**Universidad Politécnica de Madrid**
Escuela Técnica Superior De Ingenieros Industriales
Doctorado en Automática y Robótica

# Path Following Control System for car-like Unmanned Ground Systems.

**A End of Master Work submitted for the degree of**
***Master in Automation and Robotics***

**José Luis Sánchez López**
Industrial Engineer

**2012**

Escuela Técnica Superior De Ingenieros Industriales
Doctorado en Automática y Robótica
Universidad Politécnica de Madrid

# Path Following Control System for car-like Unmanned Ground Systems.

**A End of Master Work submitted for the degree of**
***Master in Automation and Robotics***

Author:
**José Luis Sánchez López**
Industrial Engineer

Director:
**Dr. Pascual Campoy Cervera**
PhD Full Professor Industrial Engineer

**2012**

*A mis padres.*

*A mis hermanos.*

*A toda mi familia.*

*A los que me han dado fuerzas cuando no las tenía.*

*A los que me han visto evolucionar.*

*A los que me han visto caer (o me han tirado) y me han visto levantarme hasta convertirme en una máquina.*

*A los que me han acompañado en mi viaje por la Universidad.*

*A aquellos con los que hay un incendio cada vez que nos juntamos.*

*A todos mis amigos y compañeros.*

*A ellos estas palabras de agradecimiento por haberme dado siempre constante apoyo y cariño.*

*Jose Luis.*

# Acknowledgements

It is a pleasure to thank those who made this Master Work possible. First of all, I am heartily thankful to my supervisor, professor Pascual Campoy Cervera, whose encouragement, guidance and support given during the last three years.

I would like to show my gratitude for all the past and current members of the Computer Vision Group, among them, must be mentioned, the professors José María Sebastian and Sergio Dominguez, Iván Mondragón, Carol Martínez, Miguel Olivares, Ignacio Mellado, Jesús Pestana, Changhong Fu, David Galindo.

Thanks to all the CAR members, from whom I received an invaluable collaboration and support.

I would like to thank institutions like the Consejo Superior de Investigaciones Científicas (CSIC) for the Author's scholarship through the program "Junta de Ampliación de Estudios. Becas predoctorales 2010"; SIEMENS, S.A. that made possible this Master Work through several contracts; and the INSIA-UPM and the people at LABIE for their support and the provided facilities.

I also want to thank to the Autonomous Systems Technologies Research & Integration Laboratory (ASTRIL) of the Arizona State University (ASU) - Arizona (USA), and specially to professor Srikanth Saripalli, for give me the opportunity to visit, collaborate and gain new experiences in both the personal and the researching level.

I'm very grateful with all my family; my parents, my sister and brother; and my friends. I'll be forever grateful with all of them for the support and confidence that they have give me.

Finally, thanks to all the persons that have collaborated on the development of this Master Work and have not been mentioned.

# Abstract

In this Master Work, a new High-Level Control System is proposed for the path following control problem for two car-like Unmanned Ground Systems (UGSs) Prototypes. The first prototype, based on a Citroën C3 is small and it has light weight and small wheels. The second one, based on an Iveco Eurocargo is bigger than the other and it needs a more complex High-Level Control System.

This High-Level Control System is composed by a State Estimator and by a Controller. The State Estimator used all the measurements and all the available info to calculate the most precise state of the UGS. Then, the Controller, composed by a Feed-Forward controller, a Feed-Back Controller and a Wheel Dynamic Compensator, generates the steering reference to achieve the path following problem.

This High-Level Controller has been tested by lots of Simulations and has been implemented in a real UGS. The goodness of the High-Level Controller is demonstrated through an increase of the performance in the path following displayed with a high number of laps achieved without lose the circuit (more safety), a high speed of the UGS, and more comfort for the UGS passengers.

# Nomenclature

The conventions used in this document are defined as follows:

- Matrix will be represented by capital letters such as R and H.
- Vectors will be represented by a lower case letters with or without a upper arrow, such as $\vec{x}$ or x.
- A vector is a column vector, unless indicated otherwise.
- Estimated variables will be represented by a hat, such as $\hat{\theta}_e$.
- Measured variables will be represented using a tilde, such as $\tilde{d}$

# Acronyms

Following is a description of most commonly acronyms referenced on this work:

**CAR:** Centre for Automation and Robotics

**CAR UPM-CSIC:** Centro de Automatica y Robotica, centro mixto Universidad Politecnica, Consejo Superior de Investigaciones Cientificas

**CDTI:** Centre for the Development of Industrial Technology

**CSIC:** Consejo Superior de Investigaciones Cientificas

**CV:** Computer Vision

**CVG:** Computer Vision Group

**DGT:** Direccion General de Trafico

**ECU:** Electronic Control Unit

**EKF:** Extended Kalman Filter

**GA:** Genetic Algorithm

**HMI:** Human Machine Interface

**INSIA:** Insituto de Investigacion del Automovil

**KF:** Kalman Filter

**NN:** Neural Network

**PLC:** Programmable Logic Controller

**UGS:** Unmanned Ground System

**UGV:** Unmanned Ground Vehicle

**UPM:** Universidad Politecnica de Madrid

**WHO:** World Health Organization

# Contents

# List of Figures

# List of Tables

# Chapter 1

# Introduction

The main application of this Master Work is the automation of the driving task, achieving the development of driver-less car. The main advantages of this driver-less cars are that they can get more efficient displacements, and more comfortability and safety of the passengers.

## 1.1. Motivation

Why automate the driving task? As it is said before, the main reasons are efficiency in the displacements and comfortability and safety of the passengers.

According to the World Health Organization (WHO), traffic accidents causes annually the loss of more than 1.2 millions of lives, and between 20 and 50 millions of non-fatal injuries [see (de la Salud, 2009)]. If not remedied, the WHO says that in 2020, traffic accidents will constitute the third leading cause of injury and overall of the death worldwide.

According to the Direccion General de Trafico (DGT) [see (de Trafico, )], in the 2003, traffic accidents caused for a driver distraction were the 24.9% of the whole traffic accidents, been the most frequent factor. The following factors of traffic accidents were inappropriate velocity of the vehicle (20.3%), offside manoeuvre (13.9%) and invasion of the other-side lane (11.6%). Only these four factors represents the 70.7% of the traffic accidents, been all of them human errors. The automation of the driving task could avoid all the accidents due to human errors, avoiding lots of deaths and injuries.

On the other hand, we can explore not only the safety, but also the efficiency of the displacements achieved with the driver-less cars. The human response time is, in average, 0.2 seconds, although it depends of lots of factors like age, training, tiredness, amount of alcohol and drugs consumed, ... However, the response time of the electronics of the car are few milliseconds (at most 0.04 seconds). The fact of responding more than 5 times quicker than humans, allow the driver-less cars, keep less safety distance respect to the previous vehicle that the human driver cars. Thanks to that, a greater packing of the vehicles along the road would be possible, achieving a more efficient use of the roads, decreasing the traffic jams [see (Francisco Aparicio, 2008)]. This efficient use of the roads and the decreasing of the traffic jams contribute to save money because of the saving of fuel and the reduction of the size needed for the roads. This also implies not only economical efficiency, but also environmental efficiency, reducing the pollution because of the reduction of the fuel and the environmental impact because of the reduction of the size of the roads.

Other advantage of the driver-less cars, in addition of the safety of the passengers and the efficiency of the displacements commented before, are the social aspects. On the one hand, the reduction of the traffic jams, implies a reduction of the displacement time, obtaining improvements for the reconciliation of family life since according to [(y Empleo, 2010)], the 64% of workers use the car to go to work in Spain, been the traffic highly congested in the cities. There are other infinite social improvements due to the driver-less car like the usage of the vehicles for disabled people, or like the possibility of execute other tasks in the vehicle without any risk like reading or using a PC.

## 1.2. State of Art

Driver-less cars are a popular topic nowadays. The advances during the last decade in the hard test-bed of DARPA [(dar, 2012)], figure 1.1, have given a strong push-up to this research line.

But it was the recent work of Google [(goo, 2010)] that gives the final shot to popular interest, figure 1.2. This project is managed by Sebastian Thrun who obtain successful results in DARPA challenge in 2005 [(S. Thrun and Mahoney, 2006)] and 2008 [(M. Montemerlo and Thrun, 2008)]. The excellent results of this approach force legal changes to give the first license to a self-driving car. Google has a fleet of at 8 cars, each with at least 4 different kinds of sensors. The information acquired by these sensors is merged with map database.

Before this popular event happened, a number of important projects had been done around the world using rather different approaches. In Europe the PROMETHEUS project (PROgraM for a European Traffic with Highest Efficiency and Unprecedented Safety) started in 1986. The fantastic results of this

Figure 1.1: Darpa Urban Challenge Logo

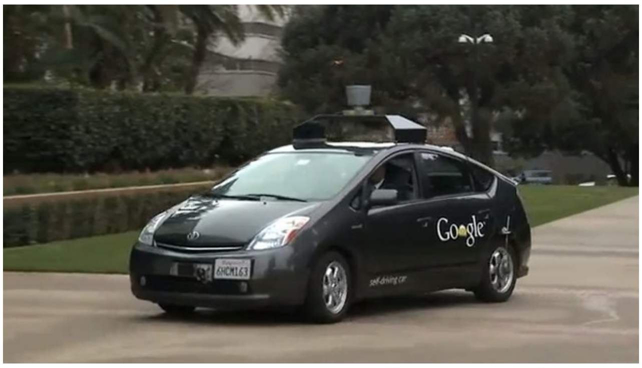


Figure 1.2: Google Inc Driver-less car

project are regarded as milestones in vehicular robotics history. Two of them were ARGO by VisLab [(Bertozzi et al., 1998)], [(Broggi et al., 2000)] and VaMoRs [(R. Behringer, 1998)] tested on 1998. Both of them use two cameras to detect road lanes and to avoid obstacles but implement different algorithms and strategies. In 1995 in the United States started the NAHSC project (National Automated Highway System Consortium) inside the California PATH (Partners for Advanced Transit and Highways) program [(Shladover, 2007)]. In 1997 was exhibited the important Demo 97 in San Diego in some cars that were guided by a magnetic guided line inside the asphalt. A array of different sensors have been installed in those cars to execute self driving tests and automated platoon formations of 8 cars. In 2010, the multidisciplinary European project SARTRE used new approaches in platoon formations and leader systems to successfully present an autonomous platooning demo travelling 120 miles [(Sar, 2012)]. The caravan was composed by one human-driven truck followed by 4 cars equipped with cameras, laser sensors, radar, and GPS technology. A complete test of different systems of leader following, lane and obstacle detection and terrain mapping has been done by the VisLab. In 2010, the laboratory directed by Alberto Broggi covered 15926 km from Parma to Shanghai with a convoy of 4 motor homes [(Vis, 2012)]. All of then were equipped with 5 cameras and 4 laser-scanners. No road maps were used. The first vehicle drove autonomously in selected sections of the trip, and other vehicles were 100% autonomous using the sensors and the GPS way-points sent by the leader vehicle.

In the national level, the most representative driver-less car is the AUTOPIA Project [(Aut, 2012)], figure 1.3, developed by the Centre for Automation and Robotics (CAR), Universidad Politecnica de Madrid (UPM) and Consejo Superior de Investigaciones Cientificas (CSIC), which intend to transfer the techniques developed for the control of autonomous robots to vehicle control, changing as little as possible the environment in which they have to evolve.

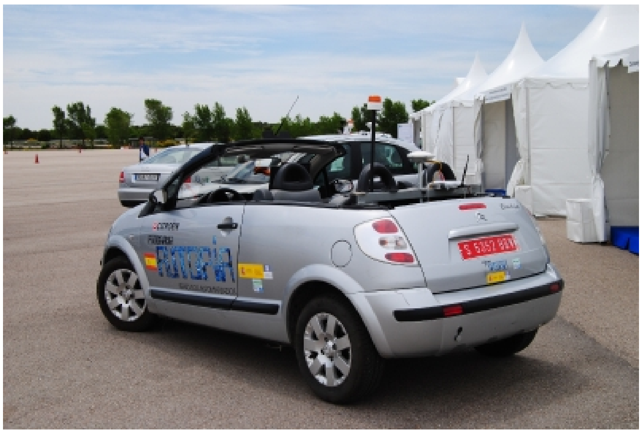


Figure 1.3: Autopia Project Driver-less Car

Most of the previous approaches use vision algorithms or sophisticated sensors to detect road lanes. But in city environments nowadays, traffic and cars parked on both sides of the streets make it almost impossible to automatize this kind of approach, as is shown inn Figure 1.4. Furthermore, the streets are full of shadows produces by trees, buildings, and other city structures causing big variations in brightness in the images. These kinds of visual effects are usually detected as occlusions by visual algorithms. Public transports take the same path every hour and every day.

In this Master Work is presented a low cost approach to a visual based driver-less vehicle robust against brightness variations. As it is described in chapter 3, the Unmanned Ground System (UGS) just has one camera and doesn't use any GPS information. A visual system has been developed to follow a predefined path and to get information about the vehicle position.

Figure 1.4: The high level of traffic nowadays in every city make the detection of the lane almost impossible

## 1.3. Master Work Organization

The Master Work is organized as follows: In chapter 2 is described the specific objective of the Master Work, and the framework and the objective of the Project in which was the Master Work included. In chapter 3, the two UGSs used, and the test circuits are described. The whole proposed High-Level Control system is shown in chapter 4, been described in chapter 5 the first part that includes the state estimation of the car, and in chapter 6 is described the pure control system. In chapters 7 and 8, results under simulation and under the real UGSs are shown respectively. Finally, chapter 9 concludes the Master Work.

# Chapter 2

# Objectives of the Master Work

This Master Work is only one part included in the Technology Transfer Project entitled ”Visually-Guided Driver-Less Car”, that is described bellow.

## 2.1. Framework of the Project

The ”Visually-Guided Driver-Less Car” Technology Transfer Project is compounded for three time-consecutive contracts between the contracting company (SIEMENS, S.A.) and the Computer Vision Group (CVG) of the CAR UPM-CSIC. Each contract has different objectives. These contracts are the following:

- Visual Guidance Feasibility Study for an Outdoor Urban Vehicle (November 2007 - March 2008)
- Visual Guidance of a Commercial Compact Car (October 2008 - December 2010)
- On board Control and Visual Guidance of a New Prototype of Urban Vehicle (April 2010 - December 2011)

The last contracts was partially funded by the Centre for the Development of Industrial Technology (CDTI).

In the two last contracts, the Insituto de Investigacion del Automovil (INSIA) was a partner. In chapter 3 is given more information about the work developed by INSIA.

After of the end of the contracts, maintenance tasks, car improvements and new tests were done.

The author of the Master Work joined at the CVG CAR UPM-CSIC in October 2009, and he began to work in the Project in a theoretical study.

The author of the Master Work, also developed his Proyecto Fin de Carrera (for the degree of Ingeniero Industrial, intensificacion Automatica-Electronica por la UPM) in the framework of this Project, whose defense was in November 2010, getting the maximum grade of Matricula de Honor.

In June 2010, he began to work on the Project with the real prototype. Since June 2011 until the official end of the Project in July 2012, the author of the Master Work was the responsible of the Project under the direct supervision of Dr. Pascual Campoy.

## 2.2. Objectives of the Project

For the understood of this Master Work, some partially objectives is explained in this section. The final objective of the Global Industrial Project was the development of a Visually-Guided Driver-Less Car. Some of the objectives of the Projects related with the Propulsion System and with the applications of the Project are confidential under contract with the company.

There were three prototypes, one for each contract. The first prototype was a little model of a car (a remote control car toy) autonomously visually-guided that is not described in this Master Work. The second an the third prototype are described in chapter 3.

The vehicles are guided thanks a camera looking downwards, that can see a path painted on the floor that it has to follow (paint path). This camera also can get other visual information painted on the floor (for more details see chapter 3). The vehicles have the steering autonomously controlled in order to follow the path. The velocity is selected manually for the user of the vehicle.

There is also other important path for the vehicles. It is the magnet path. Because of the confidentially clause, more information about that path, cannot be given. For this Master Work, the magnet path is considered as a path that it has to be followed by other known point of the vehicle, called magnet point. This magnet path cannot be measured and it is necessary to assume that there is a paint path painted that, if it is followed correctly, the magnet path will be followed correctly too.

There are tree kind of previous knowledge had about the path information:

- There is no knowledge about any path to follow.
- There is information about the magnet path, but not about the paint path.

- There is only information about the paint path, and the magnet path is not cared.

The hardest case is the first one, and the easiest is the last one.

The objectives of each contract were:

- Visual Guidance Feasibility Study for an Outdoor Urban Vehicle: the main objective was the development of a little model of a visually-guided driver-less car part of a feasibility Study.
- Visual Guidance of a Commercial Compact Car: the main objective was develop a real-size visually-guided driver-less compact car.
- On board Control and Visual Guidance of a New Prototype of Urban Vehicle: the main objective was the development of a visually-guided driver-less car with big size and big weigh, and the improvement of the control scheme of the compact car of the second contract.

Because of economical reasons of Siemens S.A., the last contract reduced its objectives. The bigger and more weigh prototype could not be finished, and the improvements of the compact car finished in advance.

## 2.3. Objectives of the Master Work

The main objective of the Master Work is the development of a full path following (see appendix A) Control System for the visually-guided driver-less vehicles.

This system attempt to improve the old control system described in chapter 3 in order to:

- Improve the security at the path following, avoiding the lost of the path and minimizing the effect when it happens.
- Increase the velocity of the car while the path is followed.
- Improve the comfortability of the passengers of the car.
- Increase the weight of the driver-less car.

This performance improvement needs the development of a High-Level Control System composed by a state estimator and a controller.

The state estimator needs to have:

- A wheel position estimator
- A UGS speed estimator
- A UGS position over the circuit estimator

- A circuit curvature estimator
- A line lost behaviour

The controller needs to be composed by:

- A wheel dynamics compensator
- A feed-forward controller
- A feed-back controller

Chapter 3

# Unmanned Ground Systems & Test Circuits

In this chapter the Unmanned Ground System (UGS) Platforms and the Test Circuits are described. In section 3.1 the first prototype is described. This prototype, based on a Citroën C3 Pluriel, was developed before the beginning of this Master Work and only the High-Level Control system (described in section 3.1.3) is replaced for the one described in chapter 4. A new prototype based on a Iveco Eurocargo is developed in section 3.2. Because of economical problems, the project finished in advance and the proposed High-Level control system never could be implemented in this prototype (see section 2.2). The last section (section 3.3) describes the Test Circuits in which the developed controller is tested. Because of the confidentially clause of the Project, the UGSs and the whole Test Circuits are not described. Only the most relevant things for this Master Work are described.

## 3.1. First Prototype based on a Citroën C3 Pluriel

The official development of this first prototype ended in December 2010 (see section 2.1). From this date, the prototype was used just for make test and for improve the control system with the one described in chapter 4

As it is explained in chapter 2, in the development of this UGS, three different partners have participated: SIEMENS S.A., INSIA and CVG. Each one has developed one different part of the UGS as it can be seen in figure 3.1

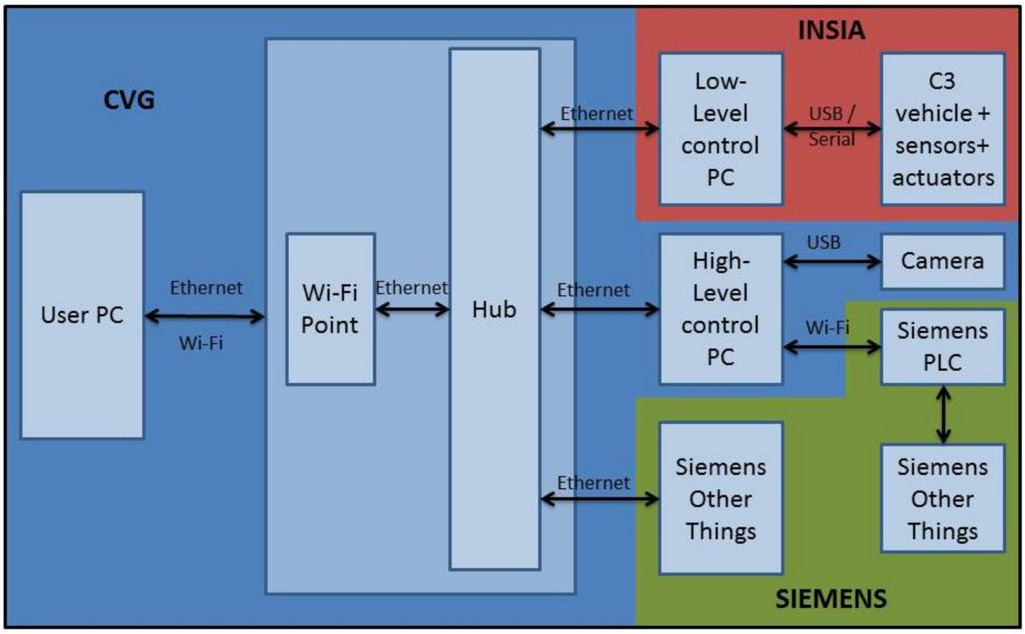


Figure 3.1: Physical Connexions, PCs and main things of the C3 Platform. In dark blue, things developed by CVG; in red things developed by INSIA and in green things developed by SIEMENS

In this Master Work, the UGS is used as an user, changing only the High-Level Controller, because of that it is only described in a user point of view.

### 3.1.1. Vehicle

The first prototype of the UGS uses a Citroën C3 Pluriel (figures 3.2 and 3.3) as the vehicle platform.

It has the following characteristics that are relevant for the Master Work:

- Distance between axis: $L = 2.46m$.
- Maximum turn of the steering wheels: $\phi_{max} = 26°$.
- The wheels have small tires and are over inflated, therefore we can consider them as rigid in control tasks.

### 3.1.2. Low Level

The Citroën C3 compact car was modified by the partner INSIA to provide the UGS platform. INSIA installed on the car some actuators, some sensors and

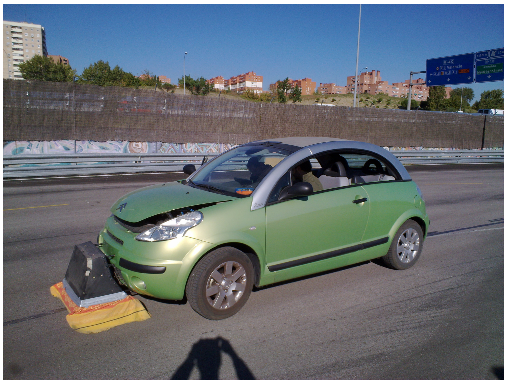


Figure 3.2: C3 platform front view

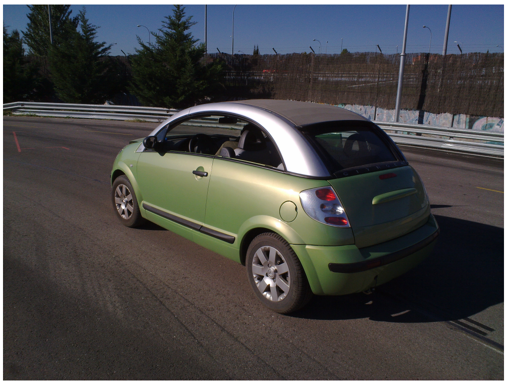


Figure 3.3: C3 platform rear view

a data acquisition card in communication with a PC (see figure 3.4). This PC, designated as ”Low-Level PC” is in charged of the Low-Level control of the UGS. It is communicated with the High-Level PC giving it the speed of the car, the angle of the steering and the emergency state of the vehicle (if enabled). The Low-Level PC receive from the High-Level PC the commands for changing the speed of the vehicle, the steering and the emergency state.

As the camera installed is only used in the high-level system, it was not included in the Low-Level system.

As the Low-Level controller is not an objective in this Master Work and is only an instrument, it was not described with precision and in depth. More information can be found on ()

### Actuators

The C3 platform has three main actuator, the steering actuator, the gas actuator and the brake actuator. The gas is actuated by using the Electronic Control Unit (ECU) of the C3. The Steering is actuated throw the aided steering of the car thanks again to the ECU. The brake is mechanically actuated thanks a electric DC motor.

### Sensors

INSIA installed on the C3 the following sensors: a steering encoder, to measure the position of the steering; a brake encoder, to measure the position of the motor of the brake and a radio receiver to enabled the option of make an emergency stop using a radio button. INSIA get the measurement of the speed of the vehicle using the speedometer of the C3.

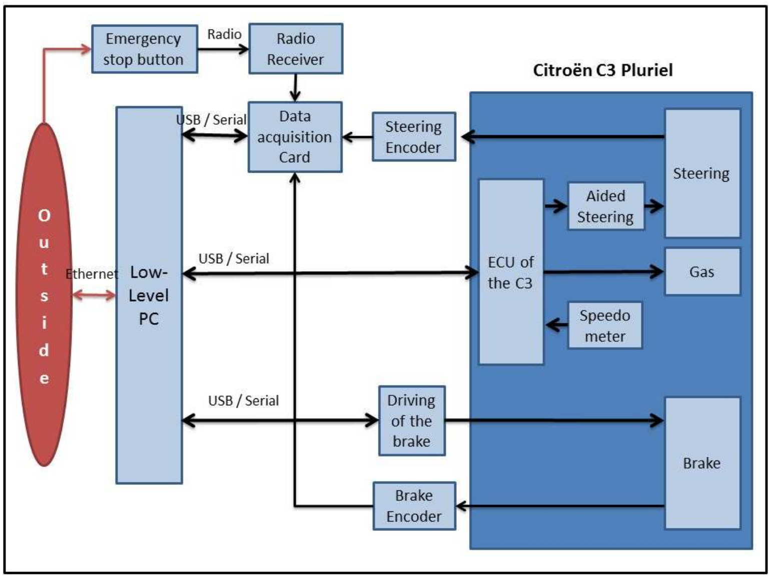


Figure 3.4: Physical Connexions, Sensors, Actuators, PC and main things installed on the C3 platform by INSIA

### Communication System

The internal communication of the Low-Level system is not detailed in this Master Work because it is not needed.

