among the linear velocity of the wheel computed as $r_{eff}\omega_w$

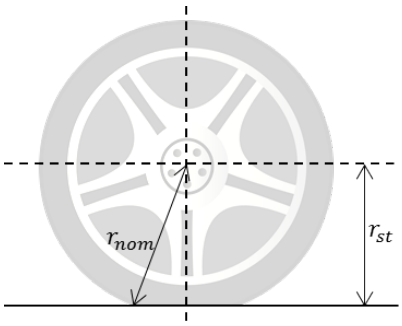


Figure 3.11: Calculation of effective tire radius

and the velocity v_{x_i} according 3.27. The tire slip is presented as a ratio:

$$\sigma_x = \frac{r_{eff}\omega_w - v_x}{r_{eff}\omega_w} \tag{3.32}$$

Now the longitudinal tire forces can be defined:

$$\begin{aligned} F_{x_f} &= C_{\sigma f} \cdot \sigma x_f \qquad f = 1,2 \\ F_{x_r} &= C_{\sigma r} \cdot \sigma x_r \qquad r = 3,4 \end{aligned} \tag{3.33}$$

where $C_{\sigma f}$ and $C_{\sigma r}$ represent the tire stiffness for front and rear tires respectively. Normally such parameters move on the interval from 30 to 80 N/m.

3.2 Dynamic vehicle model

In this section, a reduced nonlinear four wheels dynamic model is considered. It is a reduction of the one presented in the last section in which a four wheels vehicle was considered. This model assumes that the vehicle is symmetric, the *roll* and *pitch* movement are neglected and the linear movement in z axis is also neglected. Since the electric motor dynamics, drivetrain, brake system and wheel dynamics are not considered, the input of the dynamic model model will be the steering angle and the angular speed of the

wheels. The model considered is governed by the following equations:

$$\begin{cases} \ddot{x} = -\dot{y}\cdot\dot{\theta} + \frac{1}{m}\Big[(F_{x_1}+F_{x_2})\cdot cos(\delta) - (F_{y_1}+F_{y_2})\cdot sin(\delta) + F_{x_3} + F_{x_4} - F_{drag} - F_{rolling} - F_{slope}\Big] \\ \ddot{y} = -\dot{x}\cdot\dot{\theta} + \frac{1}{m}\Big[(F_{x_1}+F_{x_2})\cdot sin(\delta) - (F_{y_1}+F_{y_2})\cdot cos(\delta) + F_{y_3} + F_{y_4} - F_{tilt}\Big] \\ \ddot{\theta} = \frac{1}{I_z}\Big[(F_{x_1}+F_{x_2})\cdot sin(\delta)\cdot l_f - (F_{y_1}+F_{y_2})\cdot cos(\delta)\cdot l_f + (F_{y_3}+F_{y_4})\cdot l_r\Big] \end{cases} \tag{3.34}$$

where such a set of equations have been already defined in last section. By choosing the position, orientation and linear and angular velocities as the state variables it can be formed the state vector:

$$x = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \end{bmatrix} = \begin{bmatrix} x \\ y \\ \theta \\ \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} \tag{3.35}$$

Note that in this model the inputs are the four tire forces F_{x_i} and the steering angle in front wheels δ. The nonlinear state space model can be written as:

$$\dot{x} = f(x) + g(x,u) \tag{3.36}$$

where $f(x)$ is a vector which only depends on the states and $g(x,u)$ depends on both, the states and the inputs. The state space representation in matrix form is as follows:

$$\dot{x} = \begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \\ \dot{x}_4 \\ \dot{x}_5 \\ \dot{x}_6 \end{bmatrix} = \begin{bmatrix} x_4 \\ x_5 \\ x_6 \\ -x_5x_6 + \frac{1}{m}(-F_{drag} - F_{rolling} - F_{slope}) \\ -x_4x_6 + \frac{1}{m}(-F_{tilt} - F_{y_3} - F_{y_4}) \\ \frac{l_r}{I_z}(F_{y_3} + F_{y_4}) \end{bmatrix} + \tag{3.37}$$

$$+\begin{bmatrix} 0 \\ 0 \\ 0 \\ \frac{1}{m}\left[(F_{x_1}+F_{x_2})\cos(\delta)-(F_{y_1}+F_{y_2})\sin(\delta)+F_{x_3}+F_{x_4}\right] \\ \frac{1}{m}\left[(F_{x_1}+F_{x_2})\sin(\delta)-(F_{y_1}+F_{y_2})\cos(\delta)\right] \\ \frac{l_f}{I_z}\left[(F_{x_1}+F_{x_2})\sin(\delta)-(F_{y_1}+F_{y_2})\cos(\delta)\right] \end{bmatrix}$$

But due to the fact that F_{x_i} depend on wheels velocity, a transformation in the input vector is applied as follows:

$$u=\begin{bmatrix} F_{x_1} \\ F_{x_2} \\ F_{x_3} \\ F_{x_4} \\ \delta \end{bmatrix} \Longrightarrow u=\begin{bmatrix} u_1 \\ u_2 \end{bmatrix}=\begin{bmatrix} \omega_w \\ \delta \end{bmatrix} \tag{3.38}$$

Such transformation has been made by using (3.32) and (3.27), as illustrated in Figure 3.12.

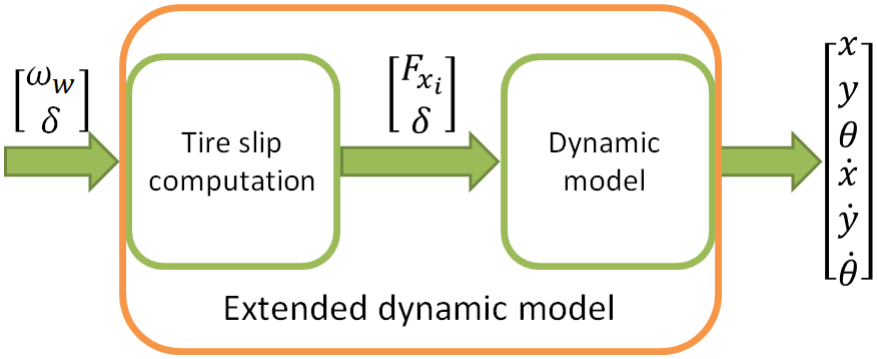


Figure 3.12: Dynamic model diagram showing the transformation from angular velocity on wheels ω_w to longitudinal tire forces F_{x_i}

Hence, the extended dynamic model can be written as:

$$\dot{x} = \begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \\ \dot{x}_4 \\ \dot{x}_5 \\ \dot{x}_6 \end{bmatrix} = \begin{bmatrix} x_4 \\ x_5 \\ x_6 \\ -x_5 x_6 - \frac{sign(x_4)}{m}(\frac{1}{2}\rho_{air} C_{drag} A_{front} x_4^2 + f_r mg + mgsin(\beta_x)) \\ -x_4 x_6 + \frac{1}{m}(2C_{\alpha r}(-\frac{x_5 - l_r x_6}{x_4}) - g\sin(\beta_y)) \\ \frac{l_r}{I_z}(\frac{2C_{\alpha r}}{m}(-\frac{x_5 - l_r x_6}{x_4})) \end{bmatrix} + \tag{3.39}$$

$$+ \begin{bmatrix} 0 \\ 0 \\ 0 \\ \frac{1}{m}\left[C_{\sigma_f}(\frac{2r_{eff}u_1 - 2x_4}{r_{eff}u_1})\cos(u_2) - (2C_{\alpha f}(u_2 - \frac{x_5 - l_f x_6}{x_4}))\sin(u_2) + C_{\sigma_f}(\frac{2r_{eff}u_1 - 2x_4}{r_{eff}u_1})\right] \\ \frac{1}{m}\left[C_{\sigma_f}(\frac{2r_{eff}u_1 - 2x_4}{r_{eff}u_1})\sin(u_2) - 2C_{\alpha f}(u_2 - \frac{x_5 - l_f x_6}{x_4})\cos(u_2)\right] \\ \frac{l_f}{I_z}\left[C_{\sigma_f}(\frac{2r_{eff}u_1 - 2x_4}{r_{eff}u_1})\sin(u_2) - 2C_{\alpha f}(u_2 - \frac{x_5 - l_f x_6}{x_4})\cos(u_2)\right] \end{bmatrix}$$

3.3 Kinematic vehicle model

In this section, a kinematic bicycle model will be presented. One of the major differences in the kinematic model with respect to the dynamic model is the null skidding assumption. There exist another important assumption which is the consideration of small lateral forces. Both assumptions share the idea of low speed, therefore it can be said a kinematic model presents acceptable results when the vehicle goes at low speed. Another difference is that this model does not consider neither the mass nor the inertia between others. In [22], it can be seen a complete development of such a kinematic model.

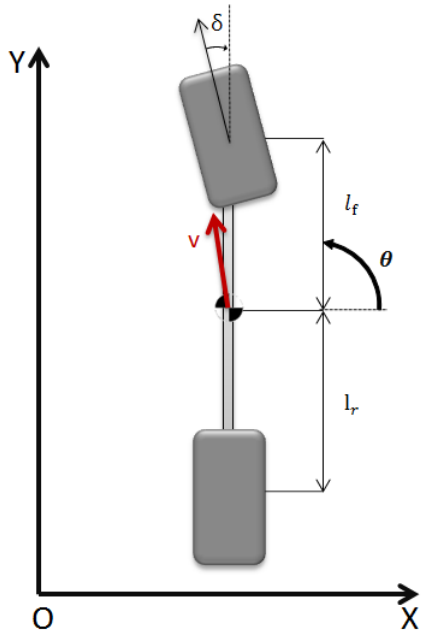


Figure 3.13: Kinematic biicycle model representation

Thus, the set of kinematic equations of the cartesian position (x, y) and orientation (θ) of the real vehicle are presented as follows:

$$\begin{cases} \dot{x} = v \cdot sin(\theta) \\ \dot{y} = v \cdot cos(\theta) \\ \dot{\theta} = \frac{v}{l_f} \cdot tan(\delta) \end{cases} \tag{3.40}$$

where v and δ represent the CoG linear velocity and the steering angle, respectively (see Figure 3.13).

In this section, an error model between the virtual and real vehicle is going to be developed. The kinematic equations for the virtual car can be defined as:

$$\begin{cases} \dot{x}_d = v_d \cdot sin(\theta_d) \\ \dot{y}_d = v_d \cdot cos(\theta_d) \\ \dot{\theta}_d = \frac{v_d}{l_f} \cdot tan(\delta_d) \end{cases} \tag{3.41}$$

where x_d, y_d and θ_d are the position and orientation of the next way-point generated by the trajectory planner, that will be explained in Chapter 4.

The error vector has been defined as the difference between real measurements and desired values mul-

tiplied by the rotation matrix over Z axis which is the orthogonal to the road plane:

$$\begin{bmatrix} x_e \\ y_e \\ \theta_e \end{bmatrix} = \begin{bmatrix} cos\theta_d & sin\theta_d & 0 \\ -sin\theta_d & cos\theta_d & 0 \\ 0 & 0 & 1 \end{bmatrix} * \begin{bmatrix} x - x_d \\ y - y_d \\ \theta - \theta_d \end{bmatrix} \tag{3.42}$$

The derivation of the error vector gives the following open loop error system which will be used later to develop the control techniques:

$$\begin{cases} \dot{x}_e = v \cdot cos(\theta_e) + y_e \cdot \frac{v_d}{l_f} \cdot tan(\delta_d) - v_d \\ \dot{y}_e = v \cdot sin(\theta_e) - x_e \cdot \frac{v_d}{l_f} \cdot tan(\delta_d) \\ \dot{\theta}_e = \frac{v}{l_f} \cdot tan(\delta) - \frac{v_d}{l_f} \cdot tan(\delta_d) \end{cases} \tag{3.43}$$

Chapter 4

Non-linear control strategies

In this chapter, we will present two nonlinear automatic control strategies: one based on the method of Lyapunov [5, 20] and the other one based on sliding mode [6, 9, 13, 23] for path following and navigation among way-points. These techniques are two of the most interesting techniques in nonlinear control and robust control. The basic idea of nonlinear control based on Lyapunov method is to define a control law assuring the stability and the asymptotic elimination of the following error of the real car to follow the prespecified way points. And, the basic idea of SMC is to reach a sliding surface in finite time and remain on this. But there exist a drawback: the chattering, that consists on a trajectory oscillation over the sliding surface. There exist many ways to reduce and eliminate this problem. One of them is by using a higher order sliding surface. Another one is by using smooth functions instead of the sign function and there are more approaches but not considered in this work. For implementing the controllers, a trajectory planner [2, 23] has been implemented and a brief explanation of this is made in this chapter. Thus this chapter explains first the trajectory planner block and then the techniques used in the control block (see Figure 4.1).

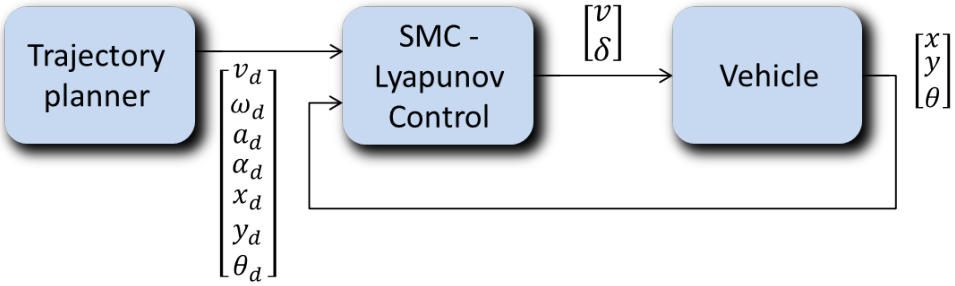


Figure 4.1: Vehicle control structure

4.1 Low level planner for trajectory tracking

When developing an autonomous vehicle system is quite important to take into account that one could choose either a Path Following Task or a Trajectory Tracking Task. It is interesting to differentiate both ideas. On one hand, a path is given by a set of points in a plane, in our case the ground plane. On the other hand, a trajectory is a particular variation of a path. It is given by the same set of points than a path but with an additional component, the time. Each point is represented as $[x_i, y_i, t_i]$ and therefore the complete trajectory can be represented as:

$$\left[(x_0, y_0, t_0), (x_1, y_1, t_1)...., (x_n, y_n, t_n)\right] \tag{4.1}$$

Due to the fact that the path following does not consider the time it will not have longitudinal error. Therefore the vehicle cannot perform a velocity control, which means a constant longitudinal velocity for the vehicle and the reduction of the control complexity being the steering angle the unique control action. In the case of the trajectory tracking, a velocity control is mandatory. As a difference with the path following, the trajectory tracking will focus on reaching a specified point of the path, such a point is the one with the same time t_i as the current time t. Note that if the vehicle accumulate some important error, the controller can present an aggressive behaviour with the objective of minimizing the actual error. Hence, it can be stated that the path following usually results in a smoother behaviour due to the lower error produced although with a constant velocity while the trajectory tracking option give us the benefit of setting a desired speed for specific segments of the trajectory.

After this discussion a trajectory tracking task has been chosen for this work. Among all the planners used in the robotic field, the polynomial category is one of the most promising techniques. With the help of [2, 23], a quintic spline trajectory planning approach has been implemented. The main advantage of this planner with respect to lower degree planners is the continuity of the velocity profile (i.e. a velocity profile with continuous acceleration). It will generate smooth trajectories with low levels of acceleration and jerk, which will ensure the passenger comfort.

Several trials have shown us that for our application, a simpler local trajectory planner could be interesting. Consequently, it was decided to modify the path planner presented in [2, 23]. The modification was to establish a constant velocity once the first segment has finished. Only when the first and the final segments arrives the algorithm performs a smooth velocity profile under bounded acceleration. Hereafter a brief pseudo code of such a modified planner is presented:

Algorithm 1 Trajectory Planner algorithm

input :

x_r : current x location
y_r : current y location
θ_r : current orientation
x_G : desired next global x position
y_G : desired next global y position
θ_G : desired next global orientation
v_i : initial segment velocity
v_f : final segment velocity
a_i : initial segment acceleration
V : desired velocity

output:

x_d : Set of desired x positions
y_d : Set of desired y positions
θ_d : Set of desired orientations
v_d : Set of desired velocities
a_d : Set of desired accelerations
ω_d : Set of desired angular velocities
α_d : Set of desired angular accelerations

if *(firstSegment || lastSegment)* **then**
    $v_f \leftarrow$ V
    $a_i \leftarrow a_f$
    1. x_d,y_d,$\theta_d \leftarrow$ Compute positions and orientations along the segment
    2. Compute the segment length
    3. Compute the segment curvature
    4. Calculate the time for the segment
    5. $v_d \leftarrow$ Compute the segment velocity profile
    6. $\omega_d \leftarrow$ Compute the angular velocity profile
else
    $v_f \leftarrow V$
    $v_i \leftarrow v_f$
    $a_f = 0$
    1. x_d,y_d,$\theta_d \leftarrow$ Compute positions and orientations along the segment
    2. Compute the segment length
    3. Compute the segment curvature
    4. Calculate the time for the segment
    5. $v_d \leftarrow v_i$
    6. $\omega_d \leftarrow$ Compute the angular velocity profile
end

4.2 Direct Lyapunov approach

The control objective of path following consists on reaching asymptotically a null difference between the position and orientation of the real car with respect to the path. As it was explained in last section, a local planner provides the trajectory to be followed. The output data of such a planner will be the setpoints for the controller (v_d and ω_d). The other input is composed by the three errors already defined in (3.42) and as control actions the velocity (v) and the angular velocity of the vehicle's CoG (ω) (see Fig. 4.2).

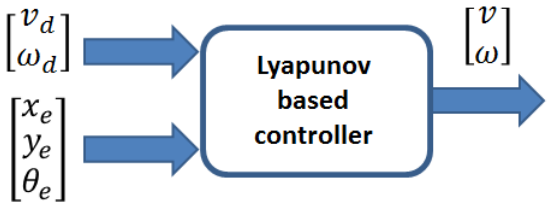


Figure 4.2: Lyapunov based controller diagram

Using (3.42), an error model can be computed for deriving the control laws. After evaluating the time derivative of the error model (3.43) and applying some operations the following dynamic error model is achieved:

$$\begin{bmatrix} \dot{x}_e \\ \dot{y}_e \\ \dot{\theta}_e \end{bmatrix} = \begin{bmatrix} \dot{e}_1 \\ \dot{e}_2 \\ \dot{e}_3 \end{bmatrix} = \begin{bmatrix} \dot{\theta} e_2 + \mu_1 \\ -\dot{\theta} e_1 + v_d \sin(e_3) \\ \mu_2 \end{bmatrix} \tag{4.2}$$

where $\mu = [\mu_1 \mu_2]^T \in \mathbb{R}^2$ is an auxiliary input vector defined as:

$$\begin{bmatrix} \mu_1 \\ \mu_2 \end{bmatrix} = \begin{bmatrix} v_d \cos(e_3) \\ \dot{\theta}_d \end{bmatrix} - \begin{bmatrix} v \\ \dot{\theta} \end{bmatrix} \tag{4.3}$$

where the first term is an auxiliary vector and the second one corresponds to the control actions that we are looking for.

Based on (4.2) and the subsequent closed-loop error system development, the auxiliary input vector can be extracted obtaining the following expression:

$$\begin{bmatrix} \mu_1 \\ \mu_2 \end{bmatrix} = \begin{bmatrix} -k_1 \cdot x_e \\ -k_2 \cdot v_d \cdot \frac{sin\theta_e}{\theta_e} \cdot y_e - k_3 \cdot \theta_e \end{bmatrix} \tag{4.4}$$

where k_1, k_2 and $k_3 \in \mathbb{R}^1$ are constants and positive gains to assure the asymptotic stability of the closed loop. Subtracting the control variables from (4.3), the following control laws are obtained:

$$\begin{bmatrix} v \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} v_d \cos(e_3) \\ \dot{\theta}_d \end{bmatrix} - \begin{bmatrix} \mu_1 \\ \mu_2 \end{bmatrix} = \begin{bmatrix} v_d \cdot cos\theta_e + k_1 \cdot x_e \\ \dot{\theta}_d + k_2 \cdot v_d \cdot \frac{sin\theta_e}{\theta_e} \cdot y_e + k_3 \cdot \theta_e \end{bmatrix} \tag{4.5}$$

Substituting (4.4) into (4.2), the following closed-loop error system is obtained:

$$\begin{bmatrix} \dot{e}_1 \\ \dot{e}_2 \\ \dot{e}_3 \end{bmatrix} = \begin{bmatrix} \dot{\theta} y_e - k_1 x_e \\ -\dot{\theta} x_e + v_d \sin(\theta_e) \\ -k_2 \cdot v_d \cdot \frac{sin\theta_e}{\theta_e} \cdot y_e - k_3 \cdot \theta_e \end{bmatrix} \tag{4.6}$$

Regarding the stability of the system the Lyapunov's theorem for local and global stability is used. To do so, a Lyapunov function candidate is defined:

$$V(e) = \frac{k_2}{2} e_1^2 + \frac{k_2}{2} e_2^2 + \frac{1}{2} e_3^2 \tag{4.7}$$

The derivative of such function is:

$$\dot{V}(e) = k_2 e_1 \dot{e}_1 + k_2 e_2 \dot{e}_2 + e_3 \dot{e}_3 \tag{4.8}$$

After substituting (4.6) into (4.8), the following expression is obtained:

$$\dot{V}(e) = -k_2 k_1 e_1^2 - k_3 e_3^2 \leq 0 \tag{4.9}$$

where k_1, k_2 and k_3 have been already defined in (4.4) as a positive definite constants. Equations (4.7) and (4.8) fulfill the theorem clauses:

- $V(0) = 0$
- $V(e) > 0, \forall e \neq 0$
- $\dot{V}(e) < 0, \forall e \neq 0$
- $V(e) \to \infty \ \ as \|e\| \to \infty$

therefore e = 0 is globally asymptotically stable.

4.3 Sliding Mode Control

The sliding mode control technique is considered as one of the most promising methods for robust control. Such controllers are able to control high order nonlinear systems with structured and unstructured uncertainties. One of the major advantages is its robustness with respect to disturbances and unmodeled dynamics what makes less important to have an exact model of the vehicle. This technique has been applied to a wide range of control problems in robotics and industrial processes and due to this fact the research about this technique is widely used at universities and industry. This method uses a sliding surface or a set of them representing either a line or a curve where the system converges. So the main concept is to reach the surface in a finite time and remain on such surface where the error is null.

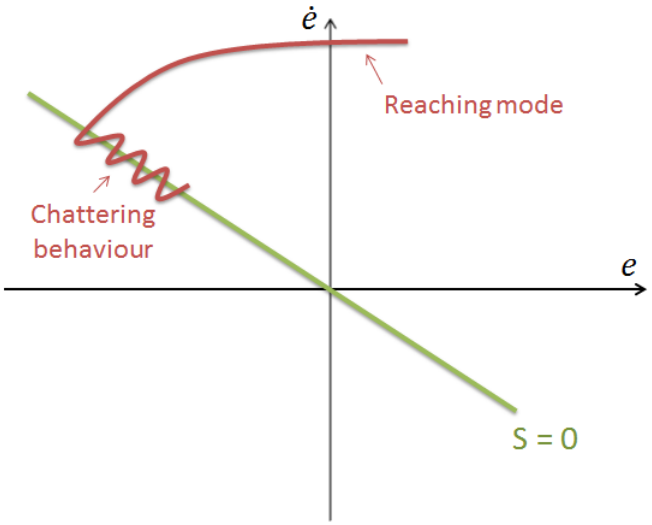


Figure 4.3: Phase portrait under sliding mode behaviour

Figure 4.3 shows the basic principle of the method and it can be seen how the trajectory reach the surface and remain there but with oscillations around it, such a behaviour is called chattering phenomena. This is a drawback and it can be reduced by using some approaches like the use of higher order sliding surface or smoother functions than the sign function as already discussed.

In this work, a stable SMC has been designed that considers as inputs the errors (3.42) and the path planner output(v_d, w_d, a_d and α_d), and as outputs the velocity (v) and the steering angle (δ) (see Fig. 4.4).

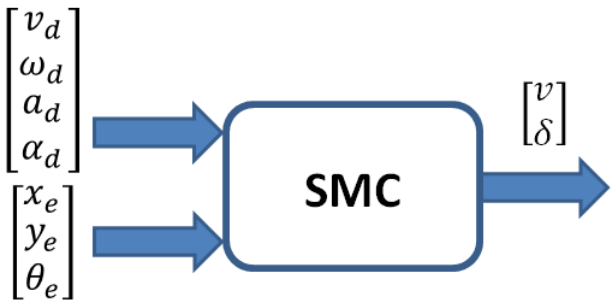


Figure 4.4: Sliding mode controller diagram

The general formula to define a sliding surface according [6] is:

$$s(e,t) = (\frac{d}{dt} + \lambda)^{(n-1)} e \tag{4.10}$$

where n represents the order of the sliding surface, λ is a positive constant and e is the error.

We use a second order sliding surface s_i per control action, hence there will be two sliding surfaces. Due to the fact that there are three error variables (x_e, y_e and θ_e), one sliding surface have to contain two of such error variables. It has been decided to couple y_e and θ_e in the same sliding equation, therefore the resulting surfaces are the following:

$$s_1 = \dot{x}_e + k_1 \cdot x_e \tag{4.11}$$

$$s_2 = \dot{y}_e + k_2 \cdot y_e + k_3 \cdot \theta_e \tag{4.12}$$

where k_1, k_2 and k_3 are positive defined parameters.