### Low Level Controllers

The Low-Level has internal controllers for the speed of the vehicle and for the steering. They are working over the Low-Level PC. The objective of these controllers is that the reference of speed and steering turn received from the High-Level PC is followed. Fuzzy Logic Controller optimized using Neural Network are used for these proposes.

## 3.1.3. High Level

The High-Level system is a key part of the C3. It is has a lot of objectives:

- Is the interface with the Low-Level.
- Is the interface with the user.
- Is the interface with some other things of Siemens that will not be described.
- Is has the Computer Vision Processing and it is the interface with the camera.

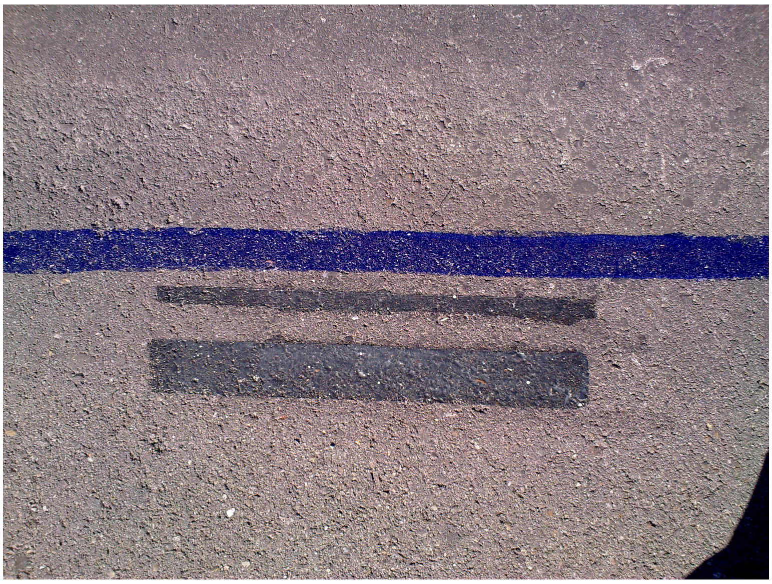


Figure 3.5: Line and code painted on the floor.

- It is in charged of the autonomous path-following of the vehicle (objective of this Master Work) and labelled as ”High-Level Controller”.
- It is in charged of keep everything working correctly thanks to the designated ”Very High Level Control System”

The High-Level PC has all the interfaces, the CV processing, the High-Level controller and the Very High-Level control System.

### Very High-Level Control System

The Very High-Level Control System of the UGS keeps everything working correctly. It is running on the High-Level PC. The Very High-Level Control System is not described in this Master Work.

### Computer Vision System

The UGS was design to follow a painted line on the floor. To keep this line away from possible vandalism we used an invisible paint that looks blue under ultraviolet light emission (see figure 3.5). An ultraviolet bulb and camera were installed inside a metal structure at the front of the vehicle (distance from the centre of the rear axis of the vehicle: $l_{1-c} = 3.41m$, $l_{2-c} = 0$). The principal aim of this structure is to control the illumination. The structure limits the field of view of the camera. The size of the field of view is just 30 x 50 cm. See figure 3.6

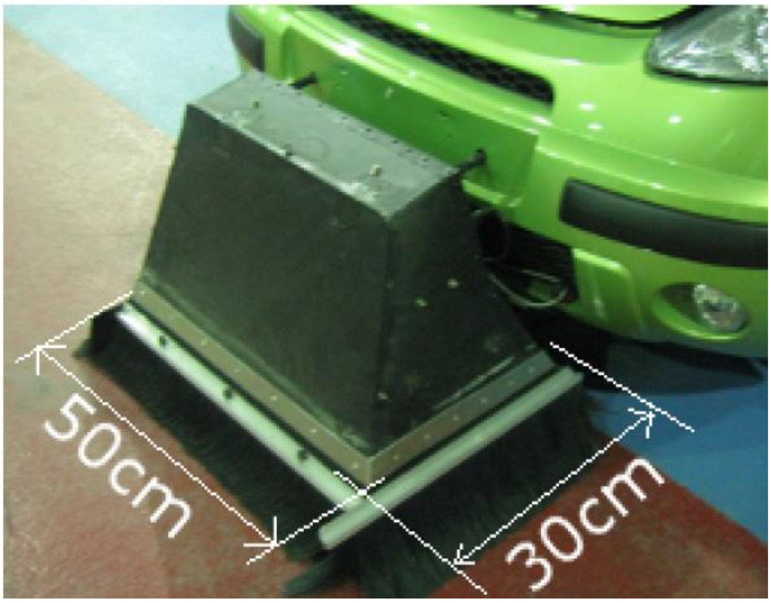


Figure 3.6: Metal structure of the visual system. Limits the field of view of the camera to 30 x 50 cm.

The images captured by the front camera under UV light are processed in real time by a computer vision algorithm. The line is sawn by the camera as blue and the code as yellow (see figure 3.7). The codes are always at the right side of the line. The algorithm detects the line to be followed by the vehicle (distance to the centre of the image $\tilde{d}$ and angle $\tilde{\theta}_e$, see figure 3.8) and visual marks, labelled as codes.

The visual algorithm has been designed with robustness in mind. It can detect fragmented lines due to occlusions or irregularities on the concrete. The system has been tested under different weather conditions. Lines are detected and marks are decoded successfully at up to 30 k/h.

The CV system works at a frequency of $f = 30Hz$.

More information about the processing algorithm is not needed for the objective of the Master Work.

### Communication & Software

The Physical connexions of the High-Level can be shown in figure 3.1, in dark blue. In figure 3.9, more details about the communication can be founded.

The High-Level is developed under a Ubuntu Linux Operative System. The system is programmed in C++ and the code of the High-Level controller is embedded is the code of the Very-High Level Controller. For the developing of the High-Level Control system some variables of the Very High-Level Controller

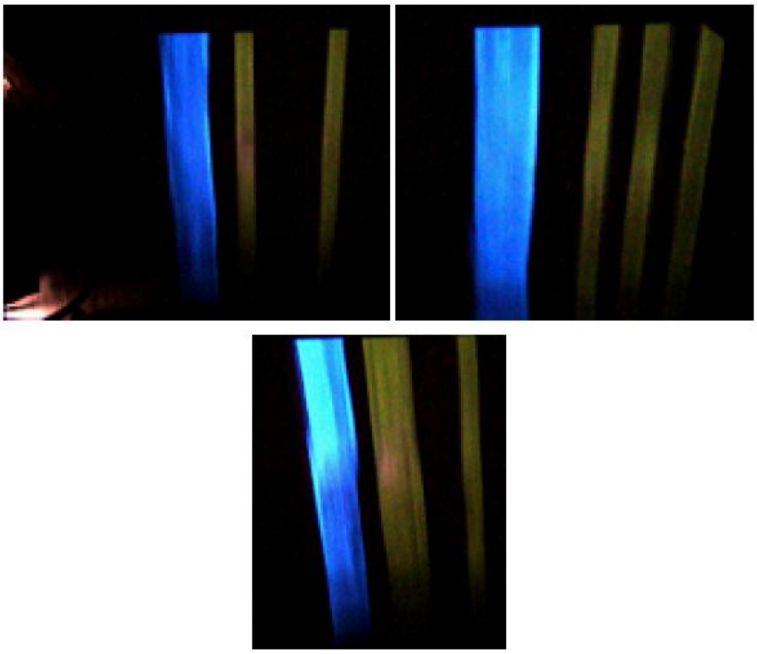


Figure 3.7: Captured images of the camera images. In blue it can sawn the line and in yellow the code

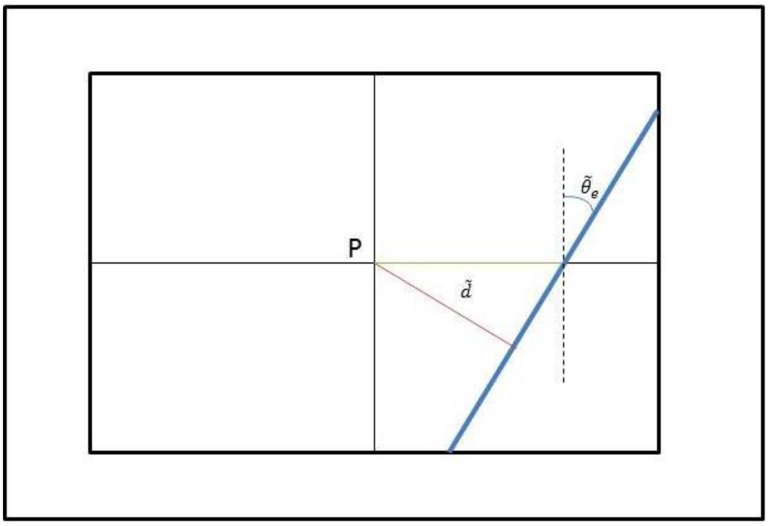


Figure 3.8: Representation of an image captured by the computer vision system

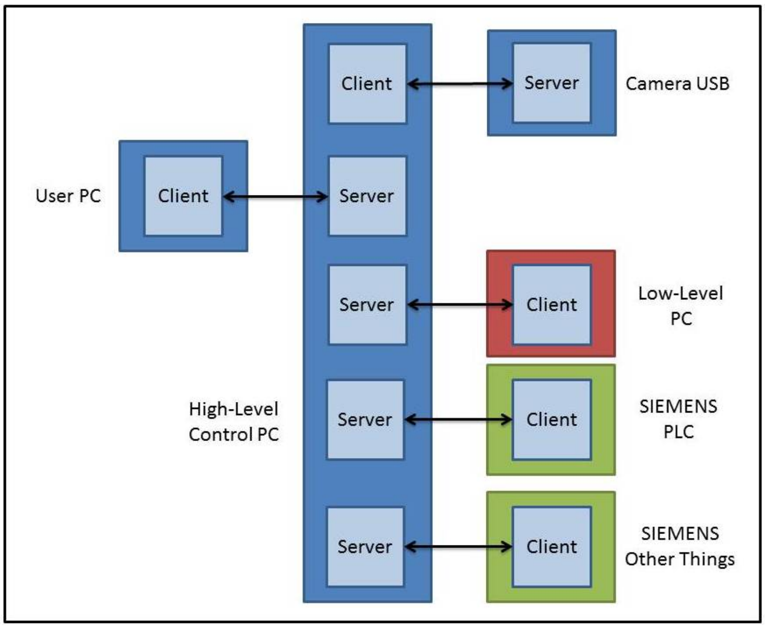


Figure 3.9: High-Level PC communications

that commented in table 3.1 are used.

### Original Control System: a Fuzzy Logic Controller

In this section, the original control system for the path following of the UGS is described. One of the objectives of this Master Work is the replacement of this System by an advanced one (see chapter 4).

The original Control System is composed by three blocks as seen in figure 3.10.

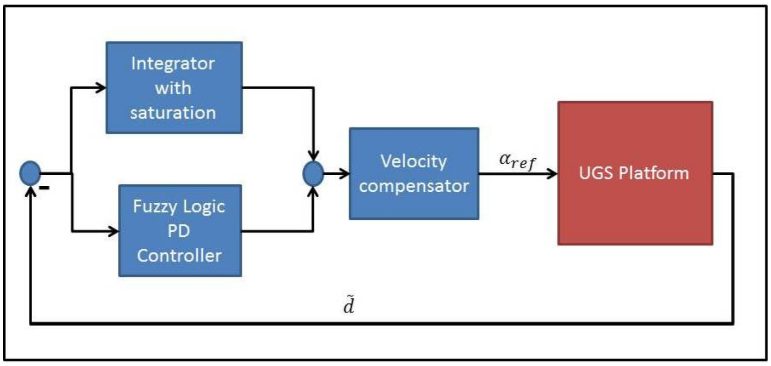


Figure 3.10: Original High-Level Controller

| Name of the variable | Type | Description |
|---|---|---|
| curTime | float | It has the time in seconds at the beginning of the High-Level Control System (in seconds) |
| vel_actual | float | It has the speed of the UGS (in $km/h$) |
| distanceToCircuitMeasuredPix | float | It has the measure of the distance to the circuit (in $pixeles$) |
| cameraLineDetected | bool | It is a boolean. It is true if line is detected, 0 in otherwise |
| angleThetaeMeasured | float | It has the measure of the angle between the UGS and the circuit (in $rad$) |
| cameraNewCodeDetected | bool | It is a boolean. It is true if new code is detected, 0 in otherwise |
| codeReaded | int | It has the information of the code read. It is a number between 0 and 31 |
| cambio | float | It is the output of the High-Level Controller that indicates the steering calculated (in $degreesofthesteering$) |
| cont_distance | float | It is the output of the High-Level Controller that indicates the advance of the UGS over the circuit (in $meters$) |

Table 3.1: Variables of the Very High-Level Controller used in the High-Level Control System

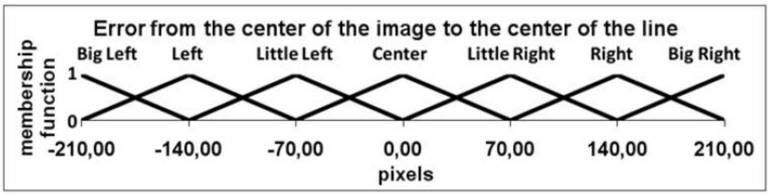


Figure 3.11: Original High-Level Controller

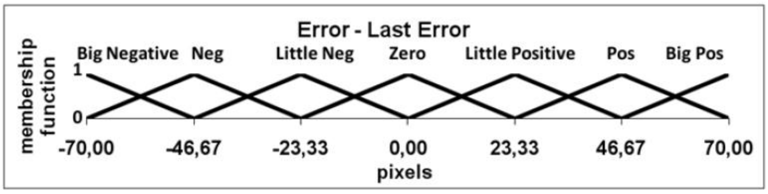


Figure 3.12: Original High-Level Controller

The first one is a Fuzzy controller that acts like a PD controller. The second one is the integral of the error with saturation. The integral is added to the output of the fuzzy controller, making a structure of Fuzzy PD + I.

The Fuzzy controller was implemented using the MOFS (Miguel Olivares' Fuzzy Software). With this software, it is possible to easily define a fuzzy controller with the required number of inputs, the types of membership functions, the defuzzification model, and the inference operator. A more detailed explanation of this software can be found in (I. Mondragon and Mejias, 2010).
The Fuzzy controller has two inputs and one output. All are fuzzyfied using triangular membership functions. The first input is defined as the error between the centre of the image and the centre of the line to follow $\tilde{d}(k)$ (Figure 3.11). The second input is the difference between current and previous error $\tilde{d}(k) - \tilde{d}(k-1)$ (Figure 3.12). The output of the controller is multiplied by a gain ($K_{fuzzy} = 1.2$), getting the absolute turn of the steering to correct this error $\alpha_{ref}$, in degrees (Figure 3.13). To obtain this output, 49 if-then rules were defined (see figure 3.14). The developed fuzzy system is a Mamdani type that use a height weight defuzzification model with the product inference model in Equation 3.1.

$$y = \frac{\sum_{l=1}^{M} \bar{y}^l \prod_{i=1}^{N} (\mu_{x_i^l}(x_i))}{\sum_{l=1}^{M} \prod_{i=1}^{N} (\mu_{x_i^l}(x_i))} \tag{3.1}$$

Where N and M represent the number of inputs variables and total number of rules respectively. $\mu_{x_i^l}$ denote the membership function of the $l$-th rule for the $i$-th input variable. $\bar{y}^l$ represent the output of the $l$-th rule.

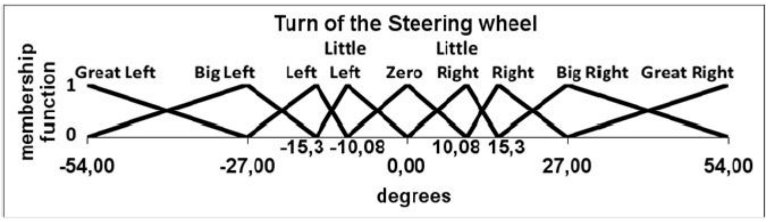
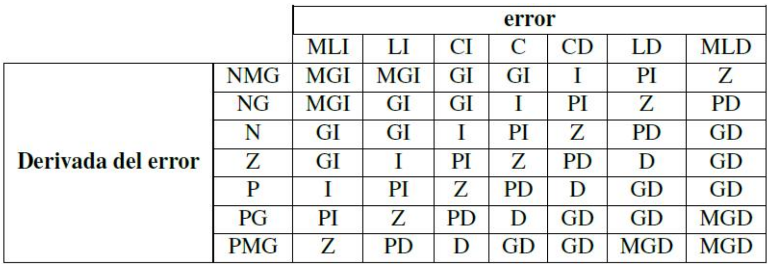


Figure 3.13: Original High-Level Controller



| | | error | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | | MLI | LI | CI | C | CD | LD | MLD |
| **Derivada del error** | NMG | MGI | MGI | GI | GI | I | PI | Z |
| | NG | MGI | GI | GI | I | PI | Z | PD |
| | N | GI | GI | I | PI | Z | PD | GD |
| | Z | GI | I | PI | Z | PD | D | GD |
| | P | I | PI | Z | PD | D | GD | GD |
| | PG | PI | Z | PD | D | GD | GD | MGD |
| | PMG | Z | PD | D | GD | GD | MGD | MGD |

Figure 3.14: Original High-Level Controller

The calculation of the integrator value is shown in Equation 3.2.

$$I_k = I_{k-1} - \tilde{d_k} \cdot (t_k - tk - 1) \cdot K_i \tag{3.2}$$

Where $\tilde{d_k}$ is the current error between the centre of the line and the centre of the image, $t_k - t_{k-1}$ is the elapsed time between two samples, and $K_i$ is a constant that appropriately weights the effect of the integrator, and for this case is equal to 0.5.

Thus the actual velocity of the car is was not included in the Fuzzy controller, but it is taken into account in the third module of the controller, multiplying the Fuzzy PD + I output by $\frac{10}{\tilde{u}_1}$ , being $\tilde{u}_1$ the current velocity of the vehicle (in km/h). The definition of the numerator value of this factor is based on the velocity, in km/h, during a skilled human driving session, in which data was acquired to tune the rule base of the fuzzy controller. It is practically impossible for a human to drive faster than 10 km/h while keeping the line-following error low enough to meet the requirements of the application. This is because the driver only sees 30 cm ahead, and, at that speed, the contents of this area change completely every 0.108 seconds.

The driving session performed by the human at 10 km/h provided the necessary training data to modify the initial base of rules of the controller and the size of the fuzzy sets of its variables. For the definition of the fuzzy sets, a heuristic method was used based on the extraction of statistical measures from the training data. For the initial base of rules, we used a supervised learning algorithm, implemented in MOFS. This algorithm evaluates the situation (value of input variables) and looks for the rules that are involved in it (active rules). Then, according to the steering command given by the human driver, the weights of these

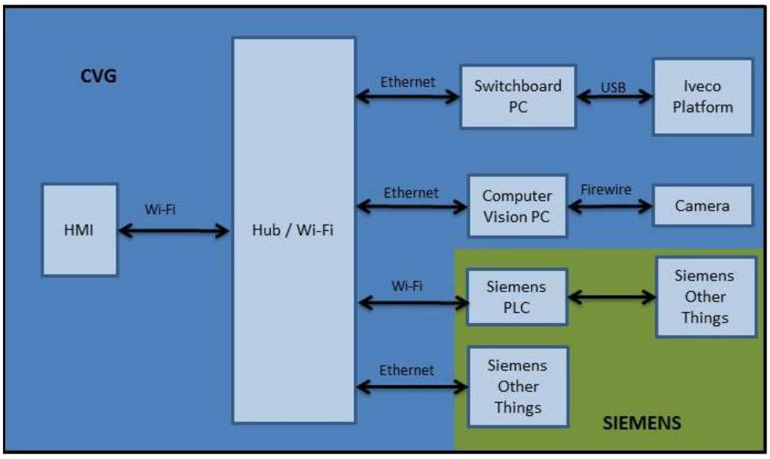


Figure 3.15: Physical Connexions, PCs and main things of the C3 Platform. In dark blue, things developed by CVG; in green, things developed by SIEMENS

rules are changed. Each time that the output of an active rule coincides with the human command, its weight will be increased. Otherwise, when the output differs from the human command, its weight will be decreased by a constant. Anytime the weight of a rule becomes negative the system sets the output of the rule to the one given by the human driver. Further details of the software are given at (I. Mondragon and Mejias, 2010).

### The Magnetic Point

The goal of the magnetic point cannot be described because of the confidentiality of the Project, but the distance to the centre of the rear axis of the car to the magnetic point are: $l_{1-m} = -38cm$ and $l_{2-m} = 0$.

## 3.2. Second Prototype based on a Iveco Eurocargo

The dates of the official development of this prototype were from April 2010 to December 2011 (see section 2.1); but because of economical problems of SIEMENS, S.A., the development of the prototype ended sooner than it must and the prototype never could finished.

In the development of this prototype, the same different partners than in the C3 Prototype had participated, but in different parts. Now INSIA only made the mechanical works and the CVG assumed the Low-Level development of the UGS.

The architecture is also different respect to the C3 Platform as it can be seen in figure 3.15.

Now, in this platform, INSIA only made mechanical works, and they did not develop any control loop or any electronics.

Now the Switchboard PC is in charged of the Low Level Control and also, the High and Very High Level Control. The CV PC only processes the images received by the camera.

### 3.2.1. Vehicle

This prototype uses a Iveco Eurocargo ML150E18 as the vehicle platform.

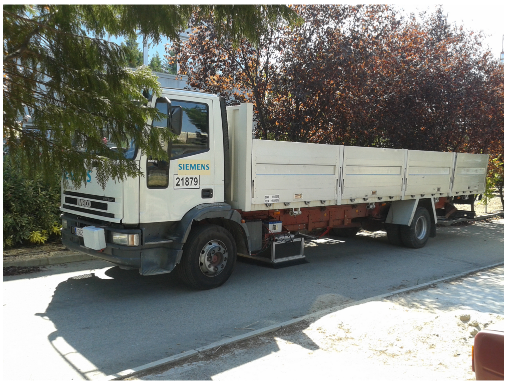


Figure 3.16: Iveco platform front view

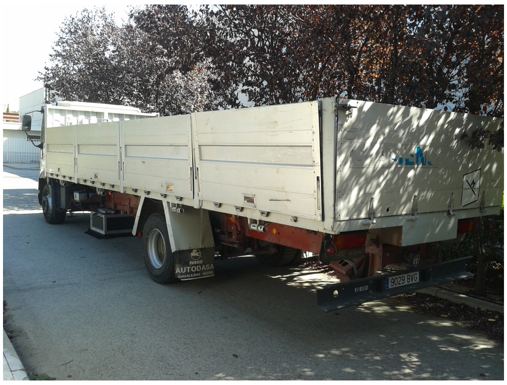


Figure 3.17: Iveco platform rear view

It has the following characteristics relevant for this Master Work:

- Distance between axis: $L = 5670mm$
- Maximun turn of the steering wheels: $\phi_{max} = 35^\circ$
- The wheels are big, and they have big tires (305/70R19.5). They cannot be considered as rigid.

### 3.2.2. Low Level

The Low-Level System is the part of the UGS closer to the vehicle. It was developed by the CVG with the help of the INSIA in the mechanical tasks. The low level structure can be seen in figure 3.18

In this platform, a Switchboard PC with a CAN-BUS interface (IXAAT USB-TO-CAN II) is connected to the vehicle. The vehicle has a encoder with a CAN interface (Bosch LWS1) to measure the steering of the vehicle. Two very Low-Level cards control the motor of the Steering (EPOS2 70/10) and the motor of the brake (EPOS2 24/5). The first one uses a Hall Sensor and a Differential encoder of the motor to close the Low-Level Loop (that uses a PID). The second one only uses a Differential Encoder (that uses a PID). The speed of the vehicle is

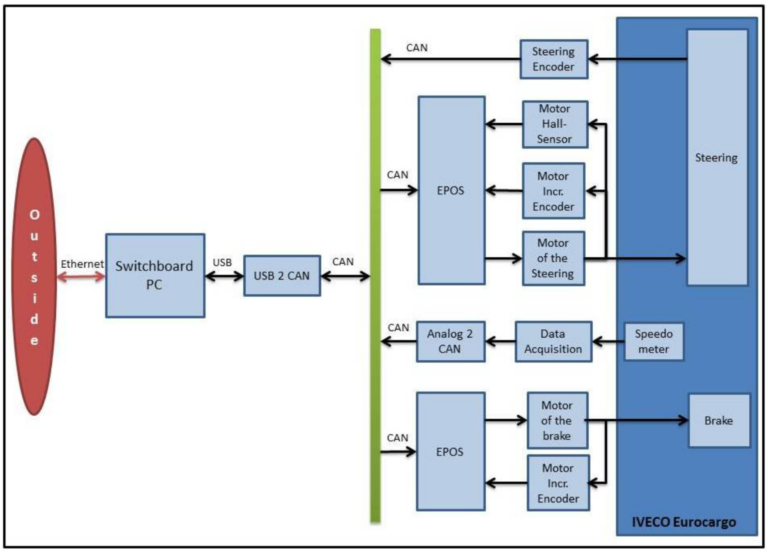


Figure 3.18: Physical Connexions, Sensors, Actuators, PCs and main things of the Low Level Iveco Platform.

acquired thanks to the speedometer, using a data acquisition card and a Analogue to CAN card. The speed of the vehicle is not actuated, just the brake.

The communication between elements uses a CAN-BUS.

As you can see, the Low-Level System of the Iveco is easier than the Low-Level of the C3.

### 3.2.3. High Level

The High-Level of the Iveco is also different from the C3. The High-Level Controller, the Very High-Level Control System and the Low-Level Control System are all running in the same PC, the Switchboard PC.

#### Very High-Level Control System

The very High-Control System of the Iveco is more complex than the C3 but it is not described because it is not necessary for this Master Work.

#### Communication & Software

The communications uses a similar Sockets Process.

Now, the High-Level is developed under Microsoft Windows 7. The very High-Level Control System is programmed in $C\sharp$. The High-Level controller is still programmed in C++ been enclosed in a similar way than in the C3.

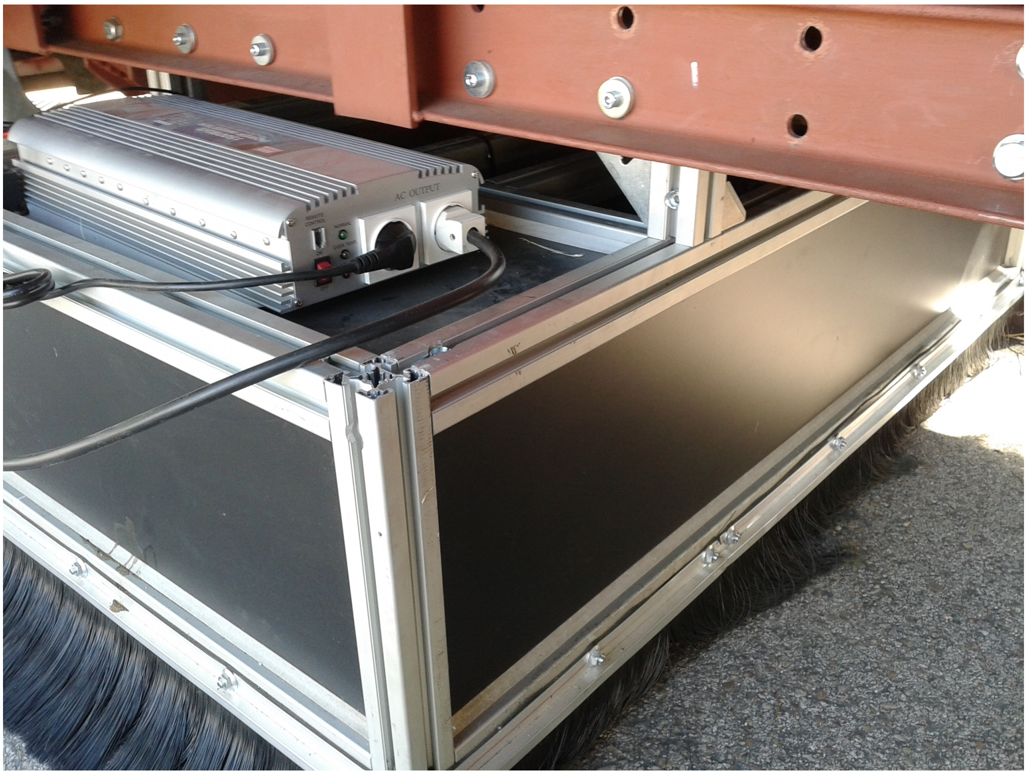


Figure 3.19: Metal structure of the Visual System. Limits the field of view to 133 x 69 cm.

### Computer Vision System

The Computer Vision System of the Iveco is similar to the C3 (see section 3.1.3), but there are some differences.

The first difference is that the CV Algorithm is executed in a different PC (the CV PC).

The camera now is a Firewire Camera instead of a USB Camera.

The structure of the camera and the four ultraviolet bulbs is bigger, been the field of view 133 x 69 cm (see figure **??**.

Now, the distances of the camera from the rear axis of the vehicle are $l_{1-c} = 370cm$ and $l_{2-c} = 0$.

The CV system works at a frequency of $f = 60Hz$.

### The Magnetic Point

For the Iveco Platform, the Magnetic point is located in: $l_{1-m} = 250cm$ and $l_{2-m} = 0$.

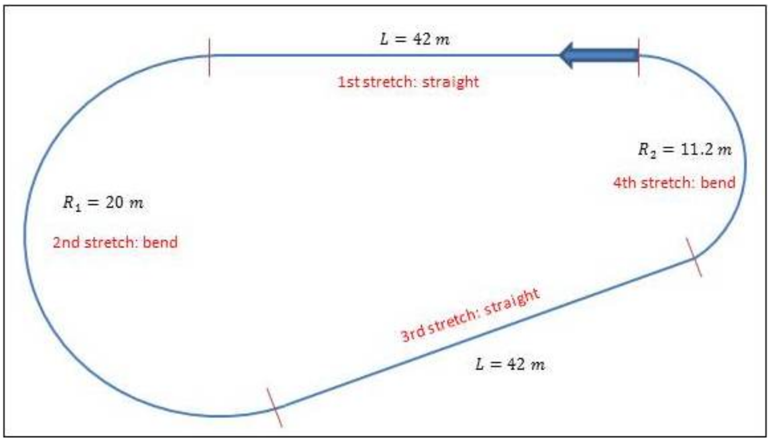


Figure 3.20: Scheme of the Egg Circuit.

## 3.3. Test Circuits

There are three real Test Circuits for the test of the UGS, the Egg Circuit, the O Circuit and the Bus Stop Circuit. All the circuits are a plane surfaces with no irregularities. The O Circuit was never available and only was simulated.

### 3.3.1. The Egg Circuit

This is the most used Circuit in this Master Work. It only has the Paint Circuit and it is composed by two 42 meters straights and two bends, one of a radius of 20 meters and the other 11.2 meters (see figures 3.20 and 3.21). Each stretch is connected in a continuous and derivative way (it means that in every point of the circuit exist the tangent to the circuit). The circuit is always travelled in anti-clockwise.

### 3.3.2. The O Circuit

This circuit never was available, but always was in mind. The known information of this circuit is the magnet path that is composed by 8 stretch (see figure 3.22). First of all, it has a 44 m straight; then a 11.5 m radius quarter of circumference; then a 2 m straight; then other 11.5 m radios quarter of circumference; and then again the previous four stretch.

The paint circuit has not a easy definition because it depends on the position of the Magnet Point and the Camera Point in the Iveco Platform.

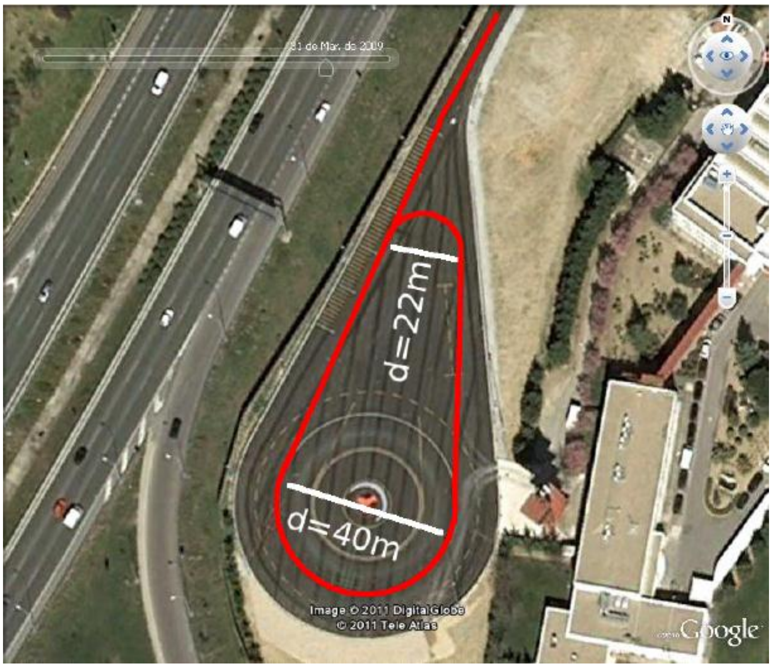


Figure 3.21: Representation of the circuit at INSIA installations took with Google Earth tool.

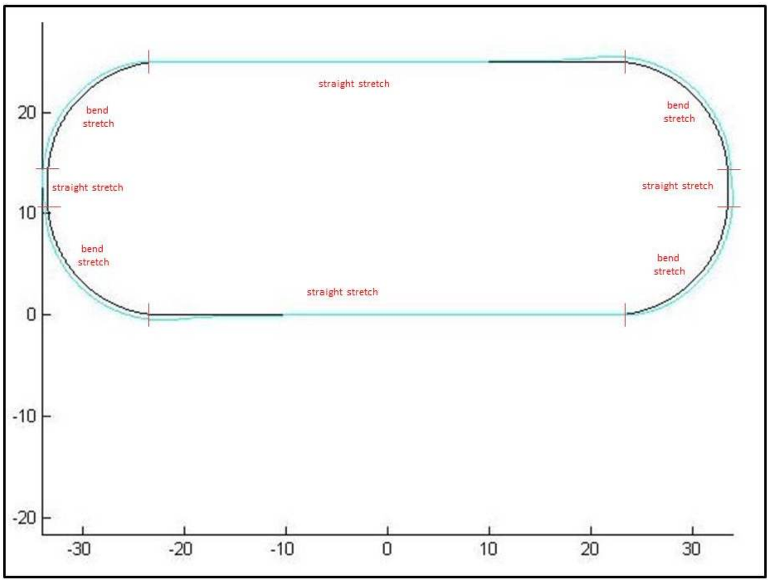


Figure 3.22: Scheme of the O Circuit. . In blue the Paint Circuit, in black the Magnet Circuit

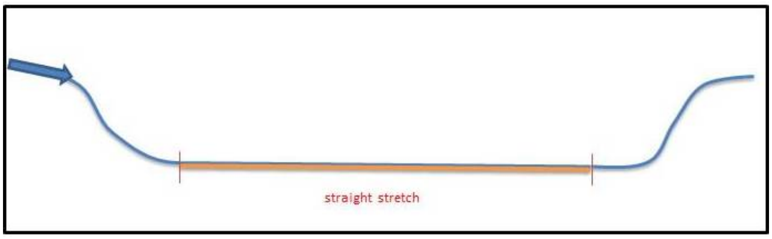


Figure 3.23: Scheme of the Bus Stop Circuit. In blue the Paint Circuit, in orange the Magnet Circuit

### 3.3.3. The Bus Stop Circuit

The Bus Stop Circuit has a unknown Paint Circuit and a straigth stretch of Magnet Circuit. It has a Bus Stop shape with a right bend, then a straight and finally a left bend (see figure 3.23).

# Chapter 4

# The Proposed High-Level Control System

In this chapter is described the proposed High-Level Control System developed as the main objective of this Master Work.

The High-Level Control System is composed by three subsystems in cascade in a closed loop (see figure 4.1): the UGS Platform, the State Estimator and the Controller (this is the order of the cascade connection).

The UGS Platform is described in chapter 3. This system represents the UGS Platform that is controlled. It has two inputs and five outputs (see figure 4.2). The inputs are the speed of reference ($u_{1-ref}$) specified by the user, and the angle of the steering of reference ($\alpha_{ref}$) that is the output of the Controller. The outputs of this system are the measurement of the speedometer ($\tilde{u}_1$), the encoder of the steering ($\tilde{\alpha}$) and the measurements of the camera ($\tilde{d}$, $\tilde{\theta}_e$ and the code detected).

A more detailed scheme can be sawn in figure 4.3.

The following system is the State Estimator (see figure 4.4). It has six inputs and nine outputs. The first five inputs are the measurements of the UGS platform (outputs of the UGS System). The other input is the circuit information previously had (sometimes none, sometimes the magnet circuit and in the rest of the cases, the paint circuit). Some of the outputs are used in the following system,

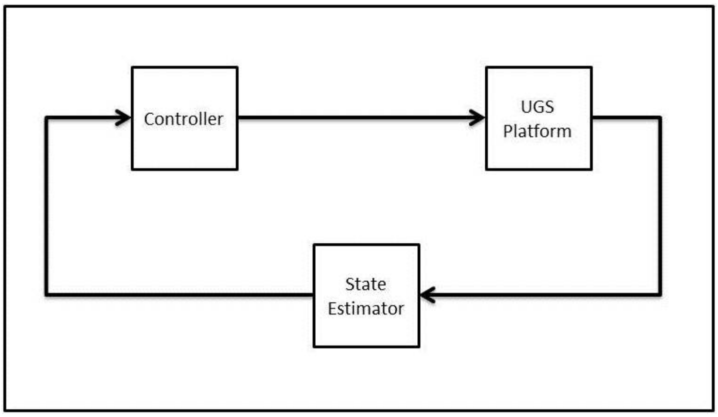


Figure 4.1: High-Level Control System Scheme

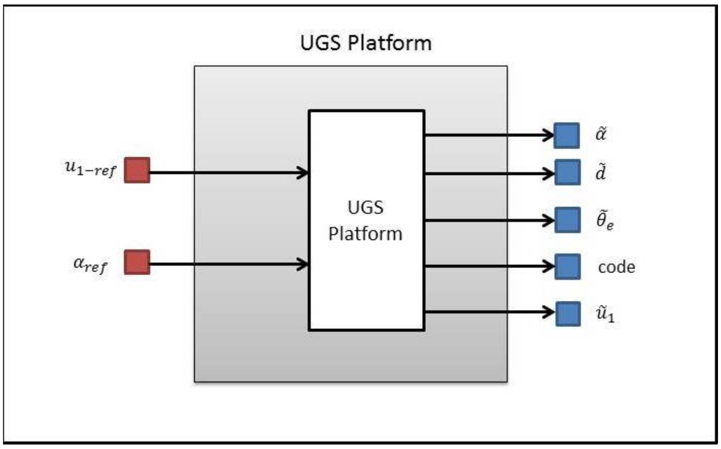


Figure 4.2: UGS Platform System: General

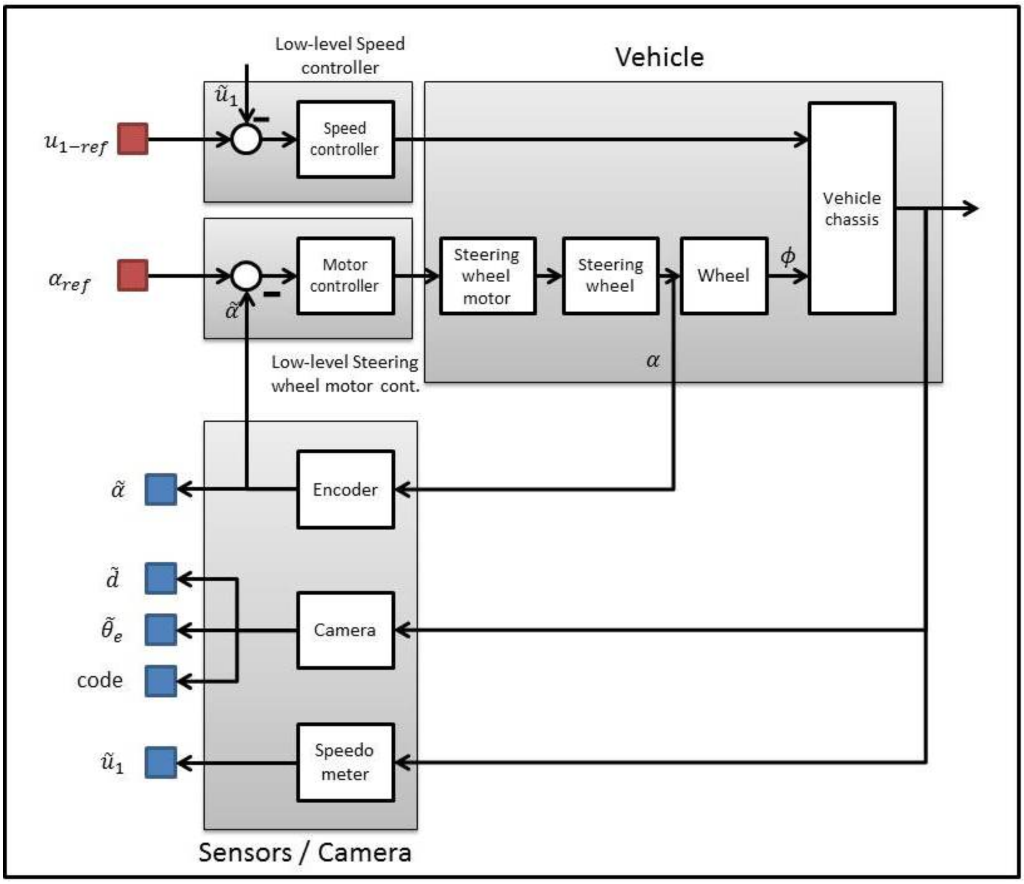


Figure 4.3: UGS Platform System: Detail

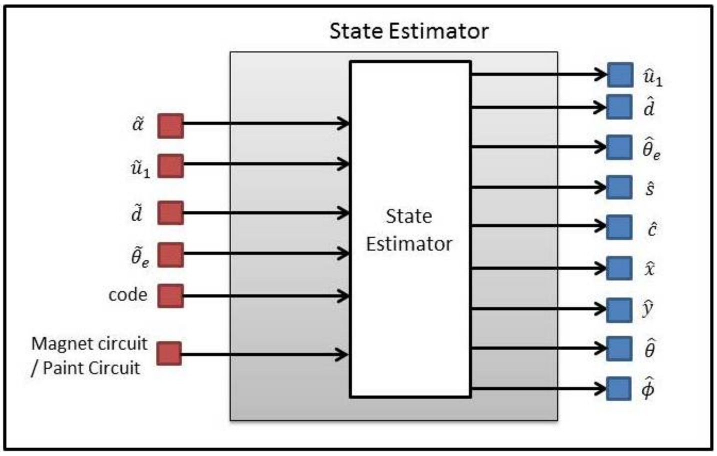


Figure 4.4: State Estimator System: General

the Controller ($\hat{d}$, $\hat{\theta}_e$, $\hat{c}$, $\hat{u}_1$ and $\hat{\phi}$). The other outputs are only for user-monitoring ($\hat{s}$, $\hat{x}$, $\hat{y}$ and $\hat{\theta}$).

The goal of this system is calculate the real state of the UGS Platform, using the information given by the measurements of the platform, some models of the UGS and the available information of the circuit. To achieve this goal, this system is composed by five subsystems (see figure 4.5), explained in chapter 5. These subsystems are: the Wheel position estimator, the Speed Estimator, the Position over circuit Estimator, the Curvature Estimator and the X-Y Position Estimator.

The last system is the Controller (figure 4.6). Its objective is achieve the path following control problem (see appendix A) using the information given by the state Estimator. Its five inputs are some of the outputs of the State Estimator ($\hat{d}$, $\hat{\theta}_e$, $\hat{c}$, $\hat{u}_1$ and $\hat{\phi}$); and it output is the control reference of the UGS platform (the non-user input of the UGS platform System).

This system is decomposed in three subsystem described in chapter 6, and they are the Feed-Back Controller, the Feed-Forward Controller and the Wheel Dynamics Compensator (figure 4.7).

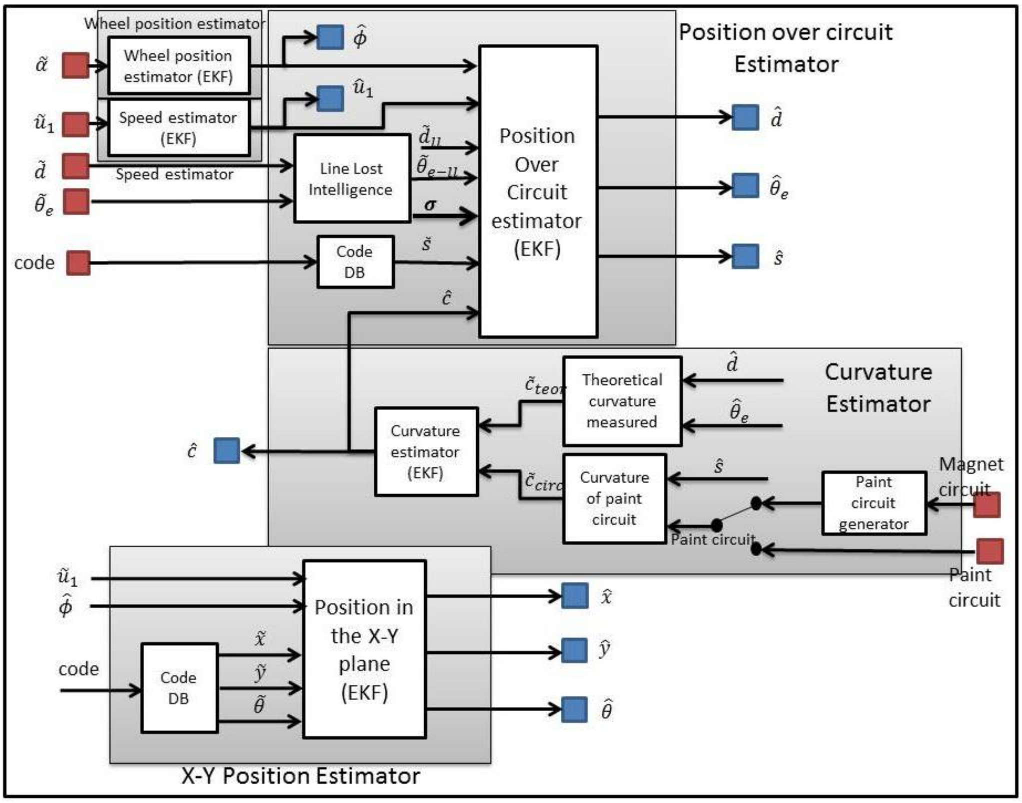


Figure 4.5: State Estimator System: Detail

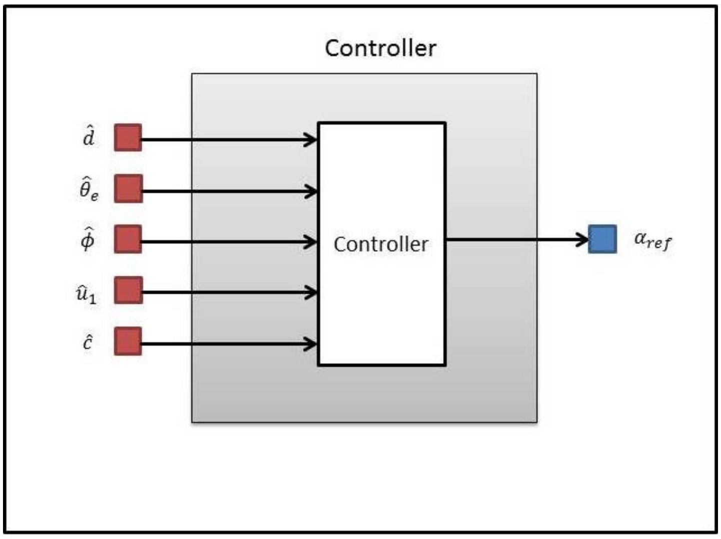


Figure 4.6: Controller: General

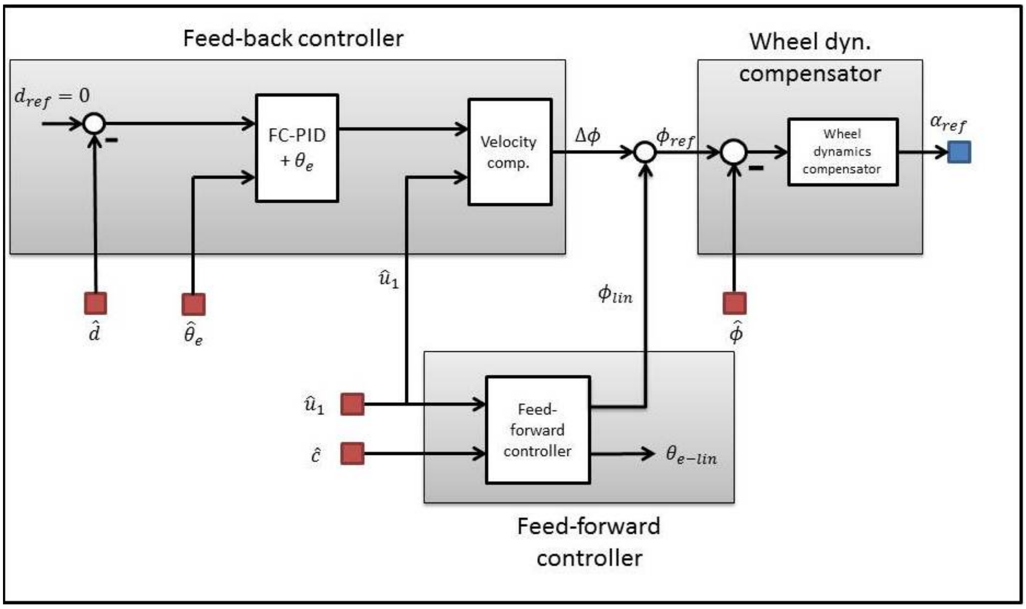


Figure 4.7: Controller: Detail

# Chapter 5

# State Estimation

In this chapter, the State Estimator System presented in chapter 4 is described.