According to Gao and Hung [9], the dynamics of the sliding surface follows the reaching law:

$$\dot{s}_i = -Q_i \cdot s_i - P_i \cdot sign(s_i) \tag{4.13}$$

where Q and P are positive defined parameters and its stability can be proven using Lyapunov theorem [6]. A Lyapunov candidate function is defined in order to prove the asymptotic stability:

$$V = \frac{1}{2} \cdot s' \cdot s \tag{4.14}$$

Evaluating its time derivative:

$$\dot{V} = s \cdot \dot{s} \tag{4.15}$$

and considering (4.13), it can be expressed as:

$$\dot{V} = s_1 \cdot (-Q_1 \cdot s_1 - P_1 \cdot sign(s_1)) + s_2 \cdot (-Q_2 \cdot s_2 - P_2 \cdot sign(s_2)) \tag{4.16}$$

or alternatively:

$$\dot{V} = -Q_1 \cdot s_1^2 - Q_2 \cdot s_2^2 - P_1 \cdot |s_1| - P_2 \cdot |s_2| \tag{4.17}$$

To fulfill the Lyapunov stability theorem Q_1, Q_2, P_1 and P_2 have to be semi-positive definite constants.
The convergence of a null error, which means to reach the sliding surfaces, can be proven by doing $s_1 = 0$ and $s_2 = 0$ such that:

$$If s_1 = 0 \implies \dot{x}_e = -k \cdot x_e \implies x_e \to 0 \tag{4.18}$$

$$If s_2 = 0 \implies \dot{y}_e = -k_2 \cdot y_e - k_3 \cdot \theta_e \tag{4.19}$$

By assuming that x_e goes to zero first, $\dot{y}_e = v_r \cdot sin(\theta_e) - x_e \cdot \omega_d$ pass to be $\dot{y}_e = v_r \cdot sin(\theta_e)$ and there exists two cases:

- If $y_e > 0$ and $\theta_e > 0 \implies \dot{y}_e < 0$ it means that y_e goes to $-\theta_e$, on the other hand if $\theta_e < 0$, $\dot{y}_e = v_r \cdot sin(\theta_e)$ and assuming v_r positive it is obtained that $\dot{y}_e < 0$.

- If $y_e < 0$ and $\theta_e < 0$ implies that y_e is increasing until $y_e = -\theta_e$, for $\theta_e < 0 \implies \dot{y}_e = v_r \cdot sin(\theta_e)$ and therefore $\dot{y}_e < 0$, thus y_e and θ_e will converge to zero.

After proving the asymptotic stability the control law will be determined. Both control actions will have the following shape:

$$u_i = u_{eq_i} - u_{c_i} \tag{4.20}$$

The first term is called equivalent control and forces the derivative of the sliding surface to be equal to zero to stay on the sliding surface.
The second part of (4.20) is the corrective control which compensates the deviations from the sliding surface to reach the sliding surface.

$$u_{c_i} = \frac{Q_i \cdot s_i + P_i \cdot sign(s_i)}{g(x)} \tag{4.21}$$

Hence, in order to obtain the control law it is necessary to find the term u_{eq_i}. To do so, the sliding surfaces

are derived and set equal to zero.

$$\dot{s_1} = \ddot{x_e} + k_1 \cdot \dot{x_e} \tag{4.22}$$

$$\dot{s_2} = \ddot{y_e} + k_2 \cdot \dot{y_e} + k_3 \cdot \dot{\theta_e} \tag{4.23}$$

By developing the two last equations:

$$\dot{s_1} = \dot{v} \cdot cos(\theta_e) + v \cdot \dot{\theta_e} \cdot sin(\theta_e) + \dot{y_e} \cdot \omega_d + y_e \cdot \dot{\omega}_d - \dot{v}_d + k_1 \cdot \dot{x_e} \tag{4.24}$$

$$\dot{s_2} = \dot{v}sin(\theta_e) + vcos(\theta_e)\dot{\theta_e} - x_e\dot{\omega}_d - \dot{x_e}\omega_d + K_2\dot{y_e} + K_3\dot{\theta_e} \tag{4.25}$$

$\dot{s_1} = 0$:

$$u_{eq_1} = \dot{v} = \frac{-v\dot{\theta_e}sin(\theta_e) - \dot{y_e}\omega_d - y_e\dot{\omega}_d + \dot{v}_d - k_1\dot{x_e}}{cos(\theta_e)} \tag{4.26}$$

$\dot{s_2} = 0$:

$$u_{eq_2} = \omega = \omega_d + \frac{-k_2\dot{y_e} + \dot{w}_d x_e + w_d\dot{x_e} - \dot{v}sin(\theta_e)}{vsin(\theta_e) + k_3} \tag{4.27}$$

where the denominator corresponds with $g(x)$ in (4.21).

Therefore, replacing the obtained equivalent control equations in the structure proposed in (4.21) and developing them, the following control laws are obtained:

$$\dot{v_c} = \frac{-v\dot{\theta_e}sin(\theta_e) - \dot{y_e}\omega_d - y_e\dot{\omega}_d + \dot{v}_d - k_1\dot{x_e} - Q_1 s_1 - P_1 sign(s_1)}{cos(\theta_e)} \tag{4.28}$$

$$v_c = \int \dot{v_c} dt \tag{4.29}$$

$$\dot{\theta_c} = \omega_d + \frac{-P_2 s_2 - Q_2 sign(s_2) - k_2\dot{y_e} + \dot{\omega}_d x_e + \omega_d\dot{x_e} - \dot{v}sin(\theta_e)}{vsin(\theta_e) + k_3} \tag{4.30}$$

$$\delta = tan^{-1}(\frac{l_f}{v} \cdot \dot{\theta_c}) \tag{4.31}$$

Equations (4.28) and (4.29) represent the velocity control law and (4.30) and (4.31) represent the steering control law. The transformation from angular CoG velocity to steering angle saw in (4.31) is obtained from the kinematic model (3.40).

Note that in the kinematic model $\dot{v}$ corresponds with $\dot{v_c}$, but if the control law is tested over a dynamic model $\dot{v} \neq \dot{v_c}$ since there exist dynamics between the throttle and the CoG acceleration. Note also that the control laws are discontinuous when crossing the sliding surface $s = 0$, which may cause the chattering

problem. This problem can be eliminated by replacing the signum function $(sgn(s))$ by a saturation function:

$$sat(s) = \begin{cases} s, & |s| \leq 1 \\ sign(s), & |s| > 1 \end{cases} \tag{4.32}$$

Chapter 5

Simulation Results

In this section, the results of the previous control methods are presented in simulation and the performance of the control strategies are compared and discussed by means of two test scenarios. The simulation has been developed in Matlab/Simulink and Unity 5.1.3 platforms. In parallel with the implementation of the controller, the trajectory planner has also been implemented which provides the specific instructions to the control area. The next steps are followed to perform the trajectory tracking:

- A set of forward way points are set to generate a route in a offline way. The space between two way points is called segment.
- When the vehicle arrives to a way point, the planner takes the next way point and perform the correct speed profile between them. From this segment, a set of sub way points are calculated with its respective position, orientation, linear velocity, angular velocity, linear acceleration and angular acceleration.
- Once such a segment has been sampled, at every sample time (100 ms) the control area takes a sub way point features as a desired configuration and perform the control.

5.1 Results in Matlab/Simulink

In order to prove in a correct way the performance of the control techniques, a set of tests has been done. Three circuits (A, B and C) has been used (see Figure 5.2). The first test has consisted in assessing the methods at different velocities for circuits A, B and C at 20, 15 and 10 $\frac{km}{h}$ respectively. The second test

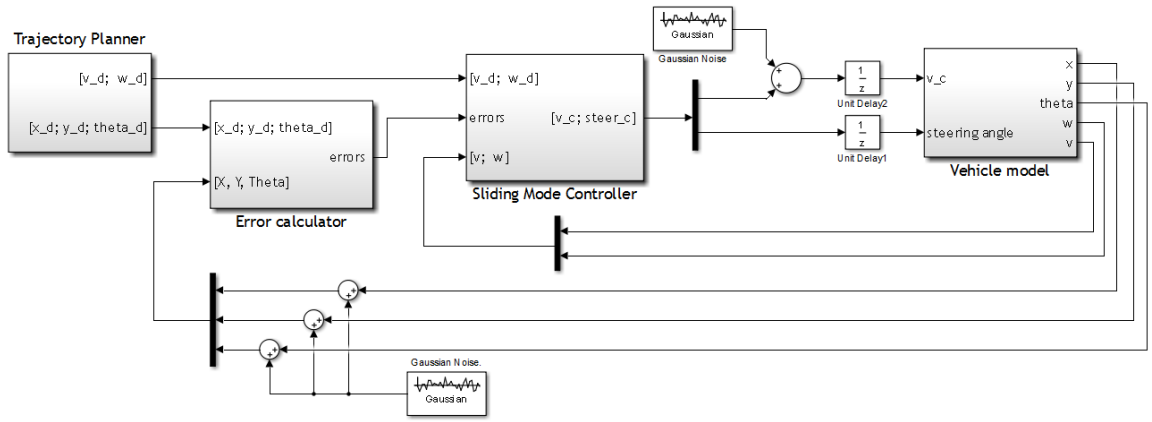


Figure 5.1: Matlab/Simulink diagram used for control techniques proofs

has consisted in proving them at different velocities but in the same circuit and with the same controller adjustment in order to check the robustness.

The used Matlab/Simulink diagram is shown in Figure 5.1. For simulating it has been taken into account Gaussian sensor noise. Inside the kinematic vehicle model, two first order filters have been considered in order to mitigate the high frequency terms of the control signals. It has also been taken into account a sample time delay inside the loop.

5.1.1 Circuits test

Circuit A

Both control algorithms have been tested in circuit A using for that the parameters in Table 5.1 and the results are presented in Figure 5.3.

Table 5.1: Circuit A control parameters

SMC		Lyapunov	
Name	Value	Name	Value
k_1	10	k_1	10
k_2	20	k_2	1
k_3	25.5	k_3	13
P_1	0.5		
Q_1	0.05		
P_2	3.7		
Q_2	0.3		

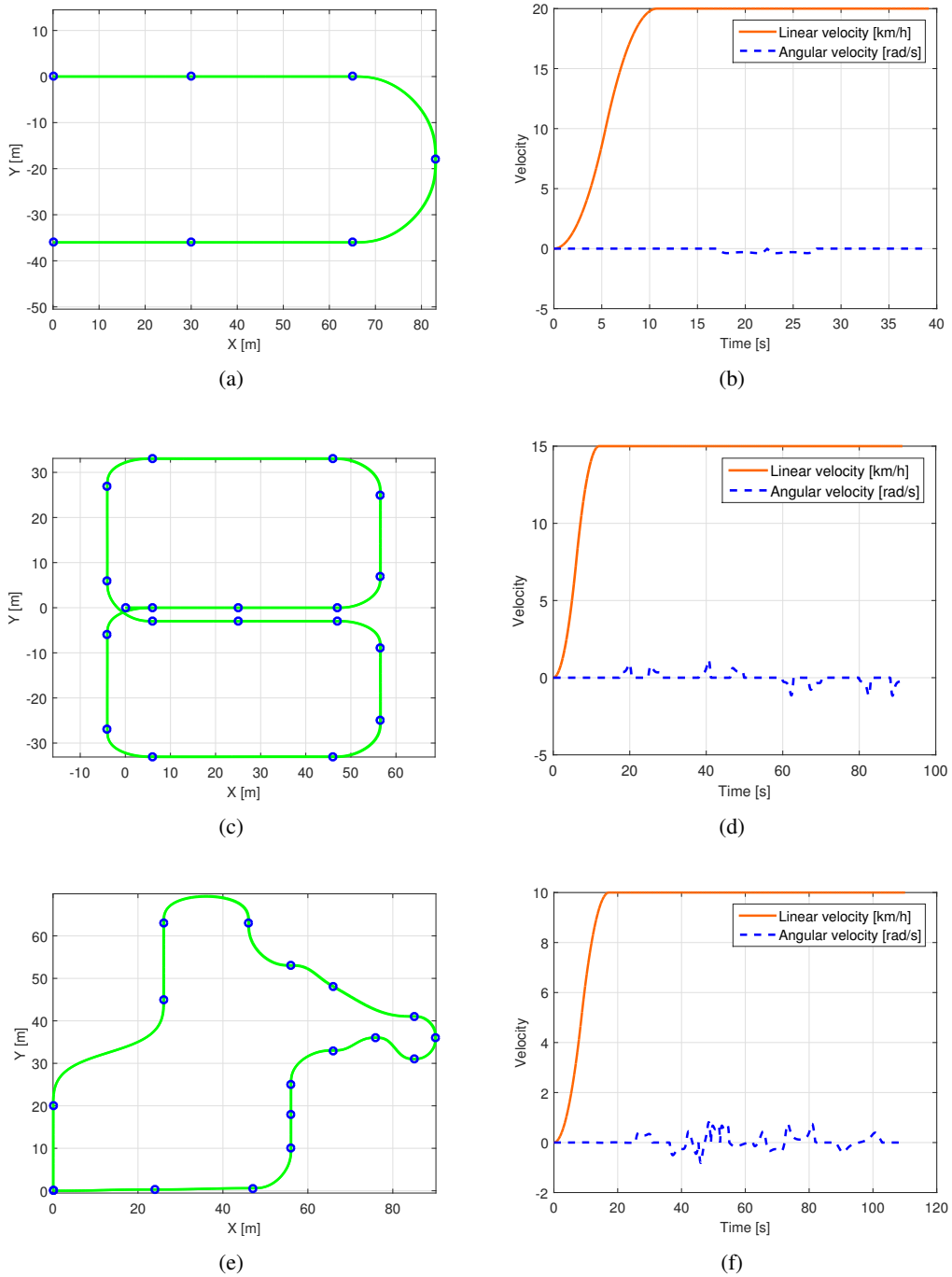
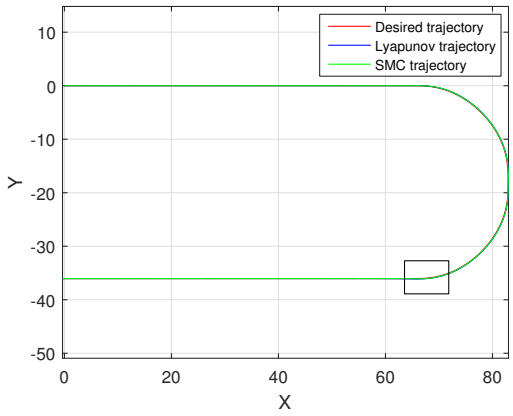
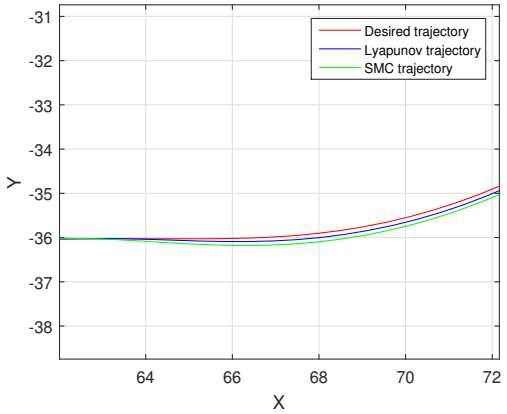


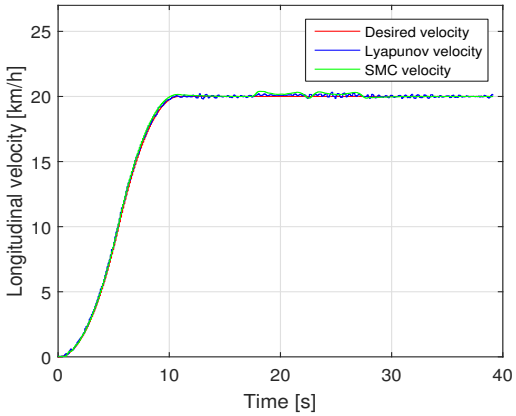
Figure 5.2: Circuit trajectories and its correspondent velocity profiles: (a) Circuit A trajectory where blue circles represent the global way points given by the global planner (b) Velocity profiles of circuit A (c) Circuit B trajectory (d) Velocity profiles of circuit B (e) Circuit C trajectory (f) Velocity profiles of circuit C


Desired trajectory
Lyapunov trajectory
SMC trajectory
Y
X

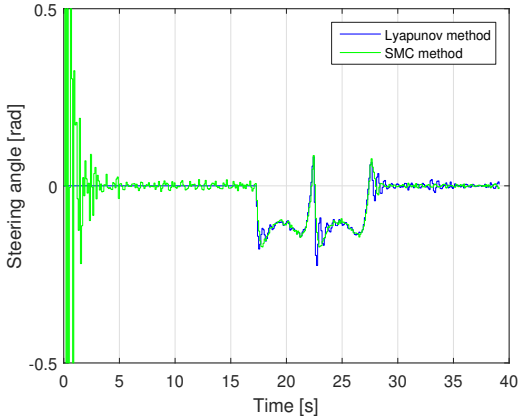
(a)


Desired trajectory
Lyapunov trajectory
SMC trajectory
Y
X

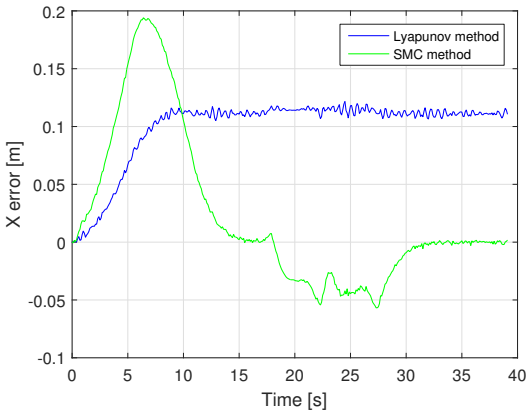
(b)


Desired velocity
Lyapunov velocity
SMC velocity
Longitudinal velocity [km/h]
Time [s]

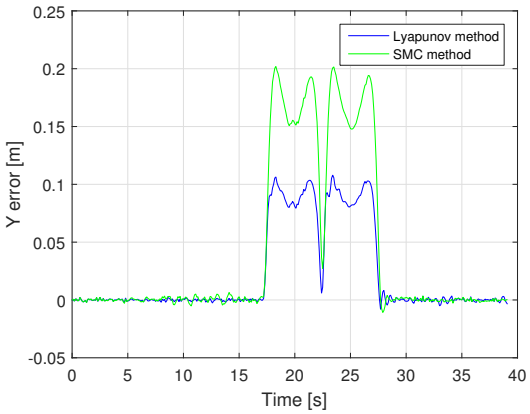
(c)


Lyapunov method
SMC method
Steering angle [rad]
Time [s]

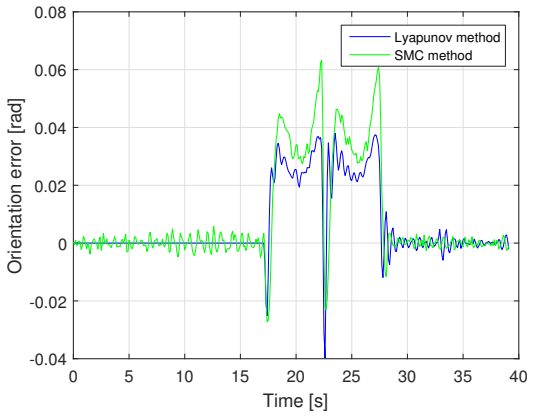
(d)


Lyapunov method
SMC method
X error [m]
Time [s]

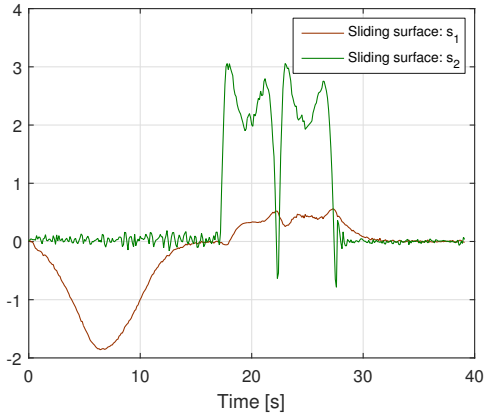
(e)


Lyapunov method
SMC method
Y error [m]
Time [s]

(f)



(g)



(h)

Figure 5.3: Circuit A results for both control techniques: (a) Desired path and real trajectory followed (b) Zoom of the rectangle in first Figure (c) Desired and real longitudinal velocity profiles (d) Steering angle control action (e) Longitudinal vehicle axis error (f) Lateral vehicle axis error (g) Orientation error (h) Sliding surfaces (in SMC algorithm)

Circuit B

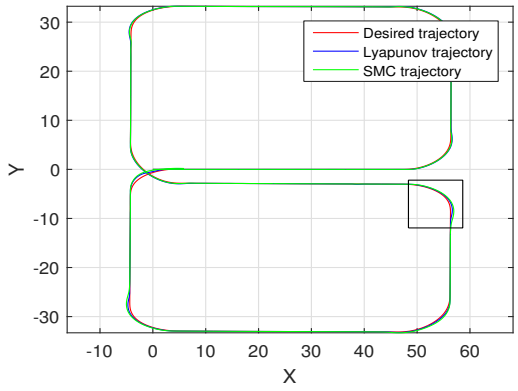
In this circuit, the maximum velocity is reduced at 15 $\frac{km}{h}$ and its geometry is a little bit more complicated. The control parameters chosen for such a test are shown in Table 5.2. The results are presented in Figure 5.4.

Table 5.2: Circuit B control parameters

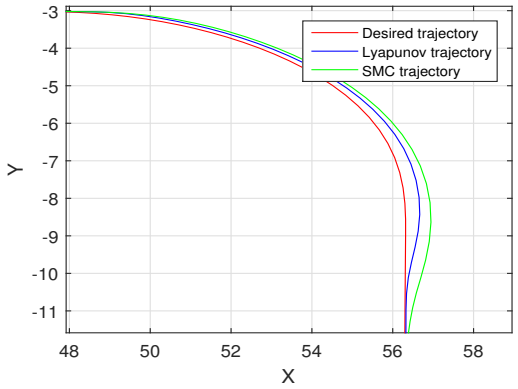
SMC		Lyapunov	
Name	Value	Name	Value
k_1	6	k_1	10
k_2	6	k_2	1
k_3	7.65	k_3	10
P_1	1		
Q_1	0.3		
P_2	4		
Q_2	0.3		

Circuit C

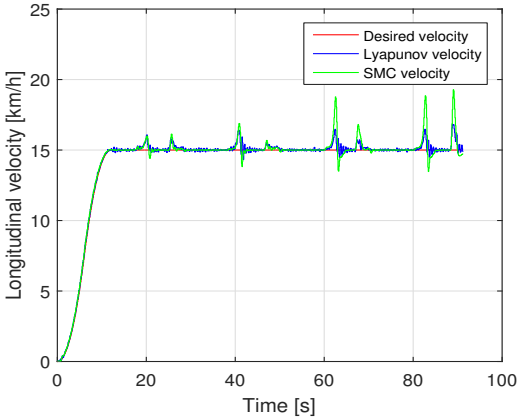
In this circuit, the maximum velocity has been reduced at 10 $\frac{km}{h}$. Table 5.3 shows the employed parameters. The geometry of such a circuit is the most complex. The results are shown in Figure 5.5.


Desired trajectory
Lyapunov trajectory
SMC trajectory
X
Y

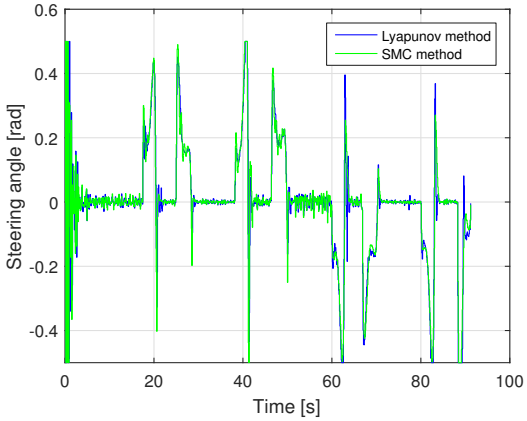
(a)


Desired trajectory
Lyapunov trajectory
SMC trajectory
X
Y

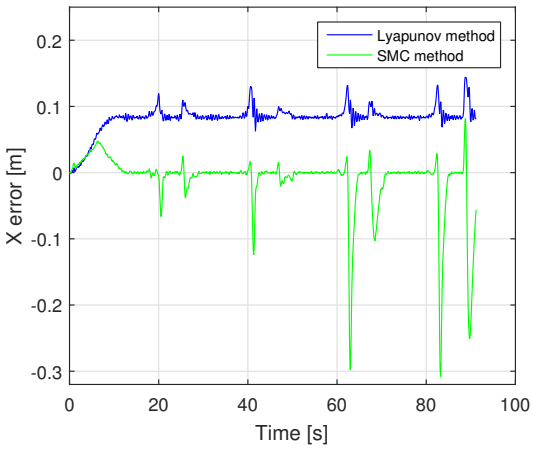
(b)


Desired velocity
Lyapunov velocity
SMC velocity
Longitudinal velocity [km/h]
Time [s]

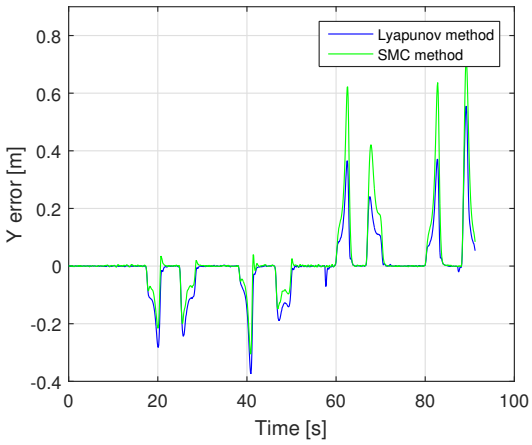
(c)


Lyapunov method
SMC method
Steering angle [rad]
Time [s]

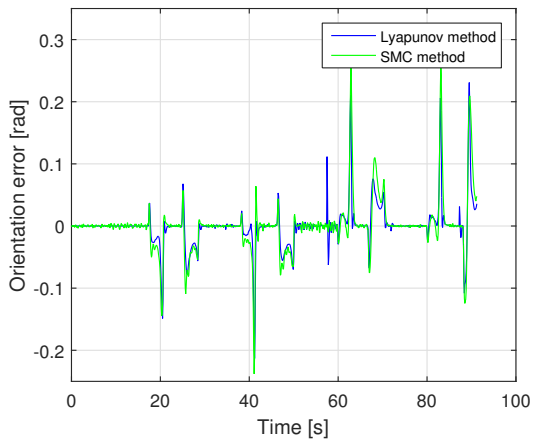
(d)


Lyapunov method
SMC method
X error [m]
Time [s]

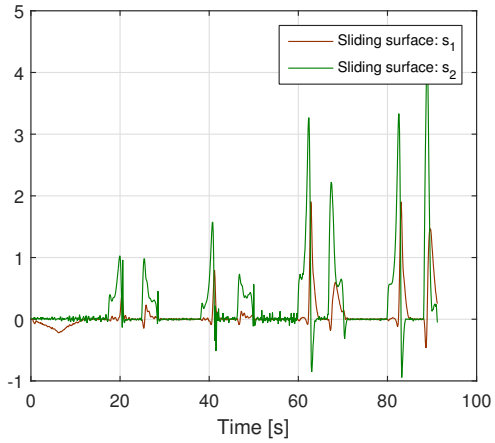
(e)