The goal of this system is calculate the real state of the UGS Platform, using the information given by the measurements of the platform, some models of the UGS and the available information of the circuit.

This system has six inputs and nine outputs. The first five inputs are the measurements of the UGS platform: the measurement of the speedometer ($\tilde{u}_1$); the measurement of the encoder of the steering ($\tilde{\alpha}$); and the measurements of the camera (minimum distance between the camera to the paint circuit $\tilde{d}$, angle between the paint circuit and the advance axis of the UGS over the paint circuit $\tilde{\theta}_e$, and the code read for the camera that has some information about the position X-Y-$\theta$ of the code and the advance over the circuit ($s$) in this code). The other input is the available circuit information had (sometimes none, sometimes the magnet circuit and in the rest of the cases, the paint circuit). Some of the outputs are used in the Controller (estimated distance between the camera to the paint circuit $\hat{d}$; estimated angle between the paint circuit and the advance axis of the UGS over the paint circuit $\hat{\theta}_e$; estimated curvature of the paint circuit $\hat{c}$; estimated speed of the UGS platform $\hat{u}_1$; and estimated angle of the steering wheel respect to the advance axis of the vehicle $\hat{\phi}$;. The other outputs are only used for user-monitoring: estimated advance of the vehicle over the circuit $\hat{s}$; and estimated $x$, $y$ and $\theta$ position of the vehicle on the plane ($\hat{x}$, $\hat{y}$ and $\hat{\theta}$).

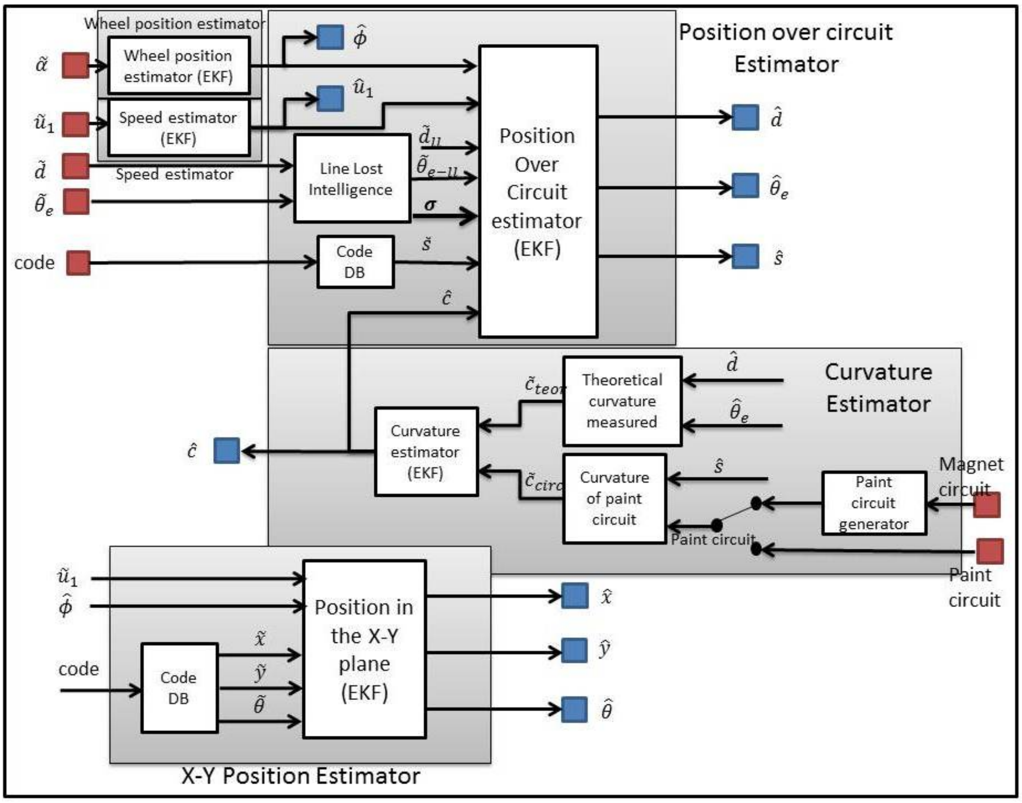


Figure 5.1: State Estimator System: Detail

This system is composed of five subsystems, described in the following sections (see figure 5.1):

- Wheel Position Estimator: section 5.1
- Speed Estimator: section 5.2
- Position over Circuit Estimator: section 5.3
- Curvature Estimator: section 5.4
- X-Y Position Estimator: section 5.5

This State Estimator is tested through some simulations in section 7.1.

## 5.1. Wheel Position Estimator

This subsystem is in charged of the estimation of the angle of the steering wheels of the vehicle.

As the only measurable variable is the angle of the steering using a encoder, this subsystem is necessary to estimate the angle of the steering wheels using the measurements of the steering and other inputs.

For the **Citroën C3 platform**, this estimator is really easy because the steering wheels have a small tire zone and are overinflated. Thanks to that, it can be assumed that the set steering-steering wheel is totally rigid. Then this subsystem is as easy as the following equation, that consider the different angle turned between the wheels and the steering wheel:

$$\tilde{\phi} = 18.0 \cdot \tilde{\alpha} \tag{5.1}$$

A filter of this measurement is not necessary because it is not too much noisy, therefore we have:

$$\hat{\phi} = \tilde{\phi} = 18.0 \cdot \tilde{\alpha} \tag{5.2}$$

Note that a Extended Kalman Filter (EKF) (see appendix B) can be used, but is easier not using it.

For the **Iveco platform**, this estimator is more difficult than before because the wheels have big tires that have a lot of hysteresis and they cannot be considered as a rigid set. After some empiric test, we establish that is necessary to use a function of the measurements of the encoder of the steering and of the velocity of the vehicle:

$$\tilde{\phi} = f(\tilde{\alpha}, \hat{u}_1) \tag{5.3}$$

To determine this function, some test were programmed, but because of the precipitated ending of the project, they never could be done. A 1-D Laser was going to be placed in the steering wheel as is shown in the figure 5.2, been:

$$\tilde{\phi} = \arctan \frac{x(\phi) - x(0)}{x_{laser}} \tag{5.4}$$

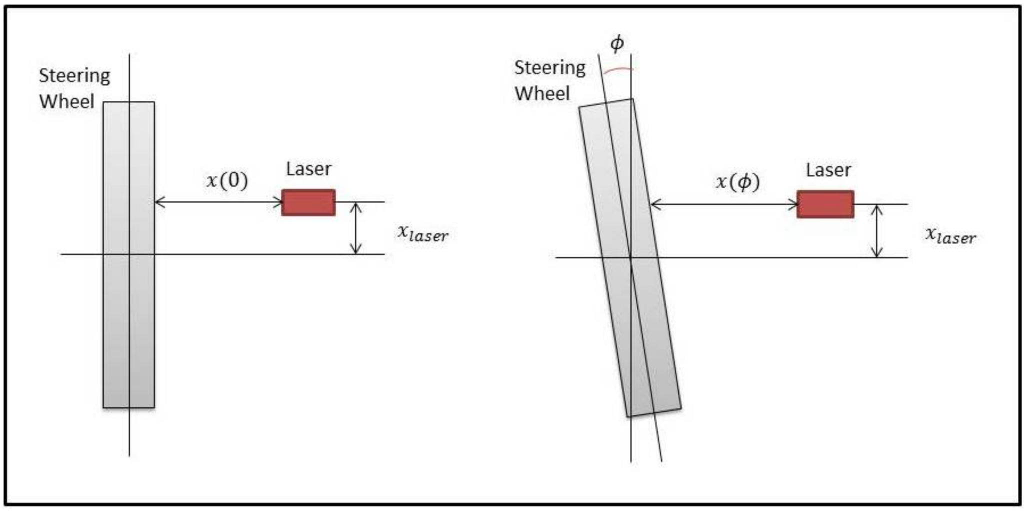


Figure 5.2: Steering Wheel Estimator

The test would consist in move the UGS platform at different speeds, moving the steering and registering the array [$\hat{u_1}$, $\tilde{\alpha}$, $\tilde{\phi}$]. Then, a 2-D function or a State Space Model with two inputs and one output could be identified. Also a Neural Network (NN) (for example a Multi-Layer Perceptron (MLP) for generalization) could be trained with these data to estimate the angle of the steering wheel ($\tilde{\phi}$). Finally, a data filter could be used if it was necessary. Note that a EKF could be used if a State Space Model is identified.

## 5.2. Speed Estimator

The goal of this subsystem is calculate the real speed of the UGS Platform. As the measurement of the speedometer is really good because it has precision enough and it is not noisy, we can affirm that:

$$\hat{u}_1 = \tilde{u}_1 \tag{5.5}$$

## 5.3. Position over Circuit Estimator

The position over Circuit Estimator is a key subsystem in the State Estimator System. Its goal is estimate the position of the UGS platform over the path that it is following.

The outputs of this system are the distance of the camera of the UGS to the path ($\hat{d}$), the angle of the advance axis of the vehicle respect to the path ($\hat{\theta}_e$) and the advance of the UGS over the path ($\hat{s}$). The two first outputs are used in the Controller (see chapter 6). The inputs of this subsystem are the estimation of the inputs of the UGS platform (speed $\hat{u}_1$ and steering wheel angle $\hat{\phi}$); the measurements of the camera (measured distance and angle between the UGS and the path $\tilde{d}$ and $\tilde{\theta}_e$, and the detected code); and the estimated curvature of the circuit (path) $\hat{c}$.

The position over circuit estimator is composed by three blocks. One of them is the code DB, that translates the code detected by the camera in a distance over the circuit information ($\tilde{s}$). Other one is a EKF that is described in 5.3.1. The last one is the Line Lost Intelligence that changes the measures of the distance to the path ($\tilde{d}$) and the angle between the UGS and the path ($\tilde{\theta}_e$) and the variances of the EKF depending of if the line is lost and the time spent since the line loss (see 5.3.2).

### 5.3.1. Position Over Circuit EKF

An EKF (see Appendix B) is used for estimate the position over the circuit. An asynchronous EKF is needed because the inputs of the process ($\hat{u}_1$, $\hat{\phi}$ and $\hat{c}$) and all the outputs of the process measured ($\tilde{d}$, $\tilde{\theta}_e$ and $\tilde{s}$) are not received at the same time.

The State Space Model of the process used is the Frenet-frame kinematic model of a car like mobile robot (see (Siciliano, 2008) and figure 5.3) that has three states that are the posture of the UGV over the path ( $\vec{x} = \begin{bmatrix} s & d & \theta_e \end{bmatrix}^t$):

$$\begin{aligned}
\dot{s} &= \frac{u_1}{1 - d \cdot c(s)} \cdot [\cos\theta_e - \frac{\tan\phi}{L} \cdot (l_2 \cdot \cos\theta_e + l_1 \cdot \sin\theta_e)] && (5.6) \\
\dot{d} &= u_1 \cdot [\sin\theta_e + \frac{\tan\phi}{L} \cdot (l_1 \cdot \cos\theta_e - l_2 \cdot \sin\theta_e)] && (5.7) \\
\dot{\theta}_e &= \frac{u_1}{L} \cdot \tan\phi - \dot{s} \cdot c(s) && (5.8)
\end{aligned}$$

The Output model of the process is linear:

$$\begin{bmatrix} d \\ \theta_e \\ s \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{bmatrix} \cdot \begin{bmatrix} s \\ d \\ \theta_e \end{bmatrix} \tag{5.9}$$

The initial state of the EKF is set up by the user, depending on where the UGV is respect to the path that it has to follow. The matrices of covariances of the EKF are calculated empirically depending of the UGS Platform.

For the **Citroën C3 platform** the matrices of covariances if the speed of the UGV is 0 m/s are following:
Matrix of covariances of the inputs of the process Q:

$$Q = \begin{bmatrix} (\frac{0.0}{3})^2 & 0 & 0 \\ 0 & (\frac{0.2}{3})^2 & 0 \\ 0 & 0 & (\frac{1.0}{3})^2 \end{bmatrix} \tag{5.10}$$

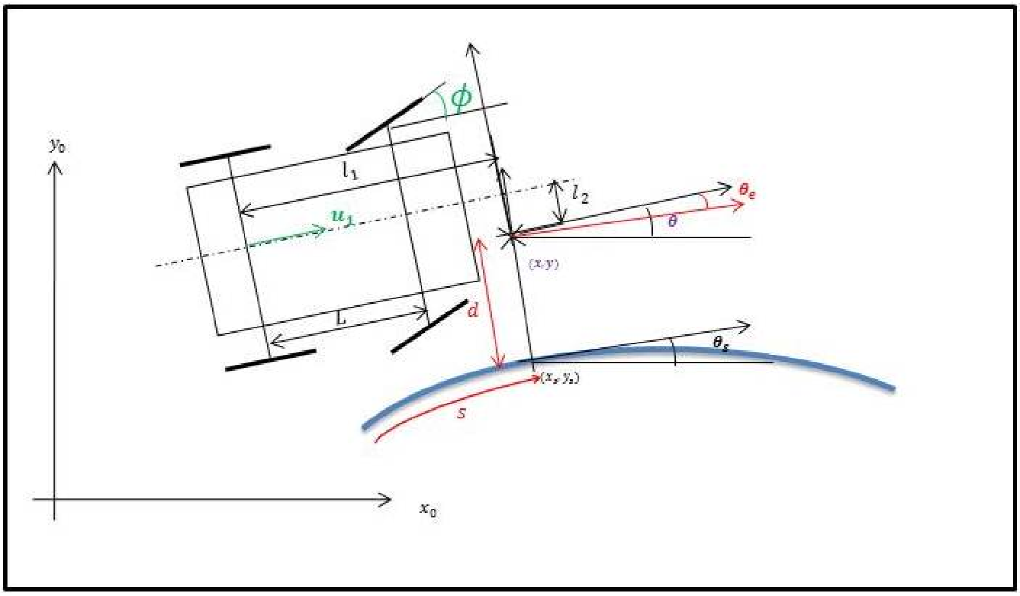


Figure 5.3: Scheme of the UGS Platform on a Fenet-Frame

Matrix of covariances of the state M:

$$M = \begin{bmatrix} (\frac{0.02}{3})^2 & 0 & 0 \\ 0 & (\frac{0.05}{3})^2 & 0 \\ 0 & 0 & (\frac{5.0\cdot\frac{\pi}{180}}{3})^2 \end{bmatrix} \tag{5.11}$$

Matrix of covariances of the initial state P:

$$P(0|0) = \begin{bmatrix} (\frac{1.0}{3})^2 & 0 & 0 \\ 0 & (\frac{0.2}{3})^2 & 0 \\ 0 & 0 & (\frac{15.0\cdot\frac{\pi}{180}}{3})^2 \end{bmatrix} \tag{5.12}$$

The matrix of covariances of the outputs of the process is not static. It changes with the speed of the UGV because of, while the speed of the vehicle is higher, less data is available for a determined distance, therefore, the avaliable measurements are more ”valuable”. To take this into account, we change the value of the variances of the matrix of the outputs (matrix R) depending on the velocity, making them smaller depending on the velocity of the UGV, using the following empirical equation:

$$\sigma_{i-v} = \sigma_{i-0} \cdot e^{\tau_v \cdot \hat{u}_1} \tag{5.13}$$

Where $\tau_v = -0.2$ and the $\sigma_{i-0}$ values are defined in table 5.1:

Resulting the matrix of covariances of the outputs of the process:

$$R = \begin{bmatrix} \sigma_{1-v} & 0 & 0 \\ 0 & \sigma_{2-v} & 0 \\ 0 & 0 & \sigma_{3-v} \end{bmatrix} \tag{5.14}$$

| | |
|---|---|
| $\sigma_{1-0}$ | $(\frac{0.02}{3})^2$ |
| $\sigma_{2-0}$ | $(\frac{0.02}{3})^2$ |
| $\sigma_{3-0}$ | $(\frac{0.02}{3})^2$ |

Table 5.1: Values of the initial variances of R for the Position Over Circuit EKF

For the **Iveco Platform**, these matrices never were calculated.

### 5.3.2. Line Lost Behaviour

If the paint circuit is lost in any sample, it is very important that the system knows that and acts in a different way acting in a safety way. If the line is lost, the last measurements of the camera ($\tilde{d}$ and $\tilde{\theta}_e$) are used but the variances of these outputs of the process are changed in order to consider them less "reliable" measurements by making this variances bigger with the number of missed samples ($n_{no-line}$), following the next equation:

$$\sigma_{i-ll} = \sigma_{i-v} \cdot e^{\tau_{ll} \cdot n_{no-line}} \tag{5.15}$$

Where $\tau_{ll} = 0.18$ is the value for the **Citroën C3 Platform** and $n_{no-line}$ is the number of consecutive missed samples. The coefficient for the **Iveco Platform** never was calculated.

## 5.4. Curvature Estimator

The Curvature Estimator is in charged of calculate the curvature of the paint circuit ($\hat{c}$). For this task, it uses the estimated position of the UGV over the circuit ($\hat{d}$, $\hat{\theta}_e$ and $\hat{s}$) and the previous circuit information available that can be none, the magnet circuit or the paint circuit.

It is composed of four blocks with different tasks. The "Paint Circuit Generator" has the goal of calculate the curvature of the Paint circuit depending on the advance over the circuit, if the Magnet Circuit is known (see section 5.4.1). The "Curvature of paint Circuit Extractor" is a easy module that extract the curvature of the paint circuit in each estimated advance of the UGV over the circuit ($\hat{s}$), getting $\tilde{c}_{circ}$. The "Theoretical Curvature Measured Calculator" has the goal of calculate a value of the curvature of the circuit ($\tilde{c}_{teor}$) using a theoretical model (see section 5.4.2). Finally, the "Curvature Estimator EKF" fuses both curvature data.

### 5.4.1. Paint Circuit Generator

The objective of this module is calculate the curvature of the paint circuit depending of the advance of the UGV over the circuit, if the magnet circuit is known. For this task, an offline Frenet-frame kinematic simulation of the UGV is done. The Frenet-Frame kinematic model of a car-like mobile robot is used (see (Siciliano, 2008) and figure 5.3). There are followed these steps:

1. First simulation for the calculation of the steering angle needed for the path following of the magnet circuit. It is calculated as is done in (J. L. Sanchez-Lopez, 2011):

   $$\phi = \arctan\left(-\frac{L \cdot \sin\theta_e}{l_{m-1} \cdot \cos\theta_e - l_{m-2} \cdot \sin\theta_e}\right) \tag{5.16}$$

   Where $l_{m-1}$ and $l_{m-2}$ is the position of the magnet point of interest in the UGV.

2. Second simulation for the calculation of the curvature of the paint circuit, using the steering angle. The curvature is calculating using the equations of the Frenet-frame kinematic model of a mobile robot and solving the value of $c$ but now, $l_1$ and $l_2$ is the position of the camera in the UGV.

The output of this module is a table that contains the curvature of the paint circuit depending of the advance over this circuit $c = c(s)$.

### 5.4.2. Theoretical Curvature Measured Calculator

To calculate a theoretical curvature, we can use the Frenet-frame kinematic model [(Siciliano, 2008)], including equation 5.6 in equation 5.8, getting:

$$\begin{aligned} \dot{d} &= u_1 \cdot [\sin\theta_e + \frac{\tan\phi}{L} \cdot (l_1 \cdot \cos\theta_e - l_2 \cdot \sin\theta_e)] && (5.17) \\ \dot{\theta}_e &= \frac{u_1}{L} \cdot \tan\phi - \\ &\quad - \frac{u_1 \cdot c(s)}{1 - d \cdot c(s)} \cdot [\cos\theta_e - \frac{\tan\phi}{L} \cdot (l_2 \cdot \cos\theta_e + l_1 \cdot \sin\theta_e)] && (5.18) \end{aligned}$$

If we calculate the linearising point of the model by making $d = 0$, $\dot{d} = 0$, $\dot{\theta}_e = 0$, we get:

$$c = \frac{\sin\theta_e}{\cos\theta \cdot (l_2 \cdot \sin\theta_e - l_1 \cdot \cos\theta_e) - \sin\theta \cdot (l_2 \cdot \cos\theta_e + l_1 \cdot \sin\theta_e)} \tag{5.19}$$

And if we make the hypothesis that the UGV is working in the linearising point we can calculate a theoretical curvature (($\tilde{c}_{teor}$) with the previous equation.

### 5.4.3. Curvature Estimator EKF

To fuse the two available circuit curvatures, we can use an EKF with two states ($x = \begin{bmatrix} x_1 & x_2 \end{bmatrix}^t$), one input of the process (increment of the advanced distance of the UGV over the circuit $\Delta s$) and two outputs (the two curvatures commented before). The state space model of the process can be a first derivative change model like:

$$x_1(k+1) = x_1(k) \tag{5.20}$$

$$x_2(k+1) = x_2(k) + x_1(k) \cdot \Delta s \tag{5.21}$$

And the output model is linear:

$$\begin{bmatrix} c \\ c \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \tag{5.22}$$

The initial state of this EKF is set to zero. The covariance matrices were adjusted empirically to the **Citroën C3 Platform**:
Matrix of covariances of the outputs of the process:

$$R = \begin{bmatrix} (\frac{400.0}{3})^2 & 0 \\ 0 & (\frac{0.0}{3})^2 \end{bmatrix} \tag{5.23}$$

Matrix of covariances of the inputs of the process:

$$Q = \begin{bmatrix} (\frac{0.05}{3})^2 \end{bmatrix} \tag{5.24}$$

Matrix of covariances of the state P:

$$M = \begin{bmatrix} (\frac{0.05}{3})^2 & 0 \\ 0 & (\frac{0.05}{3})^2 \end{bmatrix} \tag{5.25}$$

Matrix of covariances of the initial state P:

$$P(0|0) = \begin{bmatrix} (\frac{0.5}{3})^2 & 0 \\ 0 & (\frac{0.5}{3})^2 \end{bmatrix} \tag{5.26}$$

For the **Iveco Platform**, they never were calculated.

## 5.5. X-Y Position Estimator

The goal of this subsystem is the estimation of the position of the point of interest of the UGS (point of the camera) on the X-Y surface, getting the value of x, y and $\theta$ respect to a reference system of the world (see figure 5.4). In this Master Work, it is used only for monitoring, but as it is shown in chapter 9, it could be used for example, to control the UGS with the help of a GPS.

An asynchronous EKF (see appendix B) is used for this system. The inputs of the process are two: the estimated speed of the UGS ($\hat{u}_1$) and the estimated angle of the steering wheel of the UGS ($\hat{\phi}$). The outputs of the process are three and they are the posture of the UGS in the X-Y-$\theta$ space ($\tilde{x}$, $\tilde{y}$ and $\tilde{\theta}$). The only available measures of the posture of the UGS are those when a code is read

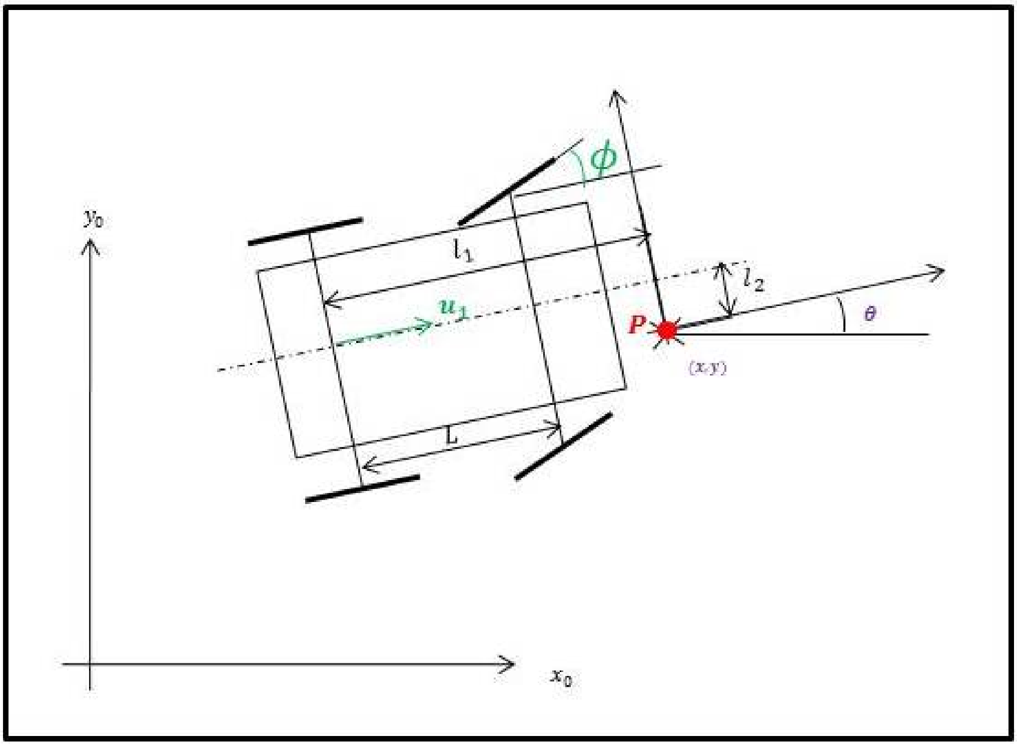


Figure 5.4: Scheme of the UGS Platform on the X-Y surface

because of the position of the codes is known. We can use as outputs of the process:

$$\tilde{x} = x_{code} \tag{5.27}$$

$$\tilde{y} = y_{code} \tag{5.28}$$

$$\tilde{\theta} = \theta_{code} + \hat{\theta}_e \tag{5.29}$$

Because of the inputs of the process are not synchronized with the outputs of the process, an asynchronous EKF is needed, activating the outputs only when the measurements are available.