Lyapunov method
SMC method
Y error [m]
Time [s]

(f)



(g)



(h)

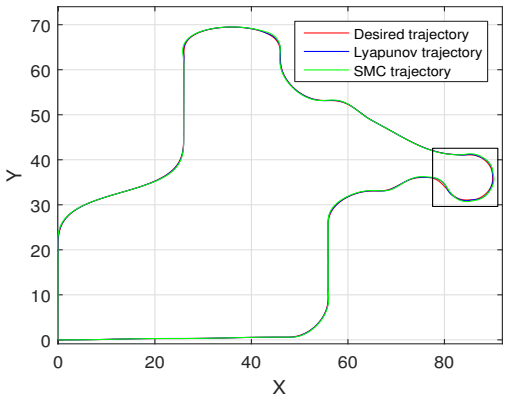
Figure 5.4: Circuit B results for both control techniques: (a) Desired path and real trajectory followed (b) Zoom of the rectangle in first Figure (c) Desired and real longitudinal velocity profiles (d) Steering angle control action (e) Longitudinal vehicle axis error (f) Lateral vehicle axis error (g) Orientation error (h) Sliding surfaces (in SMC algorithm)

Table 5.3: Circuit C control parameters

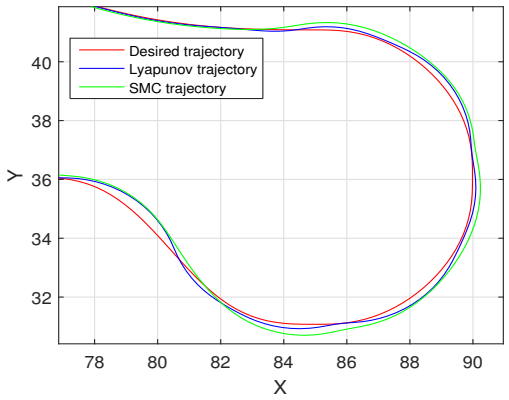
SMC		Lyapunov	
Name	Value	Name	Value
k_1	5	k_1	9
k_2	13	k_2	3
k_3	16.5	k_3	8
P_1	0.6		
Q_1	0.05		
P_2	4		
Q_2	0.1		

Conclusions of the test

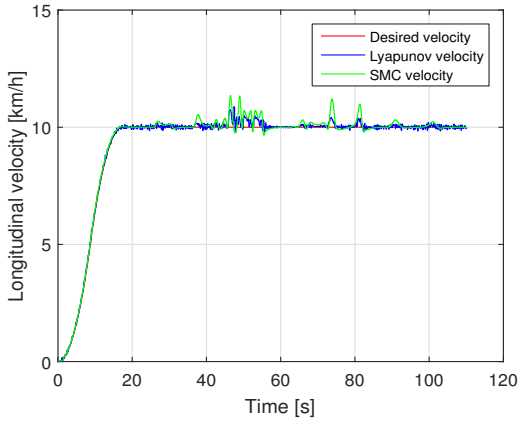
Lyapunov technique seems to have a better performance in terms of minimizing the error, but according to Figures 5.3.(e), 5.4.(e) and 5.5.(e), it can be appreciated how the Lyapunov controller has a steady state error. Due to we are testing the control laws over the kinematic model, the velocity control action is directly the current velocity of the vehicle. Therefore, both SMC control actions are coupled since the steering angle control action depends on the current vehicle velocity that corresponds with the last applied velocity action. However, Lyapunov control actions do not depend one of each other. The result of this coupling can be seen in Figures (c) and (d), where it can be appreciated how the velocity action has a peak in curves and how the steering angle is noisier in SMC technique.


Desired trajectory
Lyapunov trajectory
SMC trajectory
Y
X

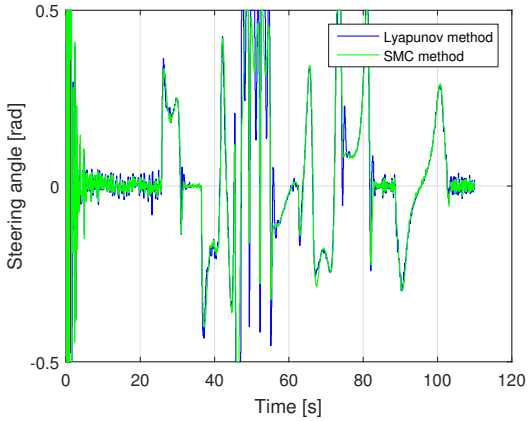
(a)


Desired trajectory
Lyapunov trajectory
SMC trajectory
Y
X

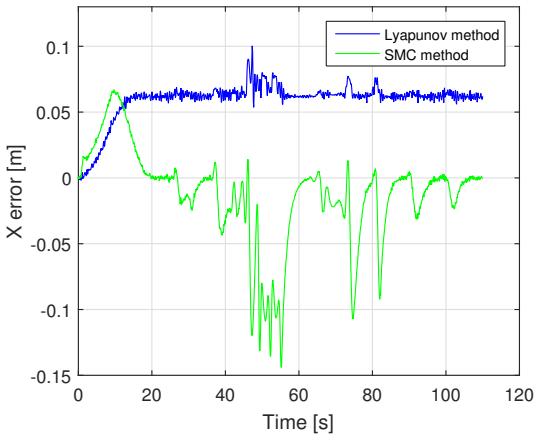
(b)


Desired velocity
Lyapunov velocity
SMC velocity
Longitudinal velocity [km/h]
Time [s]

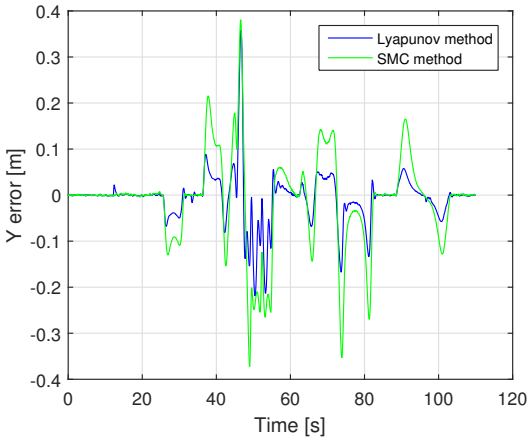
(c)


Lyapunov method
SMC method
Steering angle [rad]
Time [s]

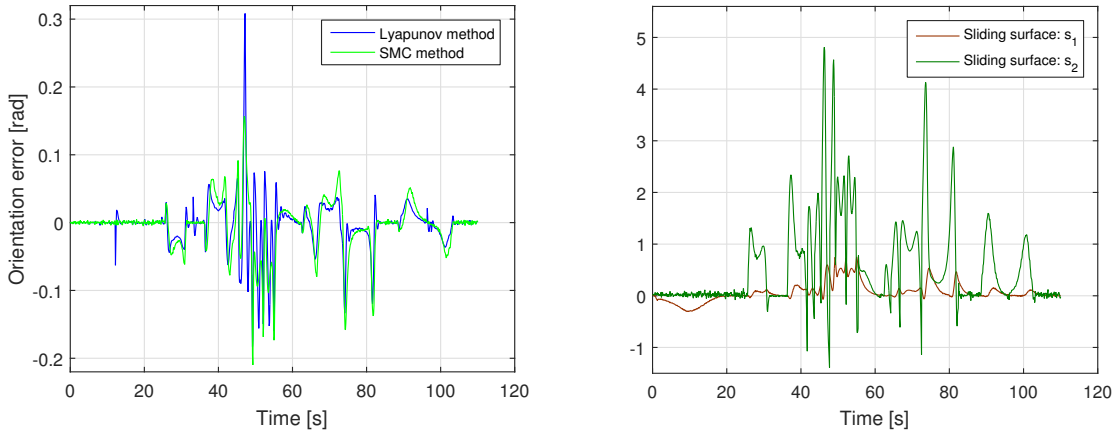
(d)


Lyapunov method
SMC method
X error [m]
Time [s]

(e)


Lyapunov method
SMC method
Y error [m]
Time [s]

(f)



(g) (h)

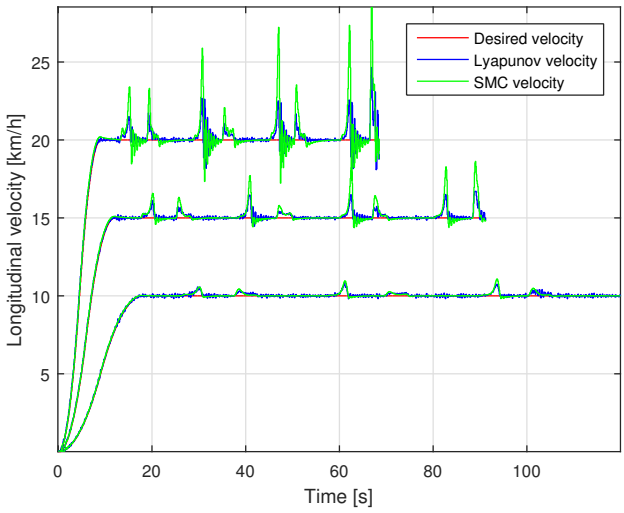
Figure 5.5: Circuit C results for both control techniques: (a) Desired path and real trajectory followed (b) Zoom of the rectangle in first Figure (c) Desired and real longitudinal velocity profiles (d) Steering angle control action (e) Longitudinal vehicle axis error (f) Lateral vehicle axis error (g) Orientation error (h) Sliding surfaces (in SMC algorithm)

Also, in Figures 5.3.(d), 5.4.(d) and 5.5.(d), the noisy behaviour of SMC steering angle at the begining of every test is due to the differences between the numerator and the denominator of (4.31) (i.e. when the vehicle starts to accelerate, the steering angle control should provide a null angular velocity since the vehicle goes straight ahead. But due to noisy measurements, (4.30) achieves a small angular velocity, and due to the fact that the longitudinal velocity of the vehicle is near to zero too, (4.31) computes a high steering angle action). This behaviour can be eliminated by forcing the steering angle action to be null during the first second.

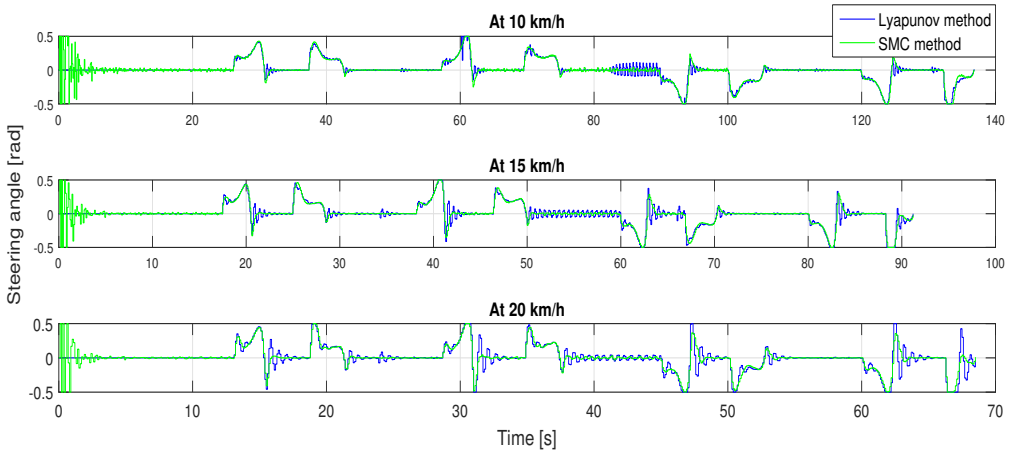
A drawback of SMC method is that it depends on seven constant parameters converting it in a control technique difficult of adjusting in comparison with Lyapunov technique that only depends on three.

5.1.2 Robustness test

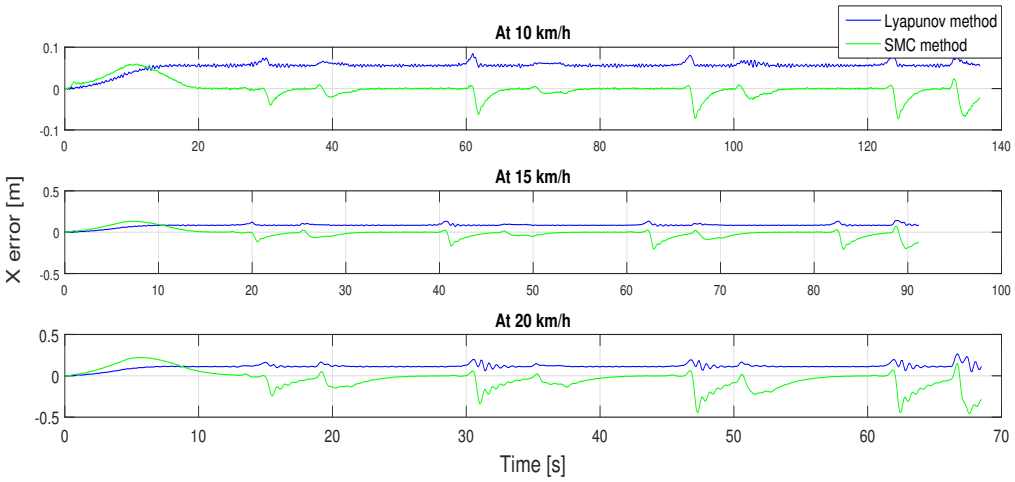
In this subsection, a test in order to find out the limits of robustness of both control techniques has been done. The experiment has consisted in using a set of velocity scenarios over the same circuit (circuit B) and with the same initial control parameters. Such a process will help us to understand in more detail the behaviour of the control algorithms. The results are presented in Figure 5.6.


Desired velocity
Lyapunov velocity
SMC velocity
Longitudinal velocity [km/h]
Time [s]

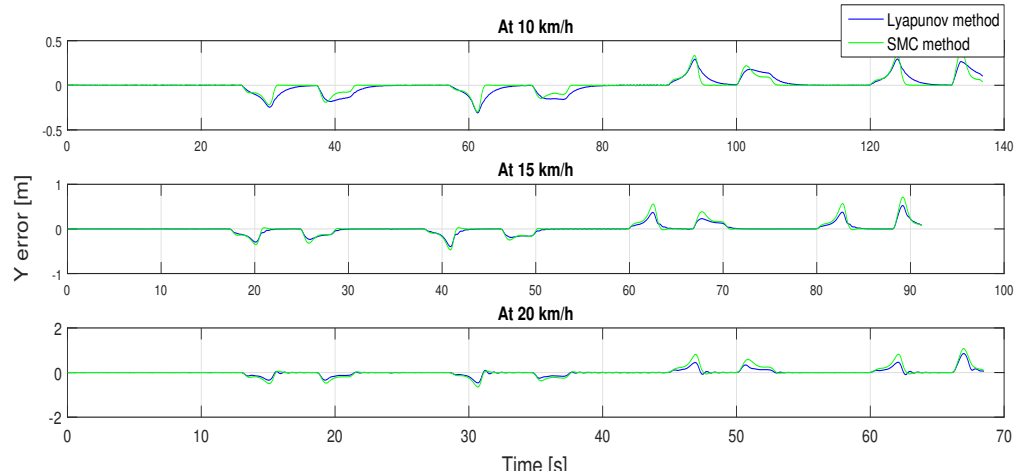
(a)


Lyapunov method
SMC method
At 10 km/h
At 15 km/h
At 20 km/h
Steering angle [rad]
Time [s]

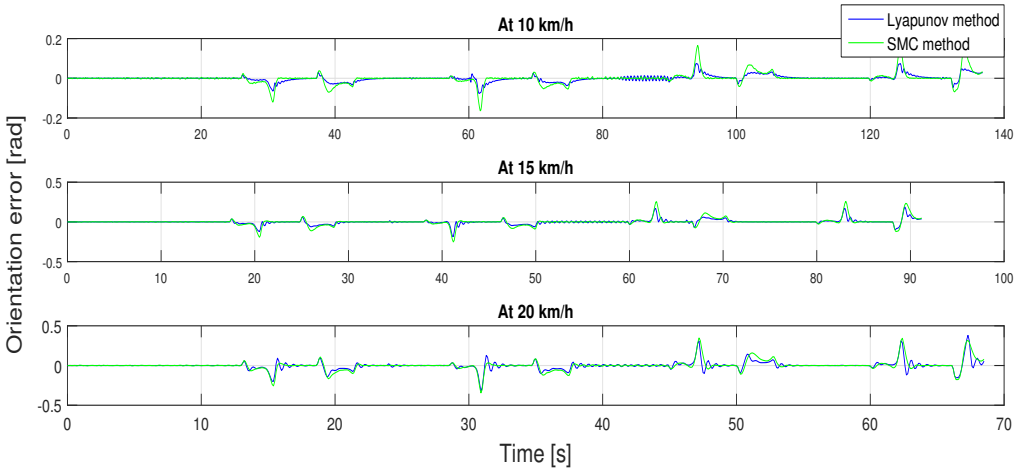
(b)


Lyapunov method
SMC method
At 10 km/h
At 15 km/h
At 20 km/h
X error [m]
Time [s]

(c)



(d)



(e)

Figure 5.6: Results of testing the control algorithms at different velocities under the same scenario and control parameters: (a) Desired and real longitudinal velocity profiles (b) Steering angle control action (c) Longitudinal vehicle axis error (s) Lateral vehicle axis error (e) Orientation error

Table 5.4: Robustness test control parameters

SMC		Lyapunov	
Name	Value	Name	Value
k_1	10	k_1	10
k_2	20	k_2	1
k_3	25.5	k_3	13
P_1	0.5		
Q_1	0.05		
P_2	3.7		
Q_2	0.3		

Conclusions of the test

In Figure 5.6.a, it can be appreciated how the SMC method computes stronger velocity control actions than Lyapunov technique under situations of higher velocity. Accordingly, it can be seen how the Lyapunov control algorithm achieves less longitudinal error than SMC algorithm in curves. In straight segments, the SMC method reaches a null longitudinal error while Lyapunov method presents a steady state error (see Figure 5.6.c).

With respect to the steering angle control action, the SMC technique computes the best actions, in spite of the first five seconds behaviour which can be eliminated as we discussed in the circuits test conclusions. Lyapunov steering angle control action performs an oscillation when it tries to stabilise to zero degrees.

Regarding lateral and orientation error, the SMC technique has a faster mitigation of the error but due to this it performs higher errors.

Lyapunov technique seems to have a better performance in terms of minimizing the error, but in Figures 5.3.(e), 5.4.(e) and 5.5.(e), it can be appreciated how the Lyapunov controller has a steady state error.

As already discussed, in Lyapunov method the control laws are decoupled. The lateral error produced in a curve does not affect to the velocity control. However, in SMC method it affects producing a longitudinal position and velocity error (see Figures (c) and (a)).

Sliding mode technique depends on more time dependent variables and more parameters of adjustment than Lyapunov technique. If the parameters have not been adjusted in an optimal way, the SMC controller will compute a set of actions worse than Lyapunov control actions. Therefore, we can assert that it is difficult to do a comparison when the adjustment of both control laws is different in terms of accuracy.

5.2 Results in Unity

In this section, it is presented a simulation test over Unity framework. Such a Unity platform is very useful due to it has implemented a more complex world than in Matlab. The model of the vehicle is a complete dynamic model which takes into account the suspension dynamics, the drivetrain dynamics and even the engine dynamics among others. The developed circuit, for testing the control techniques, corresponds to the already presented circuit B. In Figure 5.7.a such a circuit can be seen. Figure 5.7.b represents a screen shoot while simulating the test.

(a) (b)

Figure 5.7: Unity 5.1.3 scenario: (a) Urban circuit for developing the tests (b) Screen shoot of the simulation. In the left side appears the vehicle from an external view and in the right side from the driver point of view

The results of applying both control techniques are shown in Figure 5.8. The trajectories are represented in two graphs because of the desired trajectory is not the same. It occurs that since the vehicle plans each segment from its location at the beginning of such a segment such that the car plans a path a little bit different depending on the control law. For this test, the mitigation of the nervous behaviour of the steering angle, presented in Matlab results when the vehicle started the tests, is done (see Figure 5.8.d). Figure 5.8.c shows the better performance of the SMC speed control action. In Figure 5.8.d, it can be appreciated the nervous behaviour of SMC. However SMC method presents less overshoot behaviour than Lyapunov technique in the steering control action. Finally, Figures 5.8.e and 5.8.f show us that the sliding mode control obtains lower longitudinal and lateral errors which indicates that it computes control actions more appropriated.

The parameters used to adapt the control laws are showed in table 5.5. At this point both techniques have

Table 5.5: Unity circuit control parameters

SMC		Lyapunov	
Name	Value	Name	Value
k_1	0.15	k_1	1.5
k_2	5	k_2	1.6
k_3	7	k_3	0.7
P_1	0.3		
Q_1	0.05		
P_2	0.75		
Q_2	0.07		

been tested over a kinematic model in Matlab and over a Dynamic model in Unity. Due to circuit B in Matlab section and the circuit used in Unity are more or less the same we can try to compare the results of performance. Observing the results (see Figures 5.8, 5.4 and 5.6) we can conclude that the SMC technique obtains better results (in comparison with the Lyapunov approach) when it is evaluated over the dynamic model. It is a signal of robustness against unmodelled dynamics, but, as it was discussed in last sections, the behaviour depends a lot of the adjustment.

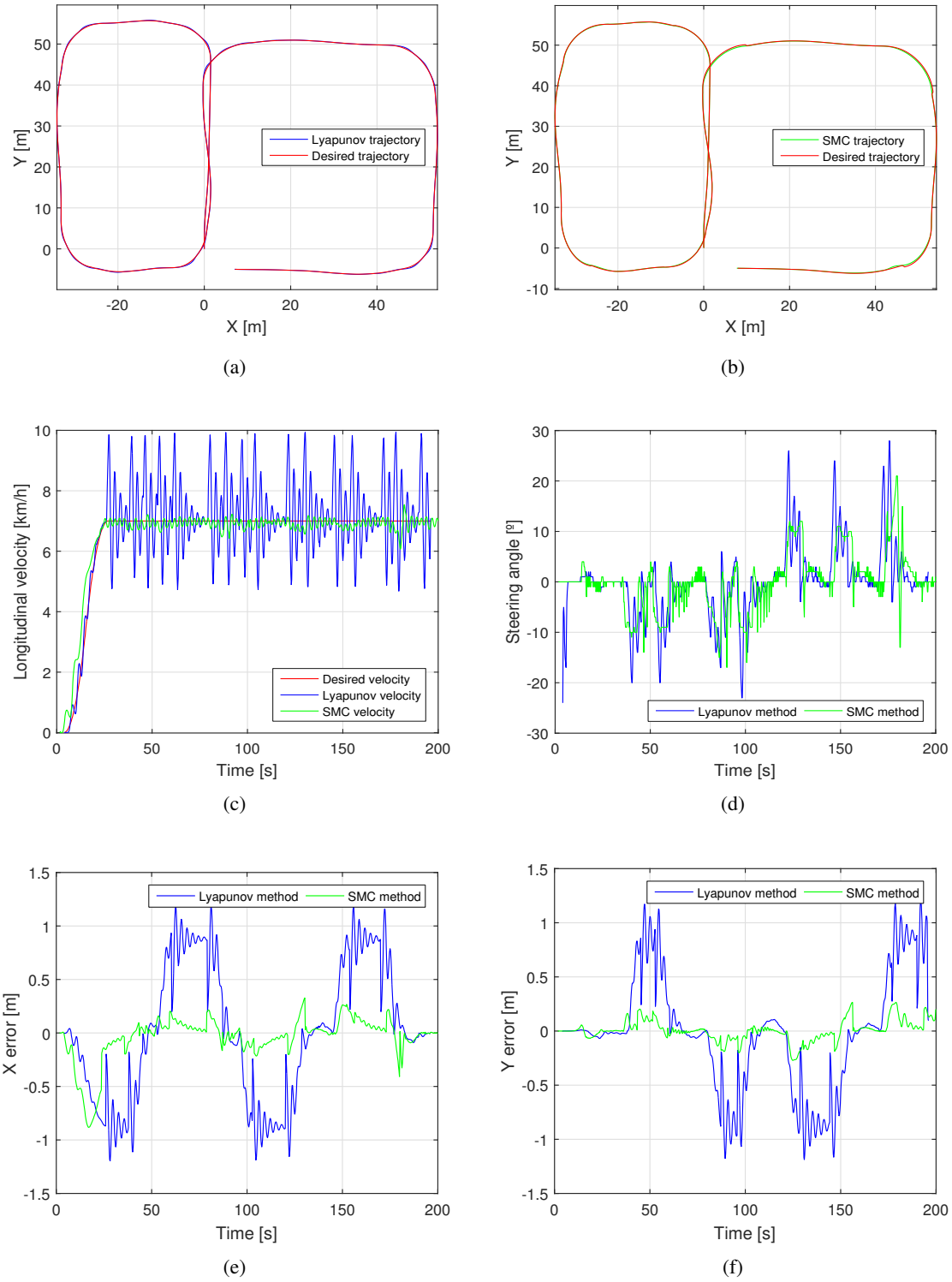


Figure 5.8: Results in Unity for both control techniques: (a) Desired path and real trajectory followed with the Lyapunov based technique (b) Desired path and real trajectory followed with the SMC technique (c) Desired and real longitudinal velocity profiles (d) Steering angle control actions (e) Longitudinal vehicle axis errors (f) Lateral vehicle axis errors

Chapter 6

Experimental Results

This chapter presents the results of the application of our control planning system to a real autonomous vehicle. Due to the lack of time, only the control based on the Lyapunov approach has been tested. Before describing the test scenarios and results, a brief explanation regarding the real implementation is done.

The autonomous vehicle is an electric car instrumented with a set of vehicle sensors (GPS, IMU, cameras, encoders, etc), a set of vehicle actuators (steering angle motor, electric vehicle motor and braking pressure pump), a low level microcontroller and a master computer. The microcontroller is the interface between the vehicle sensors and actuators and the master computer. This microcontroller is in charge of the very low level control of the actuators and communicates with the master computer via CAN-Bus protocol. The cameras take images which are processed by computer vision algorithms to obtain the free space, the objects location around the vehicle as well as to provide an accurate localization of the vehicle in the world by comparing images against a data-set of geolocalized images. The GPS and IMU provide information of the current vehicle localization which is matched with the one computed by the vision localization in order to get a robust vehicle location. The wheel encoders are used for computing the current linear velocity. With all this information, a free space area is checked before starting to plan a new route. An obstacle avoidance module sends a signal to the trajectory planner when an obstacle has been detected. In that case, the planner computes a new trajectory avoiding the collision with the obstacle.