The state space model of the process used is the kinematic model of a mobile robot moving over the X-Y surface (see (Siciliano, 2008)) that has three states, the posture of the UGS ( $\vec{x} = \begin{bmatrix} x & y & \theta \end{bmatrix}^t$):

$$\dot{x} = u_1 \cdot [\cos\theta - \frac{\tan\phi}{L} \cdot (l_2 \cdot \cos\theta + l_1 \cdot \sin\theta)] \tag{5.30}$$

$$\dot{y} = u_1 \cdot [\sin\theta + \frac{\tan\phi}{L} \cdot (l_1 \cdot \cos\theta - l_2 \cdot \sin\theta)] \tag{5.31}$$

$$\dot{\theta} = \frac{u_1}{L} \cdot \tan\phi \tag{5.32}$$

The output model of the process is linear:

$$\begin{bmatrix} x \\ y \\ \theta \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ \theta \end{bmatrix} \tag{5.33}$$

The initial state of the UGS is set up for the user, depending on where the UGS begins the race. The matrices of covariances of the EKF are calculated empirically and depending on the UGS Platform:

For the **Citroën C3 platform** are the following:
Matrix of covariances of the outputs of the process:

$$R = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & (\frac{0.9}{3})^2 \end{bmatrix} \tag{5.34}$$

Matrix of covariances of the inputs of the process:

$$Q = \begin{bmatrix} (\frac{0.0}{3})^2 & 0 \\ 0 & (\frac{0.2}{3})^2 \end{bmatrix} \tag{5.35}$$

Matrix of covariances of the state P:

$$M = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \tag{5.36}$$

Matrix of covariances of the initial state P:

$$P(0|0) = \begin{bmatrix} (\frac{0.3}{3})^2 & 0 & 0 \\ 0 & (\frac{0.15}{3})^2 & 0 \\ 0 & 0 & (\frac{15.0 \cdot \frac{\pi}{180}}{3})^2 \end{bmatrix} \tag{5.37}$$

For the **Iveco Platform**, this matrices never could be calculated.

# Chapter 6

# The Controller

In this chapter, the Controller presented in chapter 4 is described. The goal of this system is, using the information provided by the State Estimator (chapter 5), generate the steering reference to make the UGS follow the Paint Circuit.

This system has five inputs: the estimated distance of the UGS to the Paint Circuit $\hat{d}$, the estimated angle between the UGS and the Paint Circuit $\hat{\theta}_e$, the estimated angle of the steering wheel $\hat{\phi}$, the estimated speed of the UGS $\hat{u}_1$ and the estimated curvature of the path that it is following $\hat{c}$. The only output is the steering reference for the path following $\alpha_{ref}$.

The Controller is composed of three subsystems (figure 6.1): the wheel dynamics compensator (section 6.1), the Feed-forward controller (section 6.2) and the Feedback controller (section 6.3).

The control loop is closed at the frequency of the CV system, that is the lower sensor.

This system is tested through a set of simulation in section 7.2.

## 6.1. Wheel Dynamics Compensator

The Wheel Dynamics Compensator has the goal of minimize the effect of the tire deformation. It generates the reference of the steering of the UGV using the error $\phi_{error} = \phi_{ref} - \hat{\phi}$. Any kind of controller can be used here to compensate

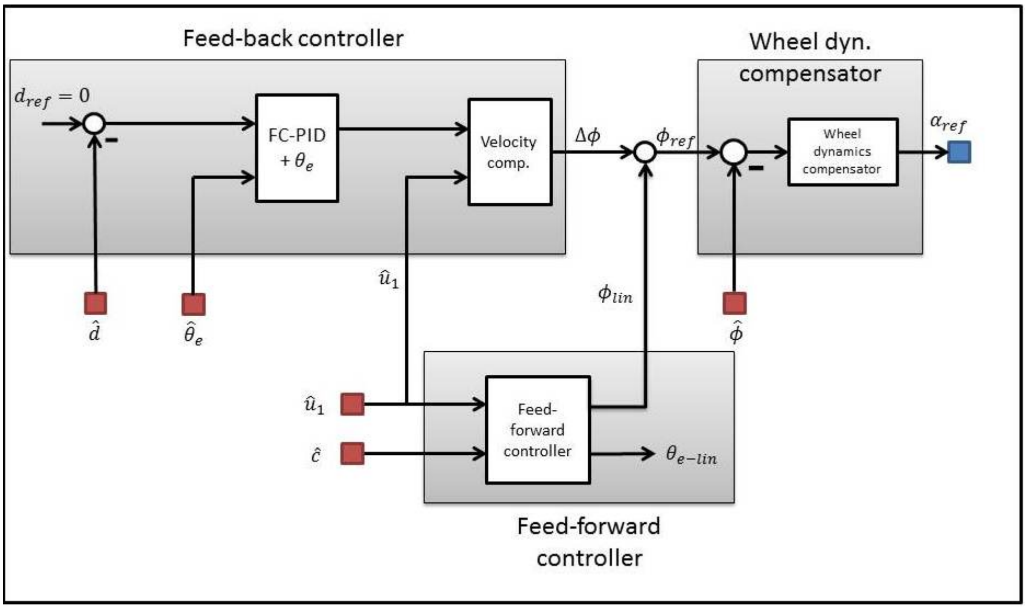


Figure 6.1: Controller: Detail

this effect: PID, adaptive PID, Fuzzy-Logic Controller, Neural Networks,... The system dynamics is modelled in the module Wheel Position Estimator (section 5.1).

For the **Citroën C3 Platform**, we said in section 5.1 that the steering wheel can be considered as a non-deformable body and therefore its dynamics is as easy as: $\hat{\phi} = \tilde{\phi} = 18.0 \cdot \tilde{\alpha}$, therefore, it is not necessary to close the loop and only an open-loop transformation is needed:

$$\alpha_{ref} = 18 \cdot \phi_{ref} \tag{6.1}$$

Note that this block is only to compensate the tire deformation effect and not for the control of the turn of the steering that is closed loop controlled in the low-level.

For the **Iveco Platform**, as the dynamic of the tire deformation never was modelled, this controller never was calculated.

## 6.2. Feed-Forward Controller

The goal of this block is the change if the operating point of the Feedback controller to improve the performance of it. It is done by adding to the output of the feedback controller $\Delta\phi$, an offset $\phi_{lin}$ calculated using the estimated speed of the UGS $\hat{u}_1$ and the estimated curvature of the circuit $\hat{c}$. With this module the UGS can go ahead to the changes of the circuit (included in the curvature information).

For this purpose, the Frenet Frame kinematic model of a car-like mobile robot (see (Siciliano, 2008) and chapter 5) is used. The operating point is calculated by doing $d = 0$ and $\dot{d} = 0$ (see (J. L. Sanchez-Lopez, 2011)), getting:

$$\begin{aligned} \phi &= \arctan\left(-\frac{L \cdot \sin\theta_e}{l_1 \cdot \cos\theta_e - l_2 \cdot \sin\theta_e}\right) && (6.2) \\ \dot{\theta}_e &= u_1 \cdot \frac{\sin\theta_e}{l_2 \cdot \sin\theta_e - l_1 \cdot \cos\theta_e} - \\ & -u_1 \cdot c \cdot \left[\cos\theta_e - \frac{\sin\theta_e \cdot (l_2 \cdot \cos\theta_e - l_1 \cdot \sin\theta_e)}{l_2 \cdot \sin\theta_e - l_1 \cdot \cos\theta_e}\right] && (6.3) \end{aligned}$$

To calculate this operating point, an EKF (see appendix B) is used, but only as a process simulator. The state space model of the process used is described in equation 6.3, been the state $x = \theta_e$. The output model of the process is described in equation 6.2. The state is added as output to the model with the only propose of monitoring. The inputs of this EKF are the estimated curvature of the circuit $\hat{c}$ and the estimated speed of the vehicle $\hat{u}_1$.

For the **Citroën C3 Platform**, we had:
Matrix of covariances of the outputs of the process. It is not needed because of, the state is not corrected. Matrix of covariances of the inputs of the process:

$$Q = \begin{bmatrix} (\frac{2.0}{3})^2 & 0 \\ 0 & (\frac{0.05}{3})^2 \end{bmatrix} \tag{6.4}$$

Matrix of covariances of the state P:

$$M = \left[(\tfrac{0.5}{3})^2\right] \tag{6.5}$$

Matrix of covariances of the initial state P:

$$P(0|0) = \left[(\tfrac{0.5}{3})^2\right] \tag{6.6}$$

For the **Iveco Platform**, it never was implemented.

## 6.3. Feedback Controller

The Feedback controller is a key block that close the control loop to achieve the path following of the UGS. The control loop is closed using the distance between the UGS and the path $d$, getting the error: $d_{error} = d_{ref} - \hat{d}$. The $d_{ref}$ is always set to zero, because the goal is keep the car over the path, getting: $d_{error} = -\hat{d}$. The feedback controller also uses other UGS variables, like the estimated speed $\hat{u}_1$ and the estimated angle between the UGS and the path $\hat{\theta}_e$.

The feedback controller is composed by two Fuzzy Logic Controllers in cascade as you can see in figure 6.1:
The first one (the $FC - (PID + \theta_e)$) is the one who takes into account the distance error to close the loop. It has four inputs: distance error $d_{error}(k)$; sum of

distance error $\int d_{error}(k) = d_{error}(k) + \int d_{error}(k-1)$; change in distance error $\dot{d}_{error} = d_{error}(k) - d_{error}(k-1)$ and angle between the UGS and the circuit $\hat{\theta}_e$. It has one output that is the reference of the steering wheel, but without taking account the speed of the UGS $\phi_{ref-woSpeed}$.
The second one (the velocity compensator), only acts to change the steering wheel reference depending on the speed of the UGS. It has two inputs: the first one is the output of the first controller, the reference of the steering wheel without speed $\phi_{ref-woSpeed}$; and the second one is the estimated speed of the UGS $\hat{u}_1$. The only output is the reference of the steering wheel $\phi_{ref}$.

Both controllers has the following characteristics:

- Type: Mamdani
- And Method: Product
- Or Method: Probabilistic Or
- Implication: Product
- Aggregation: Sum
- Defuzzyfication: Centroid

Both controllers has gains at each input and at each output. Because of that, the fuzzyfication and the defuzzyfication can be done without taking account the dimensionality. These gains are adjusted after using a Genetic Algorithm (GA) in a simulator. It could be considered as an auto-tune of the controller. The Genetic Algorithm Loop used to auto-tune the controller is shown in figure 6.2. The fitness used in the GA is the inverse of the Integral of Square Error (ISE). Other fitness could be used like Integral of Time and Square Error (ITSE), Integral of Time and Absolute Error (ITAE), Integral of Absolute Error (IAE),.. But the ISE is the one who better fits with the objective of the controller, because use the time (ITAE, ITSE,..) is not good because of the circuit changes along the time and the tune will not be good. The use of the square error is better that the absolute error, because it is minimized in a better way, taking more account the bigger ones than the smaller ones.

### 6.3.1. The first controller: $FC-(PID+\theta_e)$

As it is said before, this controller has four inputs and one output, whose number of linguistic values and their values are:

- Input 1 (distance error $d_{error}(k)$). It has three linguistic values: N, Z, P.
- Input 2 (sum of distance error $\int d_{error}(k)$). It has three linguistic values: N, Z, P.

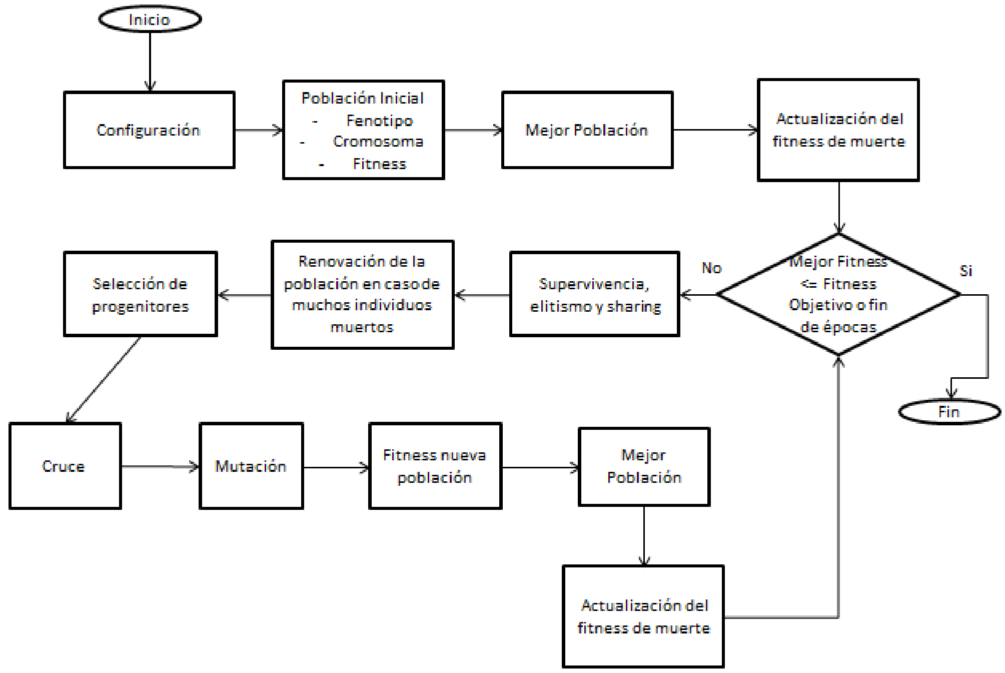

Inicio
Configuración
Población Inicial
- Fenotipo
- Cromosoma
- Fitness
Mejor Población
Actualización del fitness de muerte
Mejor Fitness <= Fitness Objetivo o fin de épocas
No
Si
Fin
Supervivencia, elitismo y sharing
Renovación de la población en caso de muchos individuos muertos
Selección de progenitores
Cruce
Mutación
Fitness nueva población
Mejor Población
Actualización del fitness de muerte

Figure 6.2: Genetic Algorithm Loop used to auto-tune the controllers

- Input 3 (change in distance error $\dot{d}_{error}$). It has three linguistic values: N, Z, P.

- Input 4 (angle between the UGS and the circuit $\hat{\theta}_e$). It has three linguistic values: N, Z, P.

- Output (steering wheel reference without taking account the speed of the UGS $\phi_{ref-woSpeed}$). It has nine linguistic values: NE, NMG, NG, NP, Z, PP, PG, PMG, PE.

The fuzzyfication of the inputs are shown in figure 6.3. As it is said before, they are done without taking into account the dimensionality.

The defuzzyfication of the output is shown in figure 6.4.

The controller has 81 if-then rules, shown in the figures 6.5 to 6.12

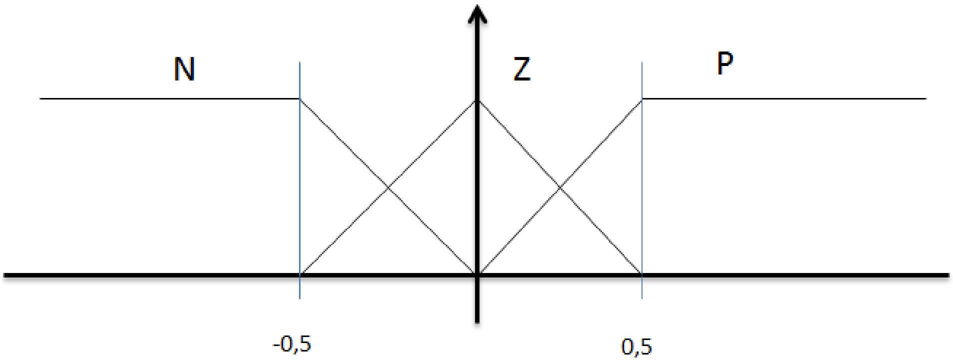


Figure 6.3: Fuzzyfication of the inputs 1 to 4 of the $FC-(PID+\theta_e)$ controller. All inputs are similar, because of are fuzzyficated without dimensionality

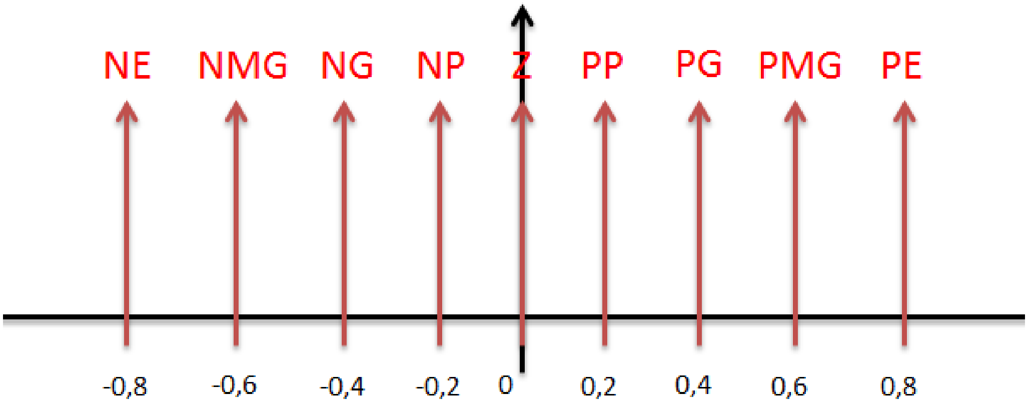
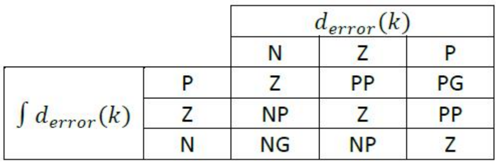
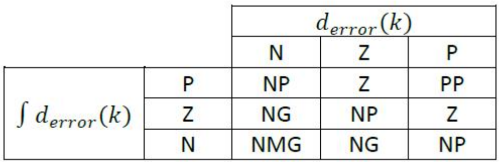


Figure 6.4: Defuzzyfication of the output of the $FC-(PID+\theta_e)$ controller.



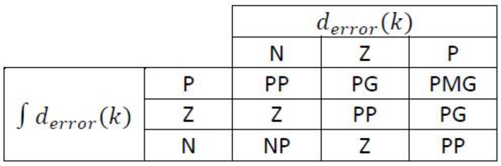
| | | $d_{error}(k)$ | | |
|---|---|---|---|---|
| | | N | Z | P |
| $\int d_{error}(k)$ | P | Z | PP | PG |
| | Z | NP | Z | PP |
| | N | NG | NP | Z |

Figure 6.5: Rules of the $FC-(PID+\theta_e)$ controller: $\dot{d}_{error}=Z$ & $\dot{\hat{\theta}}_e=Z$



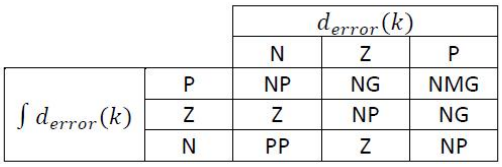
| | | $d_{error}(k)$ | | |
|---|---|---|---|---|
| | | N | Z | P |
| $\int d_{error}(k)$ | P | NP | Z | PP |
| | Z | NG | NP | Z |
| | N | NMG | NG | NP |

Figure 6.6: Rules of the $FC-(PID+\theta_e)$ controller: $\dot{d}_{error}=Z$ & $\dot{\hat{\theta}}_e=P$



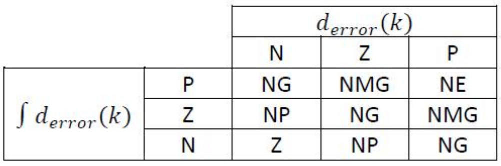
| | | $d_{error}(k)$ | | |
|---|---|---|---|---|
| | | N | Z | P |
| $\int d_{error}(k)$ | P | PP | PG | PMG |
| | Z | Z | PP | PG |
| | N | NP | Z | PP |

Figure 6.7: Rules of the $FC-(PID+\theta_e)$ controller: $\dot{d}_{error}=Z$ & $\dot{\hat{\theta}}_e=N$



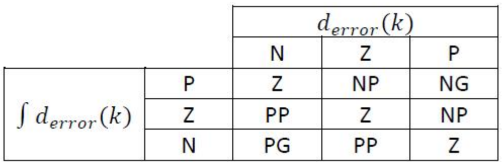
| | | $d_{error}(k)$ | | |
|---|---|---|---|---|
| | | N | Z | P |
| $\int d_{error}(k)$ | P | NP | NG | NMG |
| | Z | Z | NP | NG |
| | N | PP | Z | NP |

Figure 6.8: Rules of the $FC-(PID+\theta_e)$ controller: $\dot{d}_{error}=N$ & $\dot{\hat{\theta}}_e=Z$



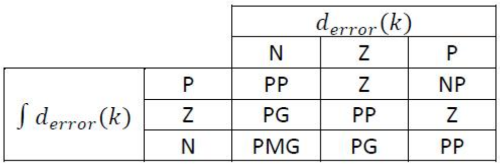
| | | $d_{error}(k)$ | | |
|---|---|---|---|---|
| | | N | Z | P |
| $\int d_{error}(k)$ | P | NG | NMG | NE |
| | Z | NP | NG | NMG |
| | N | Z | NP | NG |

Figure 6.9: Rules of the $FC-(PID+\theta_e)$ controller: $\dot{d}_{error}=N$ & $\dot{\hat{\theta}}_e=P$



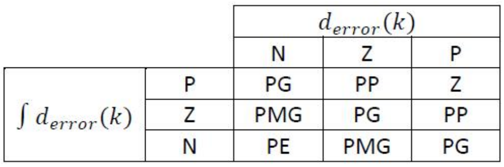
| | | $d_{error}(k)$ | | |
|---|---|---|---|---|
| | | N | Z | P |
| $\int d_{error}(k)$ | P | Z | NP | NG |
| | Z | PP | Z | NP |
| | N | PG | PP | Z |

Figure 6.10: Rules of the $FC-(PID+\theta_e)$ controller: $\dot{d}_{error}=N$ & $\dot{\hat{\theta}}_e=N$



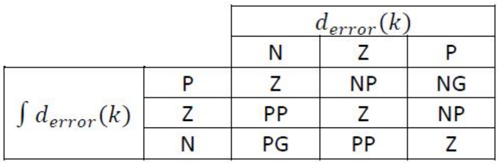
| | | $d_{error}(k)$ | | |
|---|---|---|---|---|
| | | N | Z | P |
| $\int d_{error}(k)$ | P | PP | Z | NP |
| | Z | PG | PP | Z |
| | N | PMG | PG | PP |

Figure 6.11: Rules of the $FC-(PID+\theta_e)$ controller: $\dot{d}_{error}=P$ & $\dot{\hat{\theta}}_e=Z$



| | | $d_{error}(k)$ | | |
|---|---|---|---|---|
| | | N | Z | P |
| $\int d_{error}(k)$ | P | PG | PP | Z |
| | Z | PMG | PG | PP |
| | N | PE | PMG | PG |

Figure 6.12: Rules of the $FC-(PID+\theta_e)$ controller: $\dot{d}_{error}=P$ & $\dot{\hat{\theta}}_e=P$



| | | $d_{error}(k)$ | | |
|---|---|---|---|---|
| | | N | Z | P |
| $\int d_{error}(k)$ | P | Z | NP | NG |
| | Z | PP | Z | NP |
| | N | PG | PP | Z |

Figure 6.13: Rules of the $FC-(PID+\theta_e)$ controller: $\dot{d}_{error}=P$ & $\dot{\hat{\theta}}_e=N$

### 6.3.2. The second controller: the velocity compensator

This controller has two inputs and one output, whose number of linguistic values and their values are:

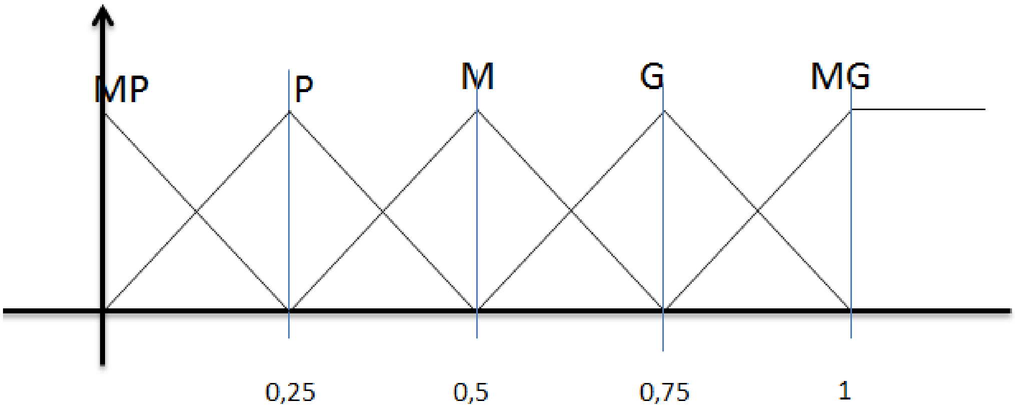


Figure 6.14: Fuzzyfication of the input 1 of the velocity compensator.