Once the trajectory planner has calculated a new trajectory, it is sent to the non-linear control module to perform the desired behaviour. If the vehicle arrives either to a new global way point or detects an

obstacle, the planner generates a new trajectory for the controller. Figure 6.1 presents such a architecture in a graphic way.

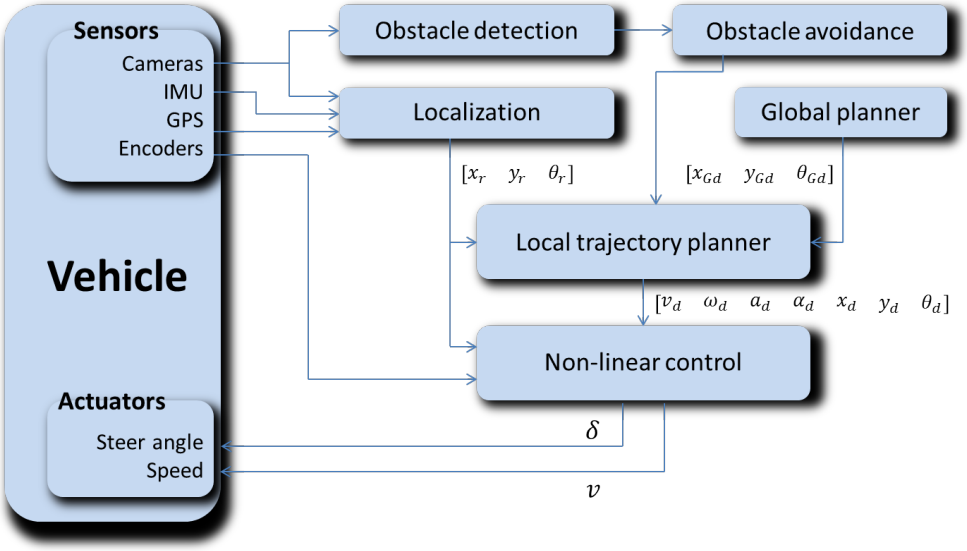


Figure 6.1: Architecture diagram of the real vehicle where main variables are showed

The scenario where the test has been performed is a simple circuit very similar to the one presented as circuit C in Chapter 5 (see Figure 5.2.a).

The test consisted in starting from an initial position (see Figure 6.2), reach a constant velocity ($11\frac{km}{h}$) while following the desired path and at the end the vehicle detected a pedestrian braking in front of him. The controller has been adjusted with the parameters showed in Table 6.1. Such a set of constant values have been found out by trial and error. Therefore, it may exist a set of them which could achieves a better response. While adjusting the parameters, we saw that the relevant control action for a good behaviour

Table 6.1: Control parameters used in the real test

Lyapunov	
Name	Value
k_1	1.5
k_2	1.6
k_3	0.7

was the steering angle so we tried to adjust in the best possible way the parameters k_2 and k_3. The maximum accuracy of the steering angle control action is 2 degrees. It is a hardware limitation imposed by the vehicle developer hence the control will never be able to mitigate all lateral error. Such a limitation produces a steering angle action a bit nervous while trying to achieve the null error. Another constraint

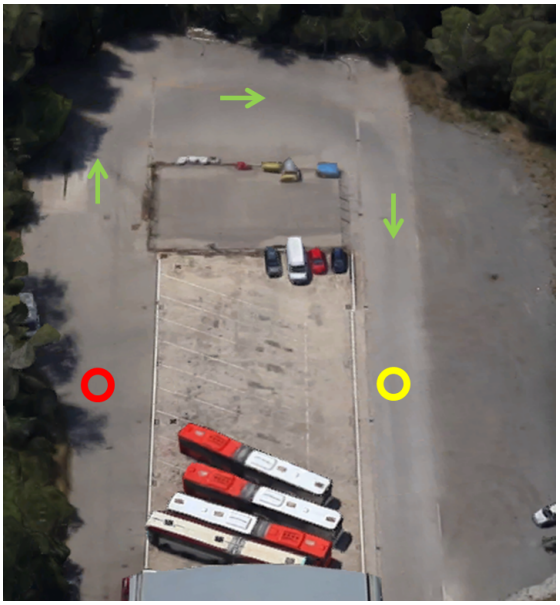


Figure 6.2: Space used to perform tests. Red circle represents the initial position, green arrows are the global way points that define the trajectory and the yellow cilcle represents the location of a pedestrian.

to take into account is that the velocity is an integer, it means that the vehicle rounds the signal sent by the controller to the nearest integer. This produces an error in the longitudinal speed control which could be dealt as a disturbance. We could try to compute control actions taken into account such a disturbance for future work. With the presented configuration and limitations the test has been performed obtaining the results showed in Figure 6.3.

In Figure 6.3, real control actions (red graphs) and real responses (blue graphs) are shown. Note that the vehicle cannot maintain the desired velocity of $11\frac{km}{h}$ since the set of parameters is not the most suitable set. Therefore, it produces an error and the speed control produces strong actions that can be seen in Figure 6.3.c. Regarding the steering angle control, it computes more or less the expected action. We have observed that Lyapunov controller corrects the orientation and lateral error in a better way than the longitudinal error. It is because the speed control law is a proportional correction. For future work, it would be interesting to add an integral part to the speed control law. Note in Figure 6.3.b that the maximum accuracy of two degrees of the steering angle response and in Figure 6.3.a the integer accuracy of the linear speed response already commented above.

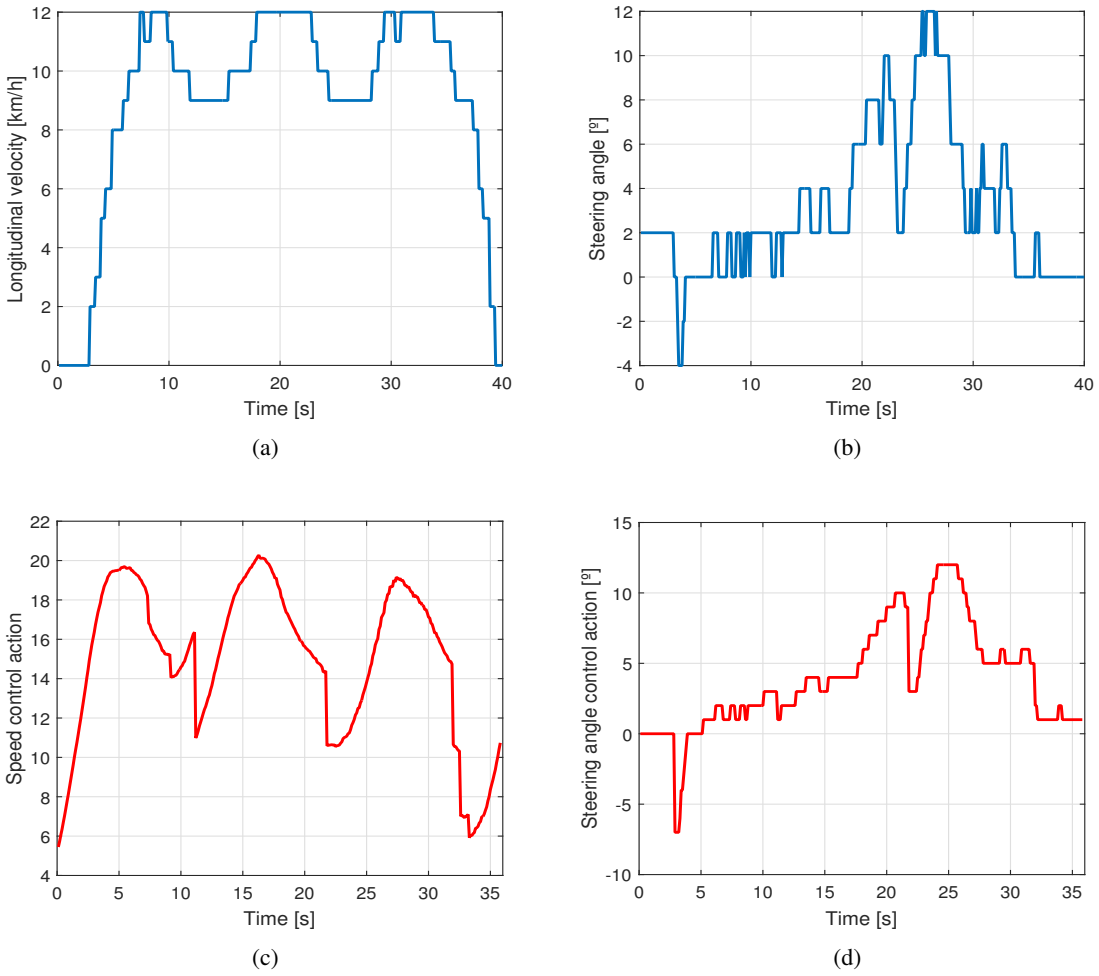
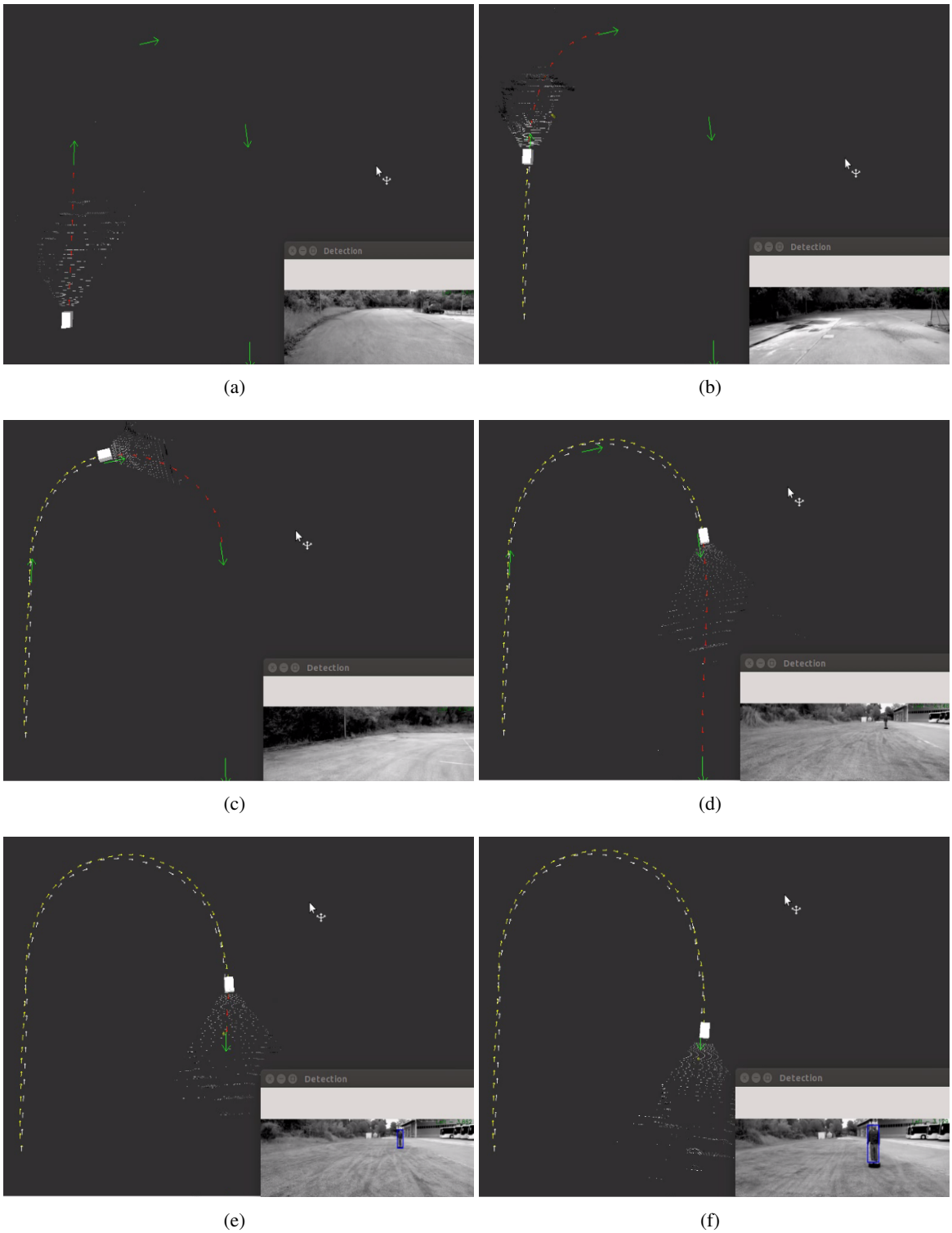


Figure 6.3: Results in real scenario: (a) Real speed of the vehicle during the test (b) Real steering angle produced after applying the steer angle control action (c) Longitudinal velocity sent to the low level controller of the vehicle which computes the real control action for the electric motor (d) Steering angle control action sent to the steer angle actuator



(a) (b)

(c) (d)

(e) (f)

Figure 6.4: Real test in rviz program (ROS) where they are displayed in real time the cameras visualisation, the global way points (green arrows), the desired trajectory (red lines), the completed path (white lines), the real trajectory (yellow lines) and the disparity (cloud of 3D points): Images (a),(b),(c),(d),(e) and (f) represent the progress of the test

In Figure 6.4, the real trajectory is shown through rviz [1]. It shows in real time the cameras visualisation, the global way points (green arrows), the desired trajectory (red lines), the completed path (white lines), the real trajectory (yellow lines) and the disparity which is the cloud of 3D points in front of the vehicle used to reconstruct the environment and to detect objects.

Conclusion of the real test

The real test has achieved a successful result being the vehicle able to follow the trajectory as Figure 6.4 shows. The controller adjustment is a difficult task that requires many trials. The vehicle does not correct the error in a quick way because the controller has been adjusted to mitigate the error providing smooth control actions to the vehicle actuators.

The stability of the control loop is guaranteed and a moderate level of robustness has been reached being able to drive the vehicle along the trajectory without needing the knowledge about physical vehicle parameters. This indicates that this controller could be used in other vehicles obtaining similar results working at low speeds, which is an interesting conclusion.

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

Chapter 7

Effects on economy, society and environment

In this section, an introduction of possible impacts on economy, society and environment are presented. It is divided in two parts: socioeconomic and environment effects.

7.1 Socioeconomic impact

Vehicles have always had a huge impact on society and economy. In some countries, the vehicle production corresponds to a high percentage of its industry. Then it is important to take care of how vehicles are integrated in society. In the same way as in the past vehicles contributed to improve people life style, now autonomous vehicles pretends to enter in society with the same idea: enhance our life quality.

The autonomous vehicle development will contribute to the traffic safety and mobility improvement. The number of accidents caused by human errors will be reduced and the necessity of learning to drive will be also reduced. The mobility of old people and people with disabilities will be enhanced.

The control and planning system developed in this thesis aims to compute a comfortable trajectory and follow it with the less possible error, which leads directly to an increment of the road safety.

The development of national research and technology transfer projects in the autonomous driving field will contribute to the technological country progress. In detail, this project could have an important socioeconomic impact for Catalonia for being in a strategical industrial zone where there are located industries like SEAT, NISAN and APP+ IDIADA.

It is important to note that the autonomous driving advances can be transferred to other automotive fields in which a high accuracy is not required. For instance, trains whose lateral movement is constrained.

7.2 Environmental impact

There exist many contamination types, but society is interested in two: noise and air pollution. With the introduction of electric autonomous vehicles, we will achieve a world without uncomfortable noise in the streets, not only for the fact of being electrics vehicles but also for the fact that all vehicles will be interconnected and the car horn could be suppressed. Regarding the pollution, the electric motor does not contaminate. It means a significantly reduction of the world atmospheric pollution. The idea of having a net of connected autonomous vehicles would reduce the number of vehicles in the streets, what would lead to an increment of parking space.

Until this moment all effects discussed were positive but there also exist negative impacts. One of them is that the vehicle batteries have a limited life, it means that we would have to replace them every defined period with the consequent accumulation and contamination. But the most important point is that, according to [18], manufacturing an electric vehicle generates as carbon emissions as building a conventional car due to its battery.

Chapter 8

Project budget

In this section, we provide some considerations about the development and implementation of the control and planning system from an economic point of view. It will be considered the study of the costs as whether the system was to be introduced in the market.
The unitary cost of the system would be composed by:

- the cost of the master computer in which the algorithms run (C_{PC}): 800 €
- the cost of automating the vehicle (installation of actuators and sensors) ($C_{A\&S}$): 3500 €
- the cost of development. It is composed by:
  - Software development (C_{Dev}): 700000 €
  - Technical support (C_{Ts}): 3000 €
  - Tests (C_{Tests}): 1000 €
  - General costs like water, electricity and others ($C_{General}$): 2000 €

Thus, the unitary cost of equipping the control and planning system to a vehicle can be calculated as:

$$C_{CPsys} = C_{PC} + C_{A\&S} + \frac{C_{Dev} + C_{Ts} + C_{Tests} + C_{General}}{n_{Vehicles}} \tag{8.1}$$

where $n_{Vehicles}$ reflects the number of vehicles equipped with this system. This number is estimated to be around 500000 units. Therefore, the unitary cost of applying this system to a vehicle is 4301.4 €. The equipment of this new system would have an important impact in the vehicle price. However, it is

difficult to estimate the real unitary cost. In order to forecast it, a more sophisticated cost analysis would have to be done.

Chapter 9

Concluding remarks

9.1 Conclusions

In this work, we have dealt with the autonomous driving problem from the planning and control point of view. We have studied the equations that govern the vehicle behaviour presenting three different models. A six degrees of freedom model which has served to understand the car dynamics and will be used for adjusting future complex simulators. A three degrees of freedom dynamic model which is a simplification of the complete model is proposed and finally a kinematic bicycle model is introduced for control design in this thesis.

Using this kinematic model two nonlinear control techniques have been used to design a controller for the autonomous vehicle. The first one is based on the direct Lyapunov design such that the controller is derived from an error model. The second approach is the Sliding Mode Control technique that is based on deriving a controller that forces system dynamics to reach the sliding surface and remaining there what ensures a null error. Furthermore, a trajectory planner to provide the current reference to the controller has been implemented.

The control planning system has been tested both in simulation and in the real autonomous vehicle. The first step has been to evaluate the algorithms in Matlab/Simulink obtaining good results. Then, they were integrated in a realistic driving simulator based on Unity obtaining successful results in spite of the higher vehicle model complexity. Finally, they were integrated in the autonomous vehicle and they were tested in a real road scenario obtaining good results.

Some aspects to be remarked:

- Simulation is very useful to develop, adjust and test the control algorithms. On the one hand Matlab is a powerful tool which has been used to perform a lot of elementary tests. It is one of the best tools for debugging algorithms because of its simplicity. On the other hand, the Unity software has been employed to test the controllers in a more complex and realistic simulated world. Using Unity, a city scenario has been built with roads, buildings, pedestrians and other vehicles which give to the developer a better perspective of the vehicle behaviour. But the most suitable thing is the possibility of testing the algorithms over a even more complex vehicle model than the explained six degrees of freedom model in Chapter 3.

- We realised of the higher importance of the trajectory planner for autonomous vehicles field. We can conclude that for autonomous vehicles, a trajectory planner that provides a constant velocity reference is a better option than a planner that provides a variable speed reference.

- Both control techniques provide very good results in simulation obtaining moderate levels of robustness in terms of parametric uncertainty since the model taken into account for developing the control (kinematic) only consider one vehicle parameter. Therefore, we can conclude that the considered nonlinear control techniques allow to perform a good control based on simplified models.

- In real tests they appear problems such as a bad GPS localization among others that make more difficult the correct development of the desired proof since they appear longitudinal and lateral errors. Hence, complex localization algorithms are essential to perform a good control task.

- The Lyapunov control law adjustment has resulted to be more easy than expected, not only in simulation but also in real scenario. It can be concluded this control method is robust, easy to adjust and applicable to other vehicles without any knowledge about them.

- The SMC control law adjustment has resulted to be more complex than Lyapunov law because it has seven parameters to tune. However, in Unity SMC law has achieved a higher performance obtaining a lower percentage of error a long the test. So, it can be concluded this control method is also quite robust, it just depends on one vehicle parameter and it is complex to adjust but it obtains very good results when is properly adjusted. This last drawback suggests that an optimal adjustment law could be very interesting.

9.2 Publications

During this master thesis development the following publication has been produced:

- Eugenio Alcalá, Laura Sellart, Vicenç Puig, Joseba Quevedo, Jordi Saludes, David Vázquez and Antonio López. Comparison of two non-linear model-based control strategies for autonomous vehicles. 24^{th} IEEE Mediterrian Conference on Control and Automation, 2016, Barcelona.

9.3 Future work

During this master thesis a lot of ideas have emerged and some of them are commented down below:

- To reprogramme the low level control board in order to obtain an steering angle accuracy of at least one degree.
- To reprogramme the velocity task in order to control the electric motor torque directly instead of the longitudinal velocity of the vehicle.
- To prove the SMC law in the real vehicle in order to be able to compare with Lyapunov technique.
- To develop an adaptive algorithm to adapt the control parameters in an optimal way.
- To prove both control laws in other real situations with higher velocities and more complex circuits.
- To develop these control techniques using the dynamic model for that and compare with the kinematic one in Matlab and Unity.
- To implement the adaptive parameters law for this new dynamic control and proving it at higher velocities in the real scenario.
- To study the delays of the control loop and try to take them into account in the control algorithm.

Bibliography

[1] ASIRT. Association for safe international road travel.

[2] BIANCO, C. G. L., PIAZZI, A., AND ROMANO, M. Velocity planning for autonomous vehicles. *IEEE Intelligent Vehicles Symposium* (2004).

[3] BROGGI, A., MEDICI, P., ZANI, P., COATI, A., AND PANCIROLI, M. Autonomous vehicles control in the vislab intercontinental autonomous challenge. *International Federation of Automatic Control* (2011).

[4] CARVALHO, A., LEFÉVRE, S., SCHILDBACH, G., KONG, J., AND BORRELLI, F. Automated driving: The role of forecasts and uncertainty — a control perspective. *European Journal of Control* (2015).

[5] E.DIXON, W., M.DAWSON, D., ZERGEROGLU, E., AND BEHAL, A. *Nonlinear Control of Wheeled Mobile Robots*. Springer, 2000.

[6] E.SLOTINE, J.-J., AND LI, W. *Applied Nonlinear Control*. Prentice Hall, 1991.

[7] FERNÁNDEZ, J. G. *A Vehicle Dynamics Model for Driving Simulators*, 2012.

[8] GANNES, L. Google introduces new self driving car at the code conference. *Google official blog* (2014).

[9] GAO, W., HUNG, J. Y., AND HUNG., J. C. Variable structure control: A survey. *IEEE* (1993).

[10] GAO, Y., GRAY, A., TSENG, H. E., AND BORRELLI, F. A tube-based robust nonlinear predictive control approach to semi-autonomous ground vehicles. *Vehicle System Dynamics: International Journal of Vehicle Mechanics and Mobility* (2014).

[11] GEIGER, A., LAUER, M., MOOSMANN, F., RANFT, B., RAPP, H., STILLER, C., AND ZIEGLER, J. Team annieway's entry to the grand cooperative driving challenge 2011. *IEEE Transactions on Intelligent Transportation Systems , vol 13*, no 3, (2012), 1008–1017.

[12] HALLECK, T. Google inc. says self-driving car will be ready by 2020. *International Business Times* (2015).

[13] K. KHALIL, H. *Nonlinear systems*. Prentice Hall, 1996.

[14] KANG, J., HINDIYEH, R. Y., MOON, S.-W., GERDES, J. C., AND YI, K. Design and testing of a controller for autonomous vehicle path tracking using gps/ins sensors. *Federation of Automatic Control* (2008).

[15] LEVINSON, J., BECKER, J. A. J., DOLSON, J., HELD, D., KAMMEL, S., KOLTER, J. Z., LANGER, D., PINK, O., PRATT, V., SOKOLSKY, M., STANEK, G., STAVENS, D., TEICHMAN, A., WERLING, M., AND THRUN, S. Towards fully autonomous driving: Systems and algorithms.

[16] MARINO, R., SCALZI, S., AND NETTO, M. A nested pid steering control for lane keeping in vision based autonomous vehicles. *Control Engineering Practice , vol 19* (2011), 1459–1467.

[17] NAM, K., OH, S., FUJIMOTO, H., AND HORI, Y. Design of adaptive sliding mode controller for robust yaw stabilization of in-wheel-motor-driven electric vehicles.

[18] NEALER, R., REICHMUTH, D., AND ANAIR, D. Cleaner caars from cradle to brave. how electric cars beat gasoline cars on lifetime global warming emissions. *Union of Concerned Scientists* (November 2015).

[19] RAJAMANI, R. *Vehicle Dynamics and Control*. Springer, 2012.

[20] SAMSON, C., AND AIT-ABDERRAHIM, K. Mobile robot control. part 1 : Feedback control of nonholonomic wheeled cart in cartesian space. *HAL-INRIA* (24 May 2006), 30–34.

[21] SIENEL, W. Estimation of the tire cornering stiffness and its application to active car steering.

[22] SNIDER, J. M. *Automatic Steering Methods for Autonomous Automobile Path Tracking*, 2009.

[23] SOLEA, R., AND NUNES, U. Trajectory planning and sliding-mode control based trajectory-tracking for cybercars. *Integrated Computer-Aided Engineering , vol 14* (2007), 33–47.

[24] TAGNE, G., TALJ, R., AND CHARARA, A. Higher-order sliding mode control for lateral dynamics of autonomous vehicles, with experimental validation. *IEEE Intelligent Vehicles Symposium (IV)* (2013), 678–683.

[25] URMSON, C. The self-driving car logs more miles on new wheels. *Google official blog* (2012).

[26] ZHAO, P., CHEN, J., DONG, Y., TAO, X., XU, T., AND MEI, T. Design of a control system for an autonomous vehicle based on adaptive-pid. *International Journal of Advanced Robotic Systems* (2012).

[27] ZIEGLER, J., DANG, T., FRANKE, U., LATEGAHN, H., BENDER, P., SCHREIBER, M., STRAUSS, T., APPENRODT, N., KELLER, C. G., KAUS, E., STILLER, C., HERRTWICH, R. G., AND ET AL. Making bertha drive — an autonomous journey on a historic route. *IEEE Intelligent Transportation Systems Magazine , vol 6* (2014), 8–20.