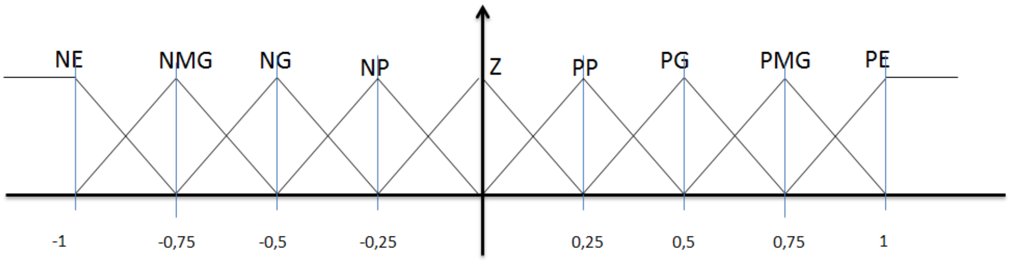


Figure 6.15: Fuzzyfication of the input 2 of the velocity compensator.

- Input 1 (steering wheel reference without taking account the speed of the UGS $\phi_{ref-woSpeed}$). It has nine linguistic values: NE, NMG, NG, NP, Z, PP, PG, PMG, PE.
- Input 2 (sum of distance error $\int d_{error}(k)$). It has five linguistic values: MP, P, M, G, MG.
- Output (steering wheel reference $\phi_{ref}$). It has seventeen linguistic values: NE, NMMG, NMG, NG, NM, NP, NMP, NMMP, Z, PMMP, PMP, PP, PM, PG, PMG, PMMG, PE.

The fuzzyfication of the first input is shown in figure 6.14. The fuzzyfication of the second input is shown in figure 6.15. As it is said before, they are done without taking into account the dimensionality.

The defuzzyfication of the output is shown in figure 6.16.

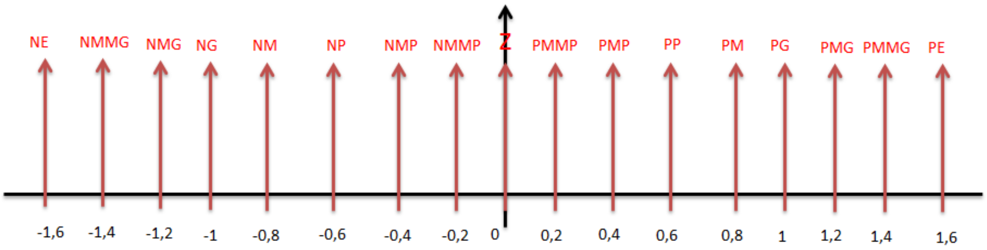
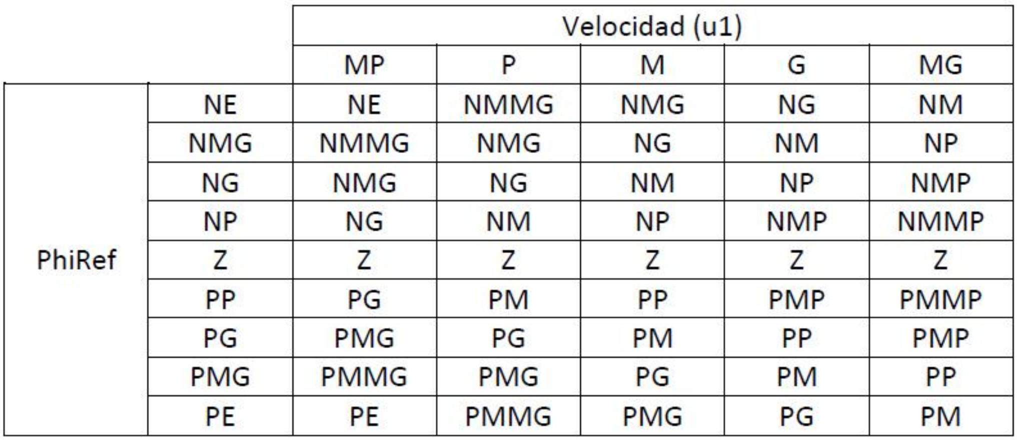


Figure 6.16: Defuzzyfication of the output of the velocity compensator.



| | | Velocidad (u1) | | | | |
|---|---|---|---|---|---|---|
| | | MP | P | M | G | MG |
| PhiRef | NE | NE | NMMG | NMG | NG | NM |
| | NMG | NMMG | NMG | NG | NM | NP |
| | NG | NMG | NG | NM | NP | NMP |
| | NP | NG | NM | NP | NMP | NMMP |
| | Z | Z | Z | Z | Z | Z |
| | PP | PG | PM | PP | PMP | PMMP |
| | PG | PMG | PG | PM | PP | PMP |
| | PMG | PMMG | PMG | PG | PM | PP |
| | PE | PE | PMMG | PMG | PG | PM |

Figure 6.17: Rules of the velocity compensator.

The controller has 47 if-then rules, shown in the figure 6.5.

# Chapter 7

# Simulation Results

In this chapter, simulation results of the proposed High-Level Controller are shown.

To simulate the UGS, it is used the non-lineal model of a Frenet-Frame kinematic model of a car-like mobile robot shown in (Siciliano, 2008). The features of the platforms like $L$, $l_{1-cam}$, $l_{2-cam}$, $l_{1-mag}$ and $l_{2-mag}$, described in chapter 3 are included in the simulations as parameters. Also white noise is included in all the measures and all the actuator values.

As the proposed High-Level controller is too big, the simulation results are divided in two parts: the simulation results of the State Estimator (section 7.1) and the Controller (section 7.2). The objective of that is evaluate the goodness of each part of the proposed control scheme separately.

## 7.1. State Estimator

In this section, the State Estimator is simulated, with the goal of see the benefits of use it on the real UGS. This first group of simulations close the control loop with a really simple PID controller with only proportional action is used, because it is not the goal of these simulations. The Wheel Position Estimator and the Speed Estimator modules are not simulated (see section 5.1 and 5.2).

## First Simulation

The **first simulation**, simulates the Citroën C3 Platform moving on the Egg-Circuit (section 3.3.1) with the speed reference shown in figure 7.1. In this simulation, the circuit simulated is the Paint Circuit and it is known a priori.

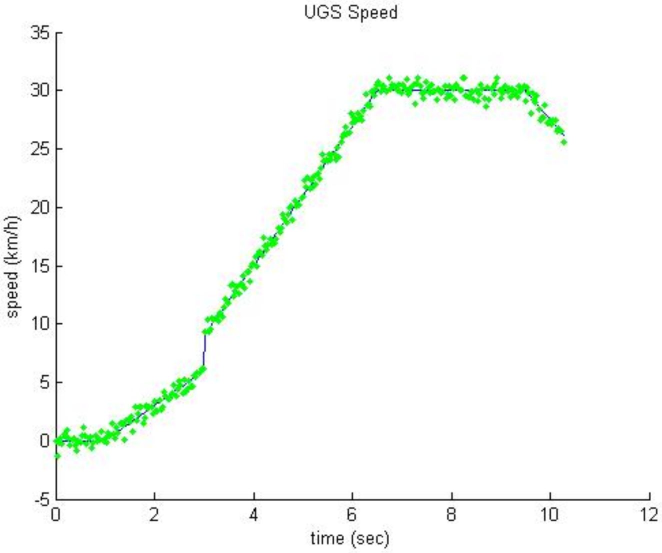


Figure 7.1: Speed Reference. In blue, the real speed of the UGS ($u_1$). In green samples of the estimated speed $\hat{u}_1$).

In figure 7.2 the distance to the circuit is represented. Figure 7.3 represents the angle to the circuit. The advance over the circuit is shown in figure 7.4. Curvature of the circuit is shown in figure 7.5. The reference of the steering wheel can be see in figure 7.6. Finally, figure 7.7 shows the X-Y position of the UGS.

As you it can see, the state estimation is really good. The estimated state is always close to the real state demonstrated by a little MSE between the real state and the estimated state, even with highly noisy measurements.

## Second Simulation

In the **second simulation**, the simulated UGS and the circuit are the same than in the first simulation, with the same speed reference, but know, the Paint Circuit is not known a priori, been its curvature estimated. The results of this simulation are shown in figures 7.8 to 7.13. In figure 7.8 the distance to the

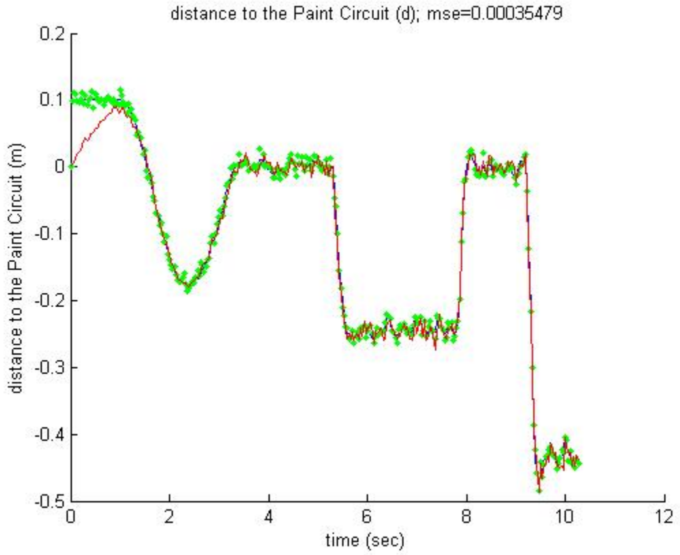


Figure 7.2: Distance of the UGS to the Paint Circuit (d). In blue, the real distance ($d$); in green the samples of the measured distance ($\tilde{d}$); and in red, the estimated distance ($\hat{d}$).

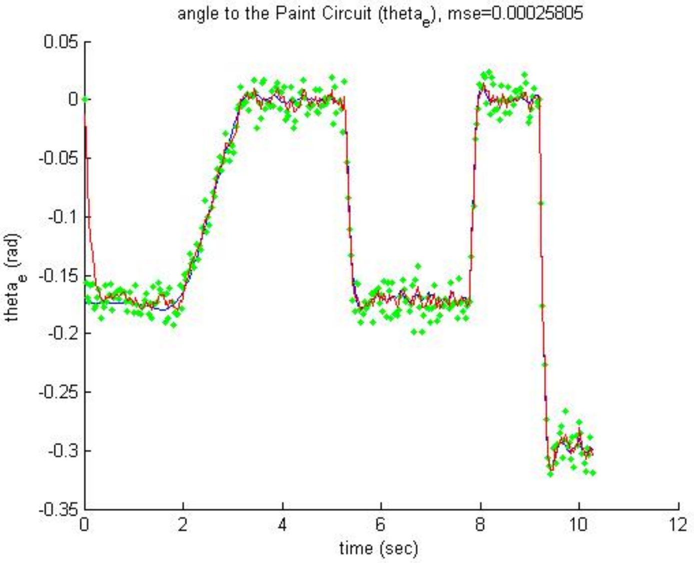


Figure 7.3: Angle $\theta_e$ between the UGS and the Paint Circuit. In blue, the real angle ($\theta_e$); in green the samples of the measured angle ($\tilde{\theta}_e$); and in red, the estimated angle ($\hat{\theta}_e$).

circuit is represented. Figure 7.9 represents the angle to the circuit. The advance over the circuit is shown in figure 7.10. Curvature of the circuit is shown in figure 7.11. The reference of the steering wheel can be see in figure 7.12. Finally, figure

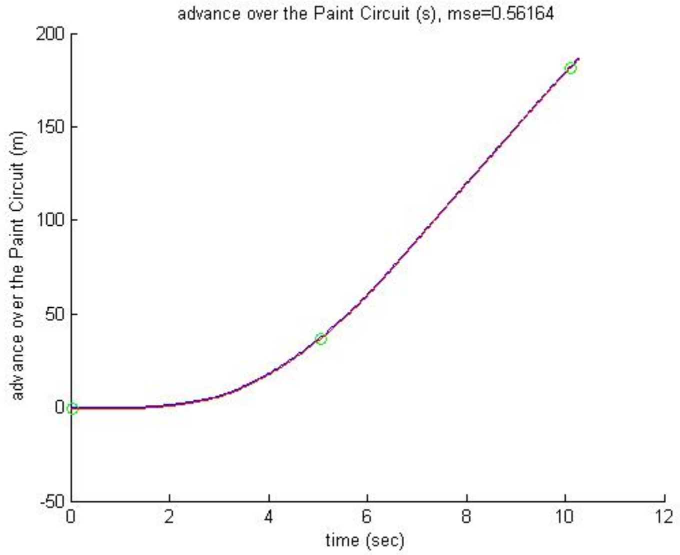


Figure 7.4: Advance of the UGS over the Paint Circuit (s). In blue, the real advance ($s$); in green the samples of the measured advance ($\tilde{s}$); and in red, the estimated advance ($\hat{s}$).

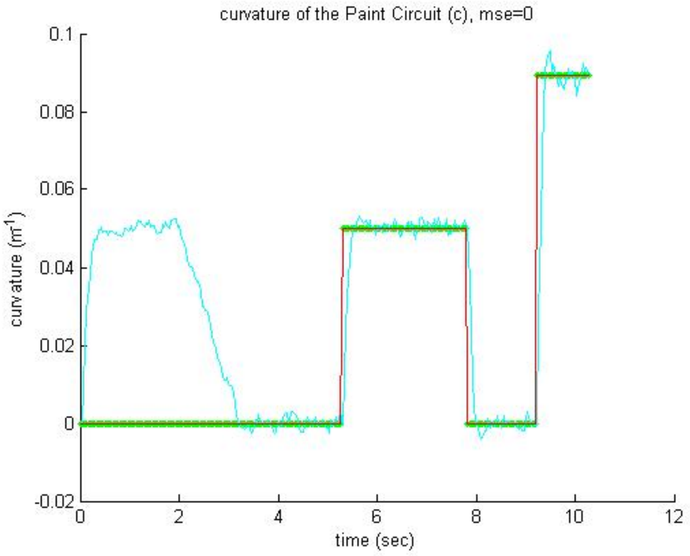


Figure 7.5: Curvature of the Paint Circuit (c). In blue, the real curvature ($c$); in green the samples of the measured curvature using the circuit information ($\tilde{c}_{circ}$); in cyan the samples of the measured curvature using the theoretical information ($\tilde{c}_{teor}$); and in red, the estimated curvature ($\hat{c}$).

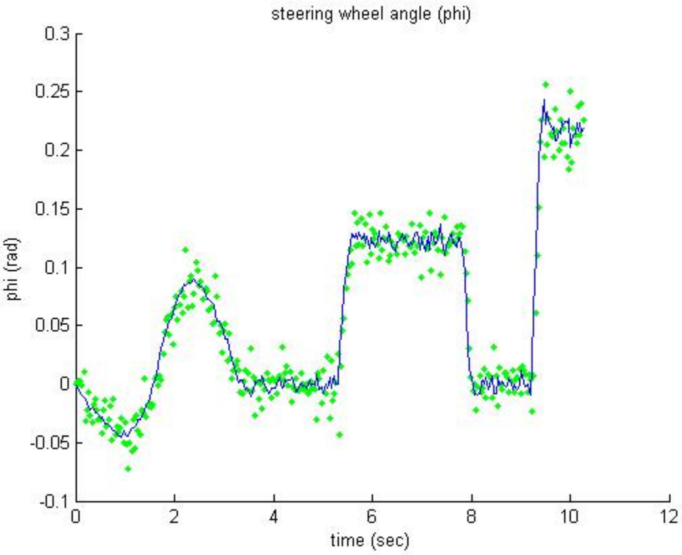


Figure 7.6: Steering wheel angle generated by the P controller ($\phi$). In blue, the wheel angle reference ($\phi_{ref}$); and in green the samples of the estimation of the steering angle ($\hat{\phi}$).

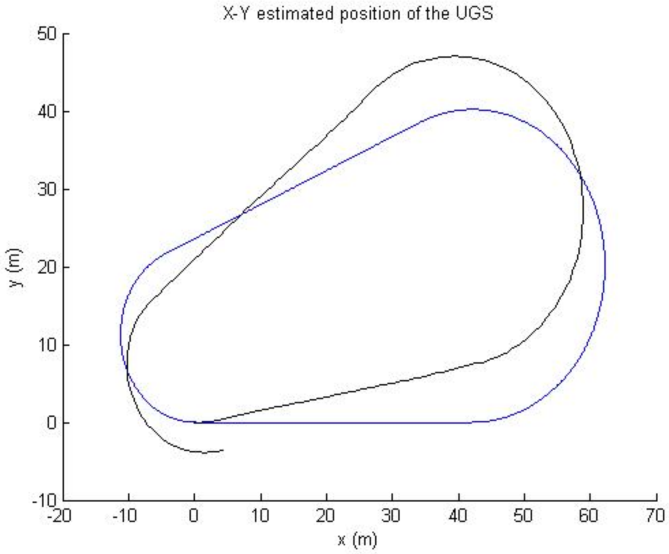


Figure 7.7: Position X-Y estimated of the vehicle. In blue the real circuit. In black the position X-Y estimated ($\hat{x}$ and $\hat{y}$). The estimation is never corrected.

7.13 shows the X-Y position of the UGS.

In this simulation we can see that the a priori knowledge of the circuit is better than the ignorance of it, because the MSE obtained between the real value and the estimated value of $\hat{d}$, $\hat{\theta}_e$, $\hat{s}$ and $\hat{c}$ are lower. We can see also with this simulation, that the Curvature Estimator can estimate a good curvature $\hat{c}$ even if no circuit information is available, only by using $\tilde{c}_{teor}$.

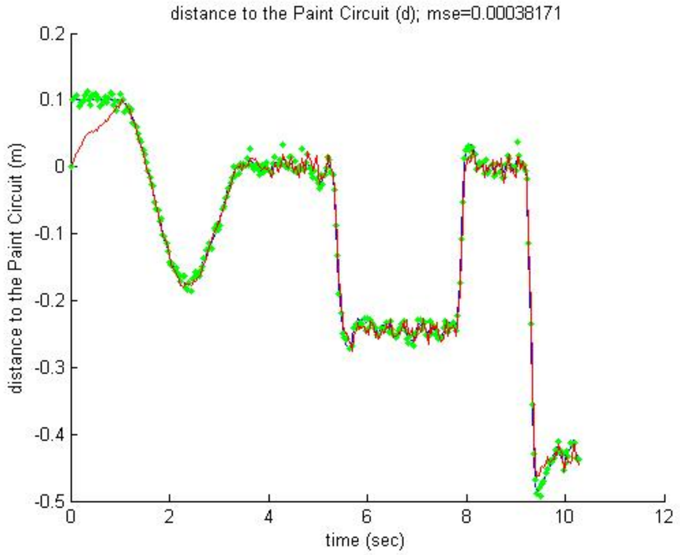


Figure 7.8: Distance of the UGS to the Paint Circuit (d). In blue, the real distance ($d$); in green the samples of the measured distance ($\tilde{d}$); and in red, the estimated distance ($\hat{d}$).

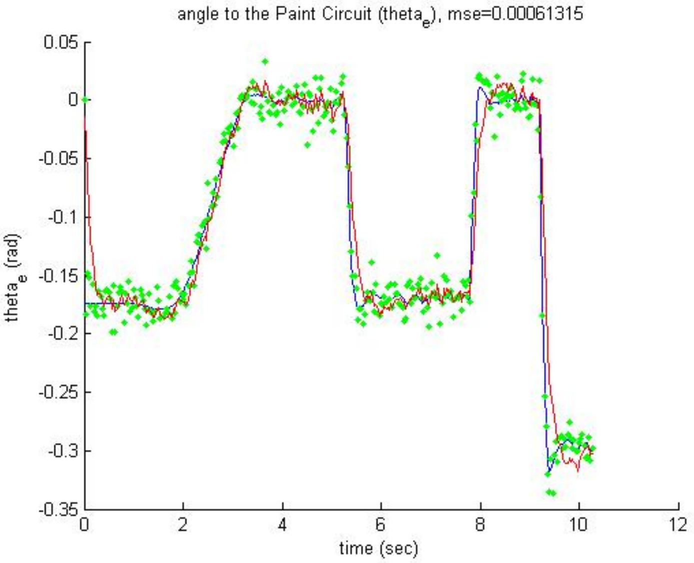


Figure 7.9: Angle $\theta_e$ between the UGS and the Paint Circuit. In blue, the real angle ($\theta_e$); in green the samples of the measured angle ($\tilde{\theta}_e$); and in red, the estimated angle ($\hat{\theta}_e$).

## Third Simulation

With the **third simulation** (the simulated UGS and the circuit are the same than in the first simulation, with the same speed reference), we want to demonstrate that the use of the Curvature Estimator is better than suppose a curvature

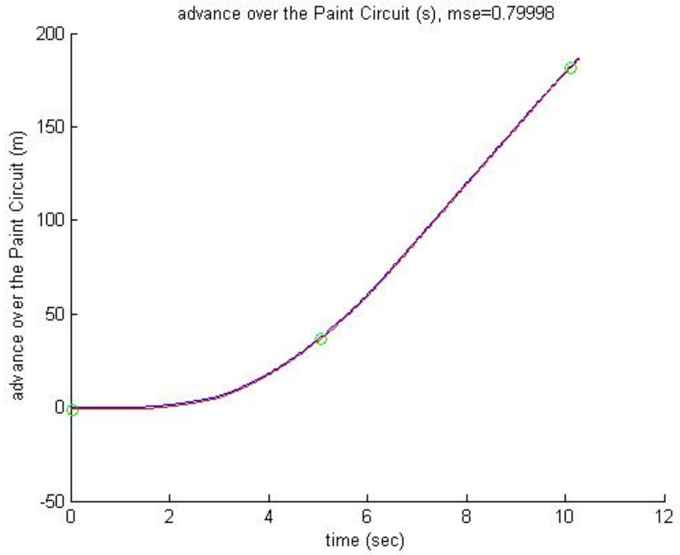


Figure 7.10: Advance of the UGS over the Paint Circuit (s). In blue, the real advance ($s$); in green the samples of the measured advance ($\tilde{s}$); and in red, the estimated advance ($\hat{s}$).

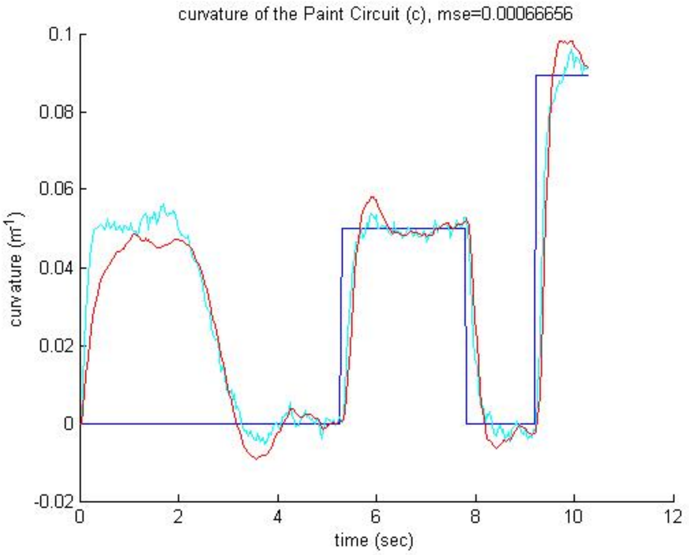


Figure 7.11: Curvature of the Paint Circuit (c). In blue, the real curvature ($c$); in cyan the samples of the measured curvature using the theoretical information ($\tilde{c}_{teor}$); and in red, the estimated curvature ($\hat{c}$).

$\hat{c} = 0$. Now the curvature is not estimated, and it is set to zero in the State Estimator. In figure 7.14 the distance to the circuit is represented. Figure 7.15 represents the angle to the circuit. The advance over the circuit is shown in figure 7.16. Curvature of the circuit is shown in figure 7.17. The reference of the steering wheel can be see in figure 7.18. Finally, figure 7.19 shows the X-Y position

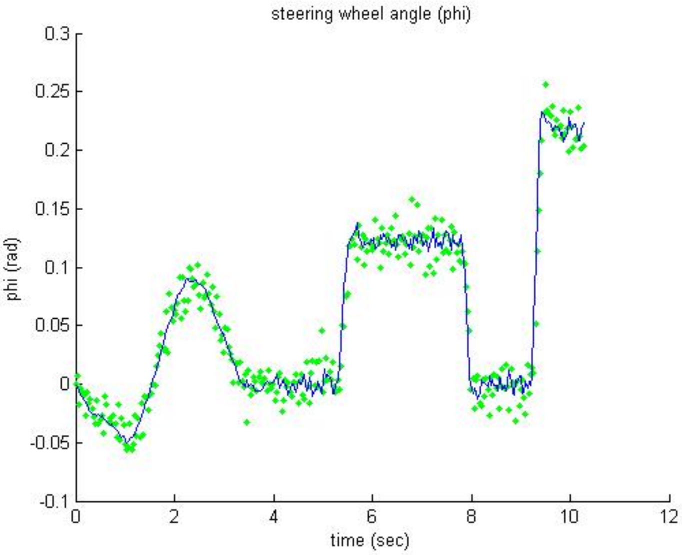


Figure 7.12: Steering wheel angle generated by the P controller ($\phi$). In blue, the wheel angle reference ($\phi_{ref}$); and in green the samples of the estimation of the steering angle ($\hat{\phi}$).

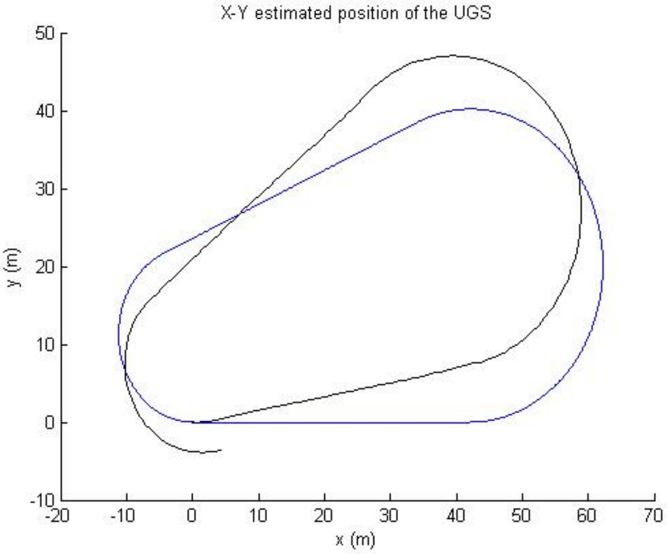


Figure 7.13: Position X-Y estimated of the vehicle. In blue the real circuit. In black the position X-Y estimated ($\hat{x}$ and $\hat{y}$). The estimation is never corrected.

of the UGS.

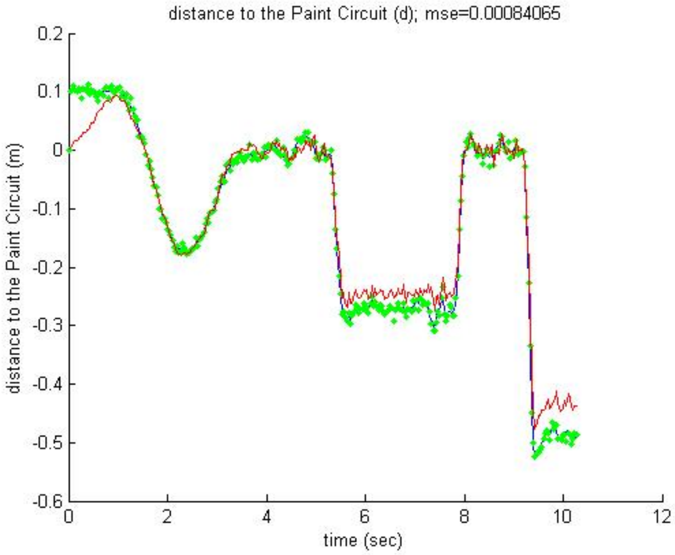


Figure 7.14: Distance of the UGS to the Paint Circuit (d). In blue, the real distance ($d$); in green the samples of the measured distance ($\tilde{d}$); and in red, the estimated distance ($\hat{d}$).

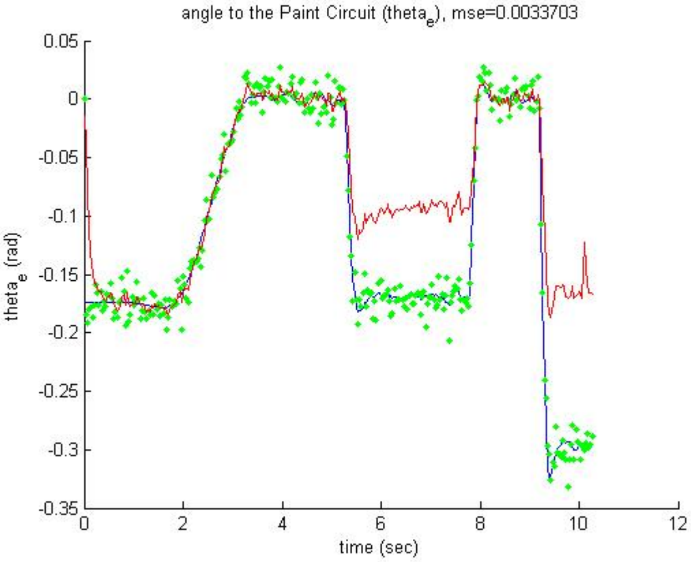


Figure 7.15: Angle $\theta_e$ between the UGS and the Paint Circuit. In blue, the real angle ($\theta_e$); in green the samples of the measured angle ($\tilde{\theta}_e$); and in red, the estimated angle ($\hat{\theta}_e$).

As you can see, the use of an unknown curvature $\hat{c} = 0$ give us worst results (bigger MSE between the real value and the estimated value of $\hat{d}$, $\hat{\theta}_e$, $\hat{s}$ and of course in $\hat{c}$) than if we know the real curvature (simulation 1) or if we estimate the curvature (simulation 2).

In the tree simulation, you can see that the X-Y position estimator does not

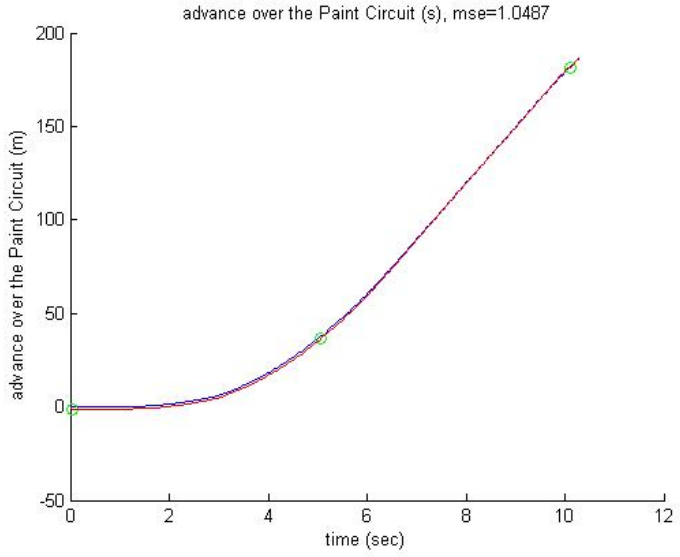


Figure 7.16: Advance of the UGS over the Paint Circuit (s). In blue, the real advance ($s$); in green the samples of the measured advance ($\tilde{s}$); and in red, the estimated advance ($\hat{s}$).

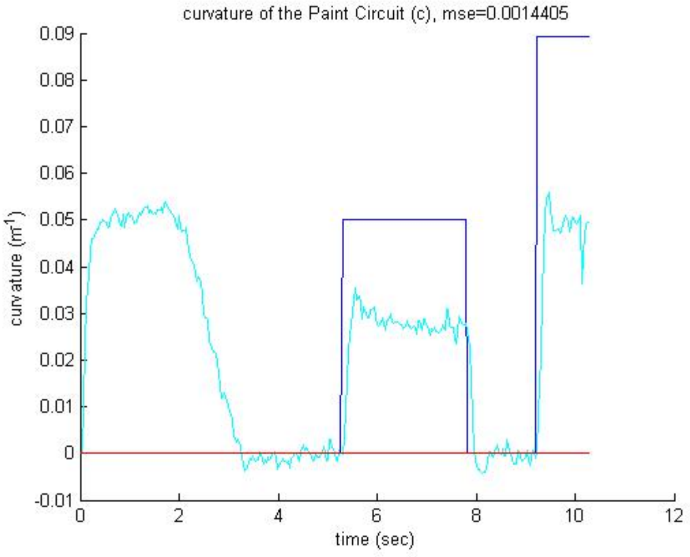


Figure 7.17: Curvature of the Paint Circuit (c). In blue, the real curvature ($c$); in cyan the samples of the measured curvature using the theoretical information ($\tilde{c}_{teor}$); and in red, the estimated curvature ($\hat{c}$).

work really bad, because even the estimated position is not similar to the circuit position, it has the same shape. We do not have to forget that in these simulations, the X-Y position is never corrected (there are no codes with the real X-Y information).

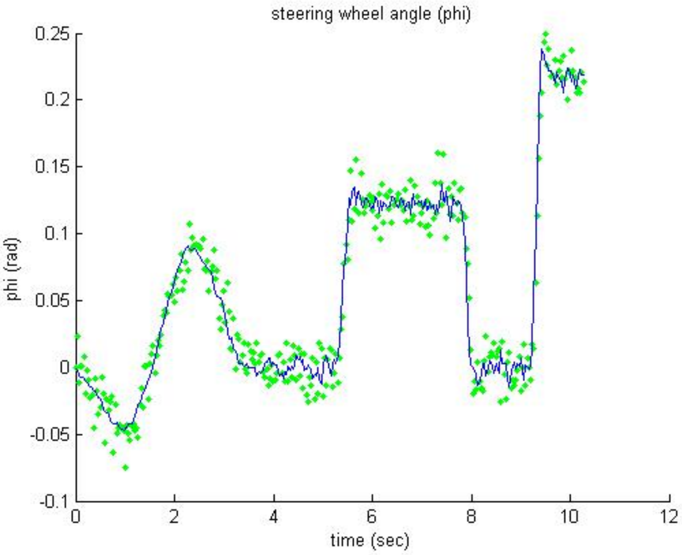


Figure 7.18: Steering wheel angle generated by the P controller ($\phi$). In blue, the wheel angle reference ($\phi_{ref}$); and in green the samples of the estimation of the steering angle ($\hat{\phi}$).

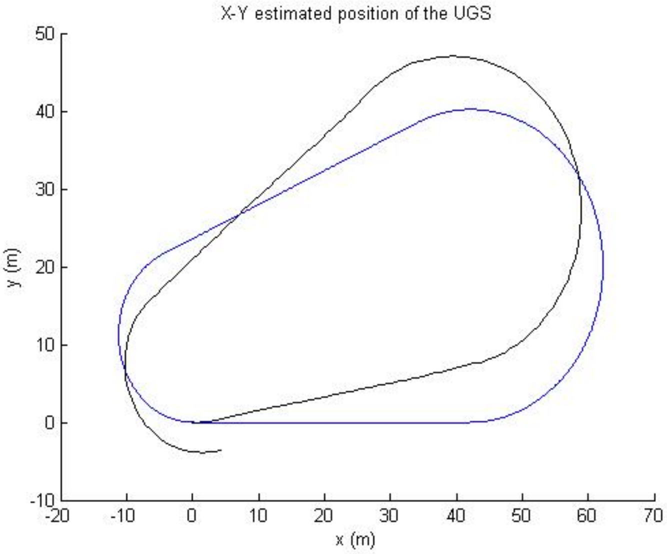


Figure 7.19: Position X-Y estimated of the vehicle. In blue the real circuit. In black the position X-Y estimated ($\hat{x}$ and $\hat{y}$). The estimation is never corrected.

## Fourth Simulation

Now, a **fourth simulation** to demonstrate the goodness of the Paint Circuit Generator (that is a sub-block of the Curvature Estimator) is done. In this simulation, the O Circuit (see section 3.3.2) is used. The Iveco Platform is also used.

The output of this sub-block is shown in figure 7.20, where you can see that it works, calculating the Paint Circuit curvature using the information of the Magnet Circuit.

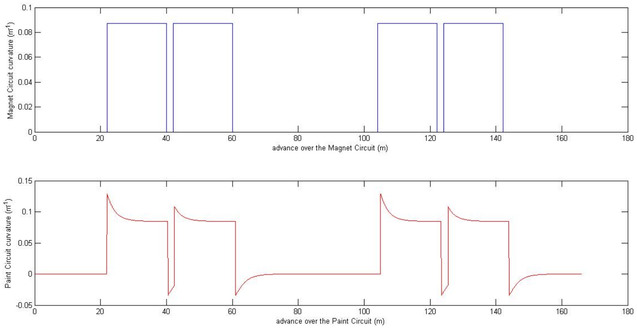


Figure 7.20: Curvatures of the O Circuit. Upper plot, in blue the curvature of the Magnet Circuit, the input of the Paint Circuit Generator; bottom plot, in red, the curvature of the Paint Circuit.

The goodness of the State Estimator had been demonstrated through this four simulations.

## 7.2. Controller

In this section the goodness of the proposed Controller is demonstrated. Two simulations are done. In the first one, the Feed-Forward controller is simulated. In the second one, the Feed-Back controller is simulated. Both controller are analysed in different simulations to avoid the possible interference between both.

### First Simulation

The **first simulation** uses the same scenario than the first simulation of the section 7.1, but now, the action of the Feed-Forward controller is added to the existing P controller. The Paint circuit is known and the state Estimator is used.

Simulation result are shown in figures 7.21 to 7.25. In figure 7.21 the distance to the circuit is represented. Figure 7.22 represents the angle to the circuit. The advance over the circuit is shown in figure 7.23. Curvature of the circuit is shown in figure 7.24. The reference of the steering wheel can be see in figure 7.25. With this Subsystem, the control action of the Feed-Back controller is reduced thanks to the change of the operating point made by the Feed-Forward

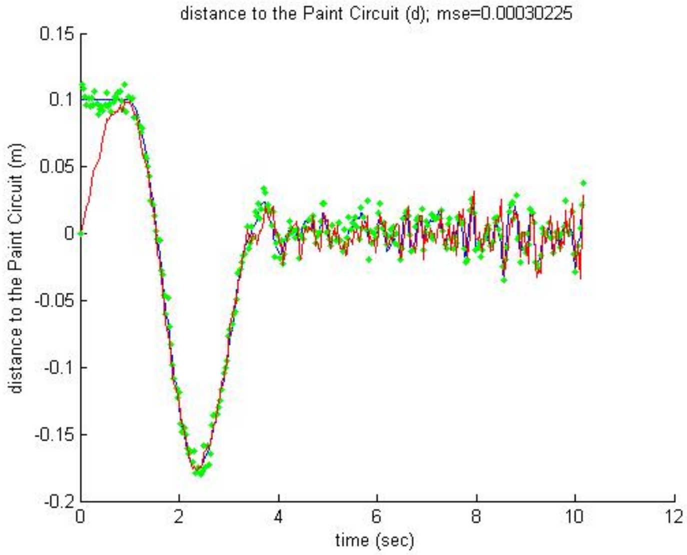


Figure 7.21: Distance of the UGS to the Paint Circuit (d). In blue, the real distance ($d$); in green the samples of the measured distance ($\tilde{d}$); and in red, the estimated distance ($\hat{d}$).

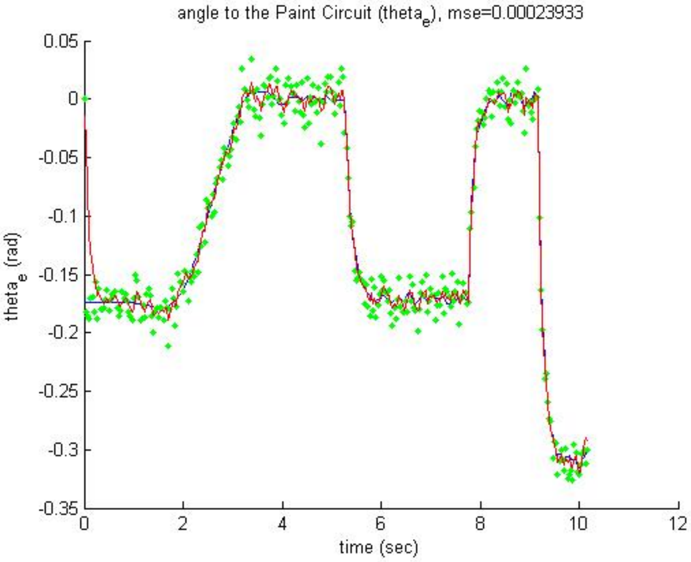


Figure 7.22: Angle $\theta_e$ between the UGS and the Paint Circuit. In blue, the real angle ($\theta_e$); in green the samples of the measured angle ($\tilde{\theta}_e$); and in red, the estimated angle ($\hat{\theta}_e$).

controller. Now the MSE between the distance to the circuit and the reference ($d_{ref} = 0$) is reduced to 0.0039 $m^2$ (without the Feed-Forward controller it was 0.0363 $m^2$, ten times more). The goodness of the Feed-Forward controller had been demonstrated.

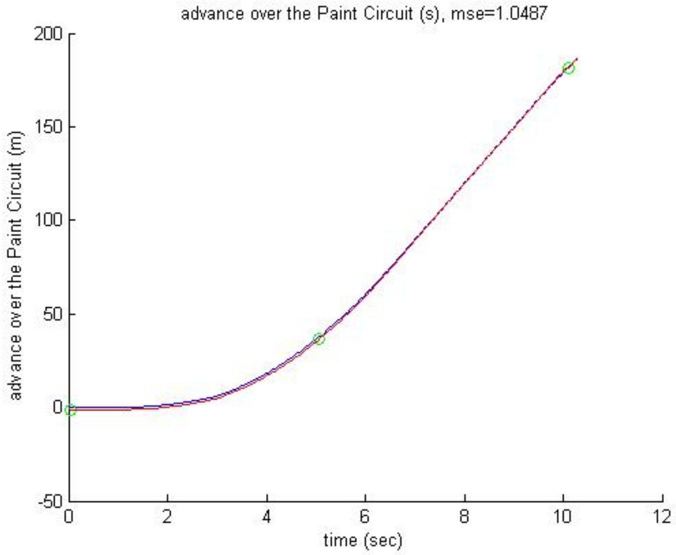


Figure 7.23: Advance of the UGS over the Paint Circuit (s). In blue, the real advance ($s$); in green the samples of the measured advance ($\tilde{s}$); and in red, the estimated advance ($\hat{s}$).

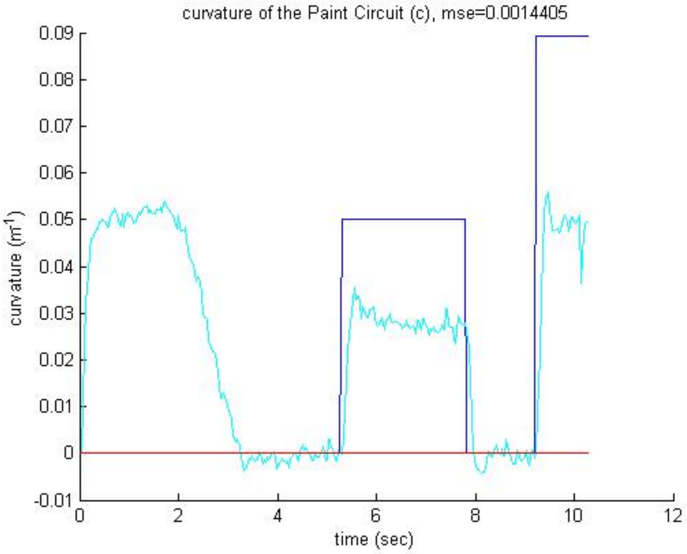


Figure 7.24: Curvature of the Paint Circuit (c). In blue, the real curvature ($c$); in cyan the samples of the measured curvature using the theoretical information ($\tilde{c}_{teor}$); and in red, the estimated curvature ($\hat{c}$).

## Second Simulation

The **second simulation** changes the scenario used before in order to put more stress to the Feed-Back controller. Now the State Estimator is removed, and the controller is used with the measurements (as we saw in section 7.1, this is a

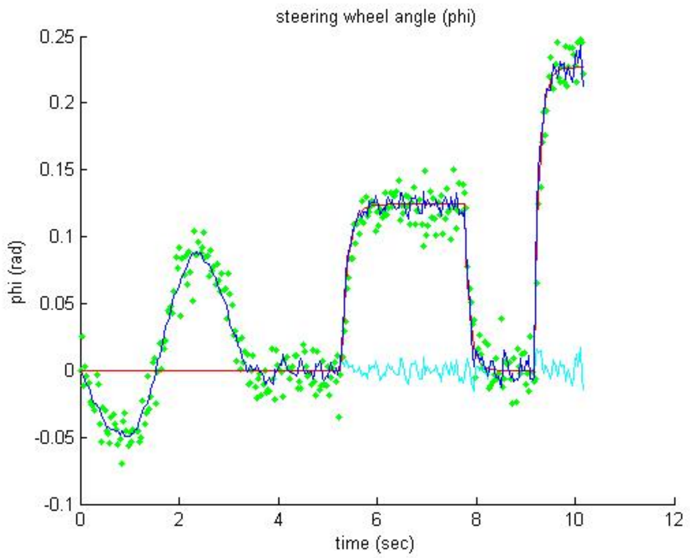


Figure 7.25: Steering wheel angle generated by the Controller ($\phi$). In blue, the wheel angle reference ($\phi_{ref} = \Delta\phi + \phi_{lin}$); in red the action of the Feed-Forward controller ($\phi_{lin}$); in cyan, the action of the P controller ($\Delta\phi$); and in green the samples of the estimation of the steering angle ($\hat{\phi}$).

worst case). There is no Feed-Forward controller. The UGS platform simulated is the Citroën C3 and the circuit used is the Egg Circuit. To put more stress in the controller, a non-modelled wheel dynamics is simulated as a part of the platform. More stress is added to the controller by a step-changing speed (see figure 7.26).

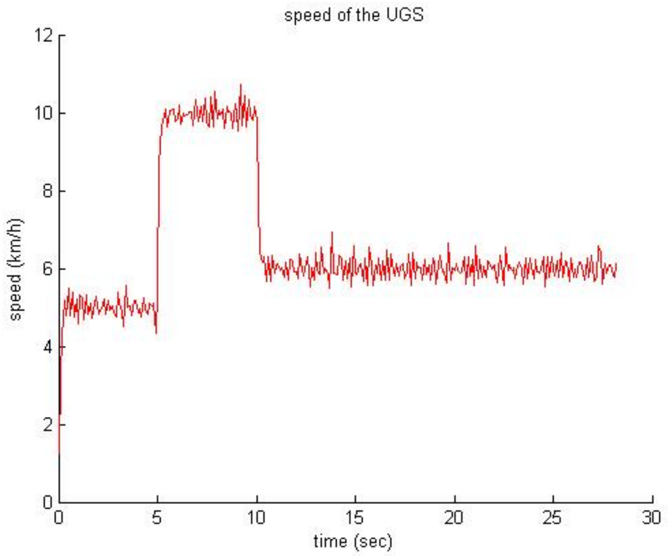


Figure 7.26: Measured Speed of the UGS ($\tilde{u}_1$). It is used as estimated speed of the UGS ($\hat{u}_1$).

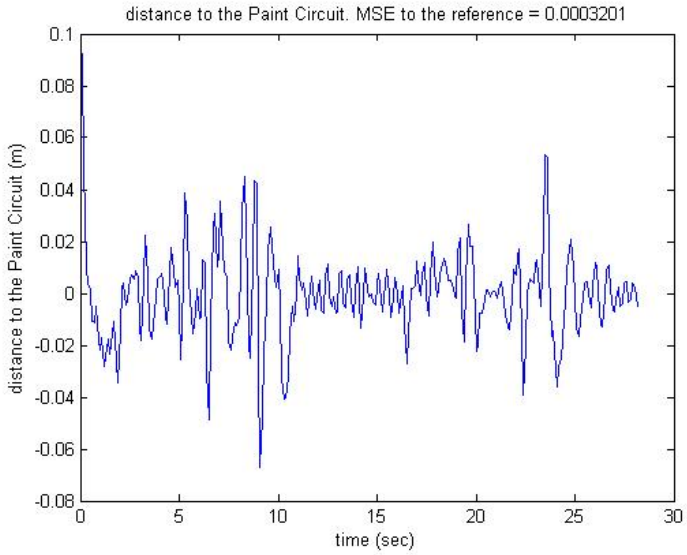


Figure 7.27: Distance of the UGS to the Paint Circuit (d).

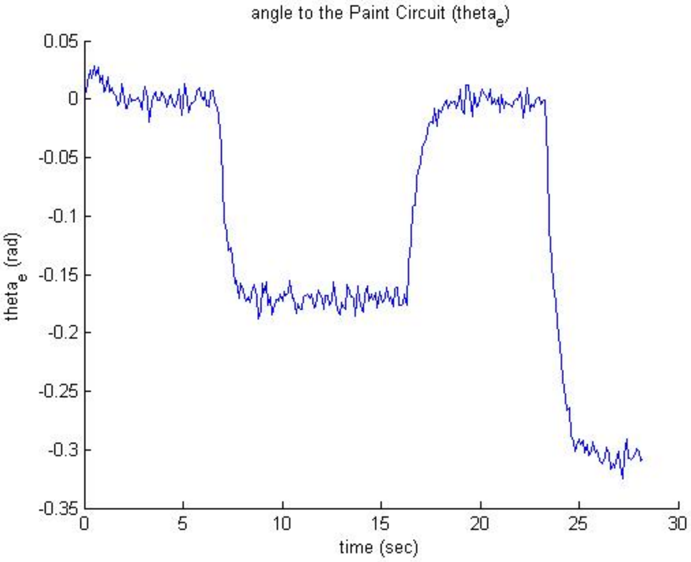


Figure 7.28: Angle $\theta_e$ between the UGS and the Paint Circuit.

The results of this simulation are shown in figures 7.27 to 7.29. In figure 7.27 the distance to the circuit is shown. Figure 7.28 shows the angle to the circuit. Finally, the steering angle is shown in figure 7.29 The goodness of this Feed-Back controller is demonstrated thanks to the really low MSE between the measured distance to the Paint Circuit and the reference ($d_{ref} = 0$), that is $MSE = 0.0003201m^2$.

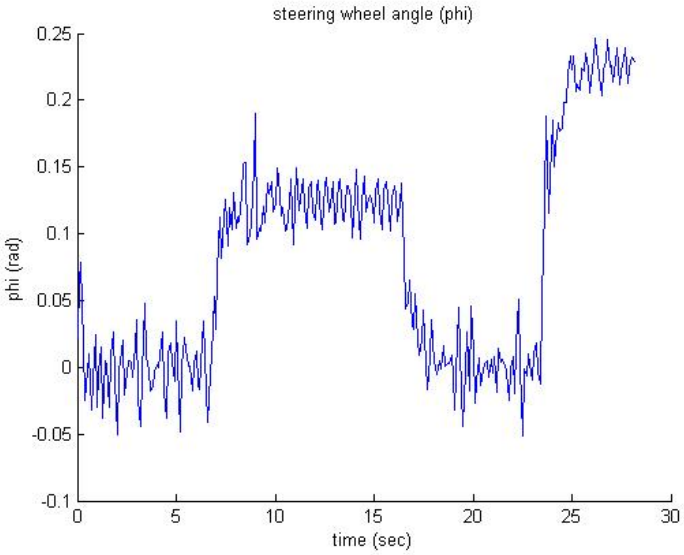


Figure 7.29: Steering wheel angle generated by the Feed-Back Controller ($\phi_{ref}$). There is no Feed-Forward Controller ($\phi_{lin} = 0$).

Thanks to the two previous simulation, the goodness of the proposed Controller is demonstrated.

# Chapter 8

# Results on real UGS

In this chapter the results of the implementation of the proposed High-Level Controller in the UGS are shown. As is it said in previous chapters, the Project ends before than it has to. That means that the proposed High-Level Controller never could be implemented in the Iveco Platform and it was partially implemented in the Citroën C3 Platform.

For the **Citroën C3 Platform**, the whole State Estimator was implemented and tested, but the subsystem Speed Estimator and Wheel Position Estimator are really simple (see chapter 5). The Controller was partially implemented. The Wheel Dynamic Compensator was really simple for this platform (see chapter 6). The Feed-Back Controller never could be implemented therefore, the old Feed-Back Controller of this Platform (see section 3.1.3) was used as a Feed-Back Controller.

Lots of tests were done to prove the goodness of this proposed High-Level Control System. The subsystem was implemented gradually, testing a lot each step to avoid non-identified problems.

All the test were done using the Egg Circuit.

## 8.1. Previous Tests

In this section, the old High-Level Control System is tested. These tests are done to demonstrate the improvements of the proposed High-Level Control

System in the following sections.

The **first test** (figures 8.1 to 8.5) shows the performance of the old High-Level controller. If the UGS is moving with a low speed (between 9 and 17 km/h, figure 8.1), the Paint Circuit can be followed. The plot of time vs distance to the circuit (figure 8.2) shows that the UGS is always following the reference ($d_{ref} = 0$). The MSE between the measured distance and the reference is $3761.22 pix^2$. With a green circle are shown lost samples. When too much samples are lost, the performance of the controller is not good and the UGS lose the path. In figure 8.3, the angle between the UGS and the circuit is shown (
$tilde\theta_e$). In figure 8.4, the estimated advance over the circuit is shown ($\hat{s}$). Now, the estimated advance is calculated using the equation $\hat{d}(k) = \hat{d}(k-1) + \hat{u}_1 \dot{\Delta} time$. In green, samples of the measured advance over the circuit, provided by the codes. The correction of the state estimation does not use any KF. Finally, in figure 8.5, the steering reference ($\phi_{ref}$) is shown. This is the output of the old High-Level Control System.

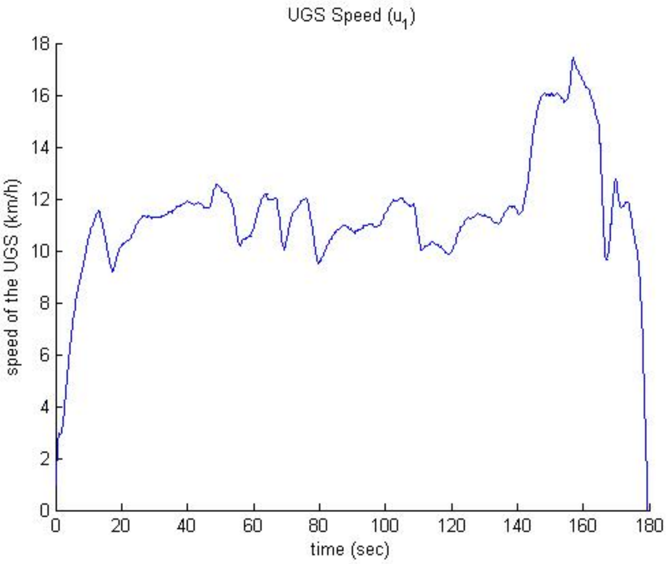


Figure 8.1: Measured Speed of the UGS ($\tilde{u}_1$).

In the **second test** (figures 8.6 to 8.10) the speed of the UGS is increased up to $24km/h$ (figure 8.6). With this speed the Paint Circuit is lost. You can see in figure 8.7, that the distance measured became bigger than in the previous test, and finally, when the UGS lost a sample, the UGS cannot recover the circuit. In figure 8.8 the angle $\tilde{\theta}_e$ is shown and in figure 8.9 you can see the estimated

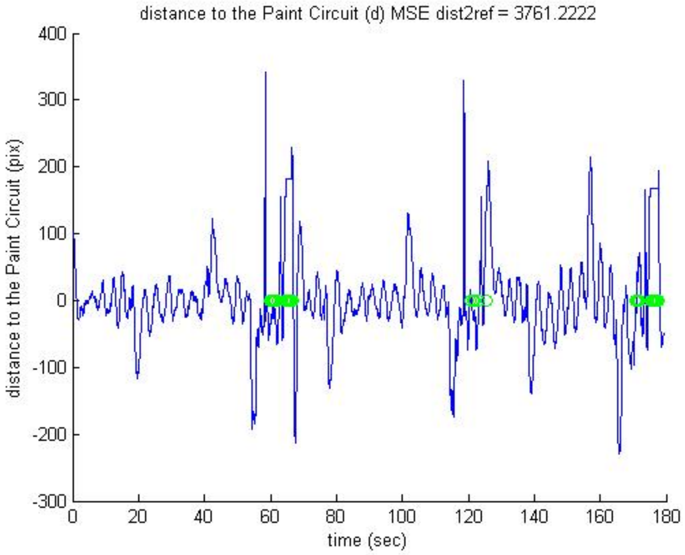


Figure 8.2: Distance of the UGS to the Paint Circuit (d). In green measured distance to the Paint Circuit ($\tilde{d}$). In blue estimated distance to the Paint Circuit ($\hat{d} = \tilde{d}$). If a sample of the distance is lost (a green circle), the last value is used.

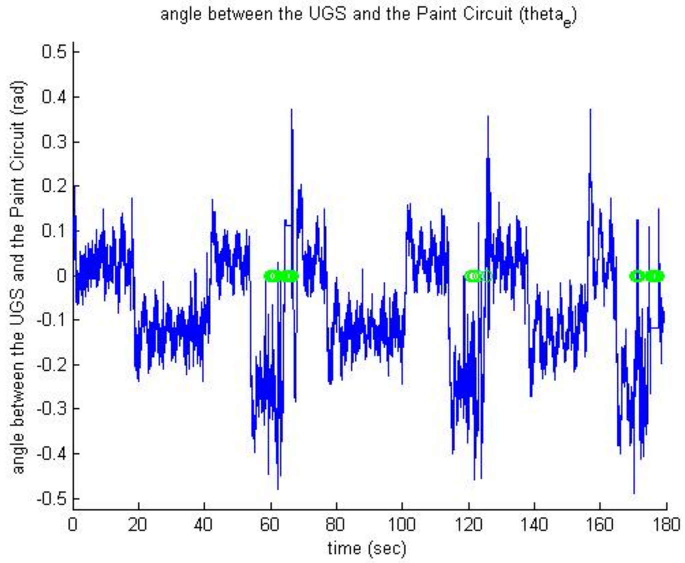


Figure 8.3: Angle $\theta_e$ between the UGS and the Paint Circuit. In green, measured angle ($\tilde{\theta}_e$). In blue estimated angle ($\hat{\theta}_e = \tilde{\theta}_e$). If a sample of the angle is lost (a green circle), the last value is used.

advance. Finally, figure 8.10 shows the steering wheel reference.

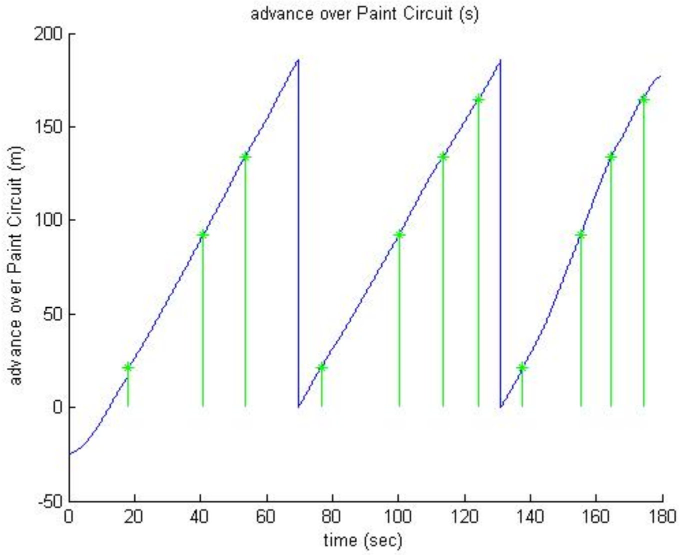


Figure 8.4: Advance over the Paint Circuit ($s$). In green, measured advance over the circuit ($\tilde{s}$); in blue, estimated advance over the circuit ($\hat{d}$) using the equation $\hat{d}(k) = \hat{d}(k-1) + \hat{u}_1 \dot{\Delta} time$.

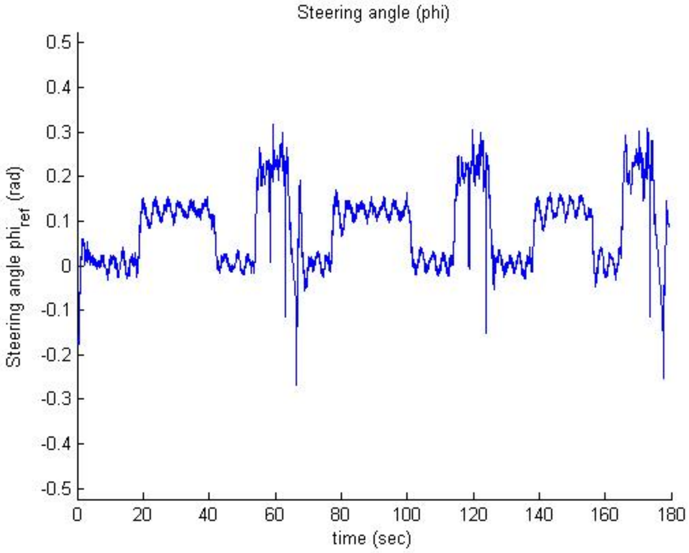


Figure 8.5: Steering wheel angle generated by the old High-Level Controller ($\phi_{ref}$).

## 8.2. First Addition: The Position Over Circuit Estimator & The X-Y Position Estimator

Now, the Position Over Circuit Estimator and the X-Y Position estimator that are subsystems of the State Estimator, are added to the High-Level Control

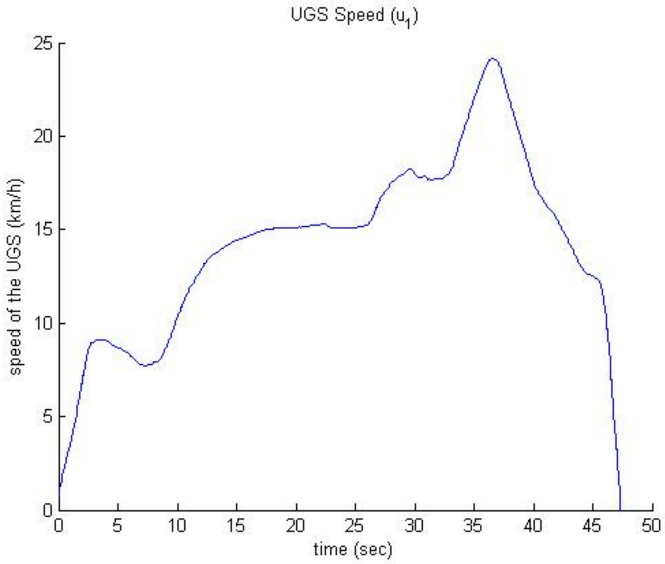


Figure 8.6: Measured Speed of the UGS ($\tilde{u}_1$).

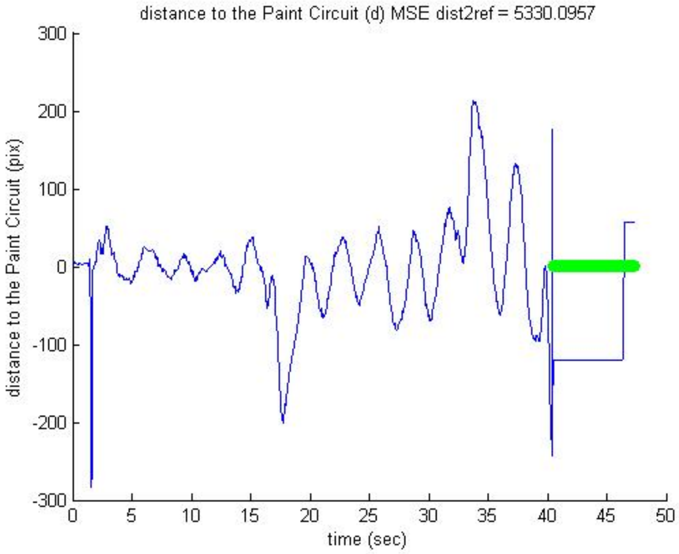


Figure 8.7: Distance of the UGS to the Paint Circuit (d). In green measured distance to the Paint Circuit ($\tilde{d}$). In blue estimated distance to the Paint Circuit ($\hat{d} = \tilde{d}$). If a sample of the distance is lost (a green circle), the last value is used.

System of the UGS.

In this test (figures 8.11 to 8.17), the UGS is moving with a similar speed than in the first test of the section 8.1 (see figure 8.11), but now the performance was augmented and allows the UGS to achieve more than 8 laps without lose the path (in figure 8.14, you can see the estimated advance over the circuit). Even if some samples of the distance are lost (in figure 8.12 is represented the measured and the estimated distance to the path), the Line Lost Intelligence allows the UGS to

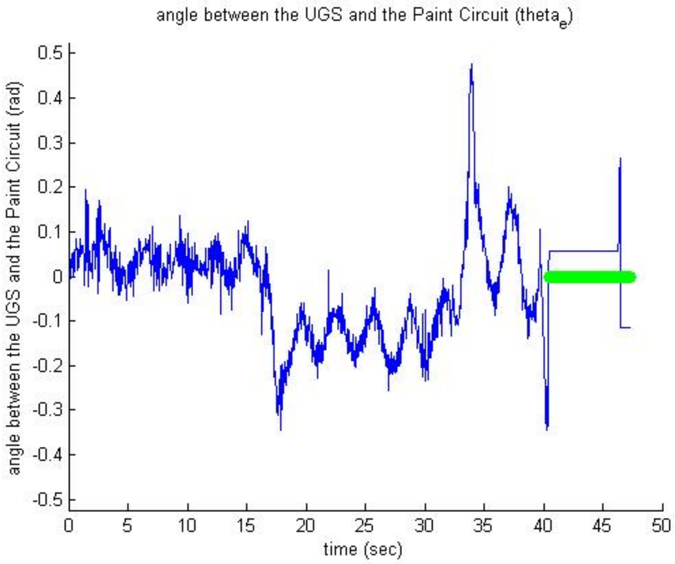


Figure 8.8: Angle $\theta_e$ between the UGS and the Paint Circuit. In green, measured angle ($\tilde{\theta}_e$). In blue estimated angle ($\hat{\theta}_e = \tilde{\theta}_e$). If a sample of the angle is lost (a green circle), the last value is used.

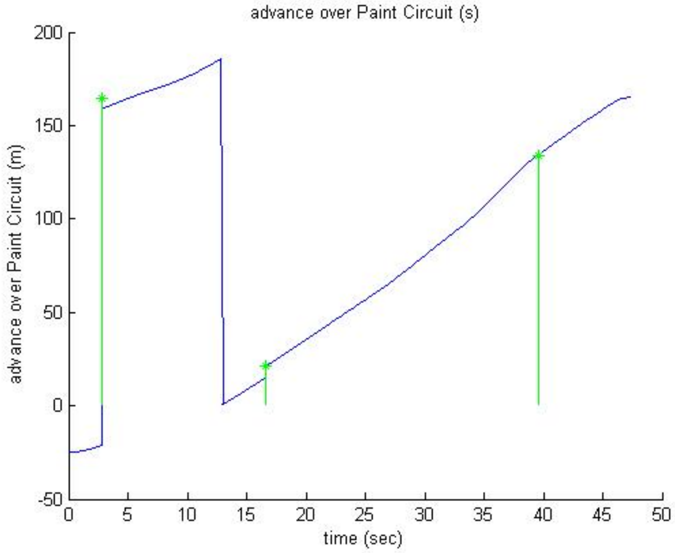


Figure 8.9: Advance over the Paint Circuit ($s$). In green, measured advance over the circuit ($\tilde{s}$); in blue, estimated advance over the circuit ($\hat{d}$) using the equation $\hat{d}(k) = \hat{d}(k-1) + \hat{u}_1 \dot{\Delta} time$.

continue. Now, the estimated curvature of the circuit $\hat{c}$ is set to zero (figure 8.15). In figure 8.13, you can see the measured and estimated angle between the UGS and the circuit. The X-Y Position Estimator of the UGS does not work really well (figure 8.17), but this is because only at least four times per lap, the X-Y position is corrected, not been enough. Finally, in figure 8.16, you can see the

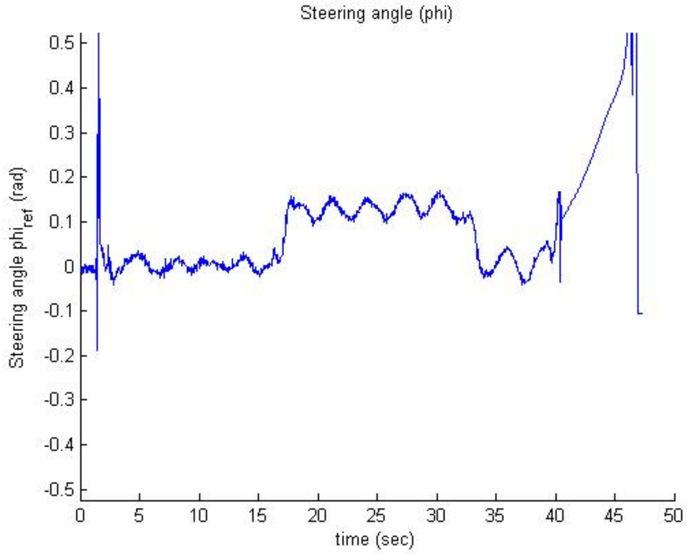


Figure 8.10: Steering wheel angle generated by the old High-Level Controller ($\phi_{ref}$).

steering wheel reference.

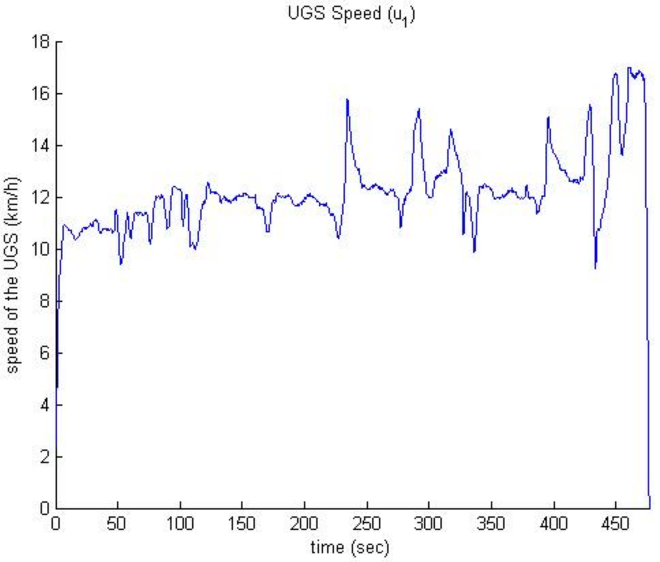


Figure 8.11: Measured Speed of the UGS ($\tilde{u}_1$).

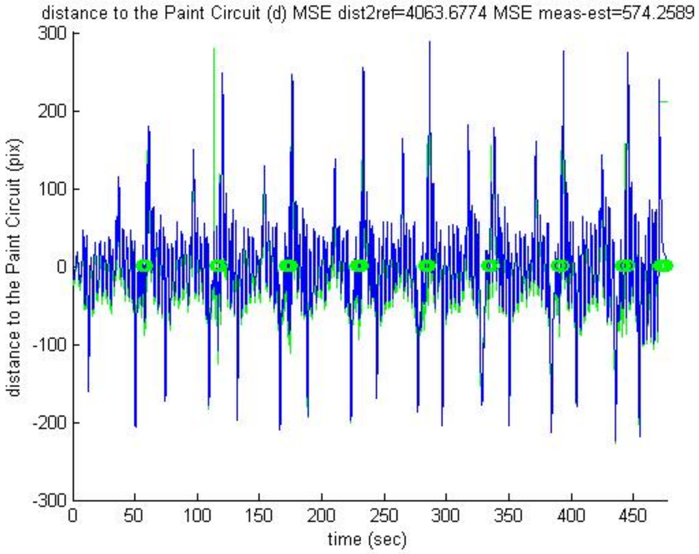


Figure 8.12: Distance of the UGS to the Paint Circuit (d). In green measured distance to the Paint Circuit ($\tilde{d}$). In blue estimated distance to the Paint Circuit ($\hat{d}$). Lost samples are represented by a green circle.

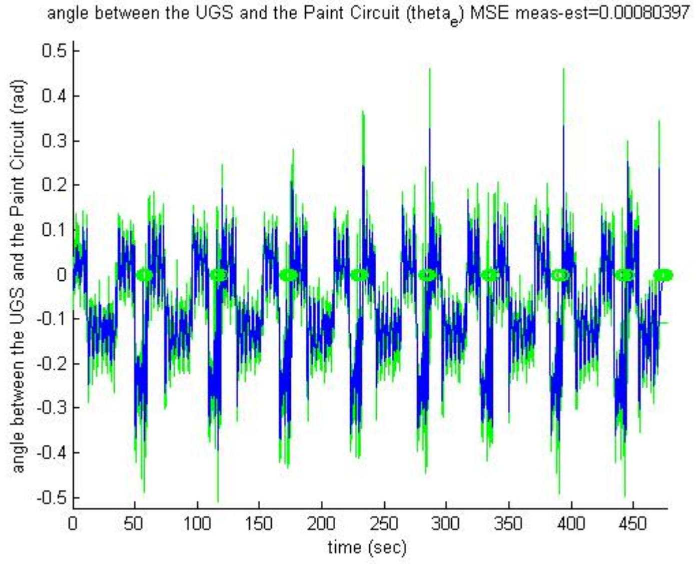


Figure 8.13: Angle $\theta_e$ between the UGS and the Paint Circuit. In green, measured angle ($\tilde{\theta}_e$). In blue estimated angle ($\hat{\theta}_e$). Lost samples are represented by a green circle.

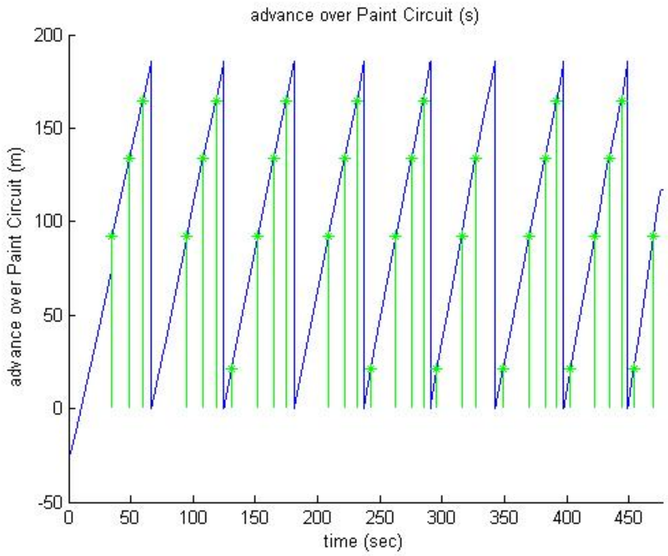


Figure 8.14: Advance over the Paint Circuit ($s$). In green, measured advance over the circuit ($\tilde{s}$); in blue, estimated advance over the circuit ($\hat{d}$).

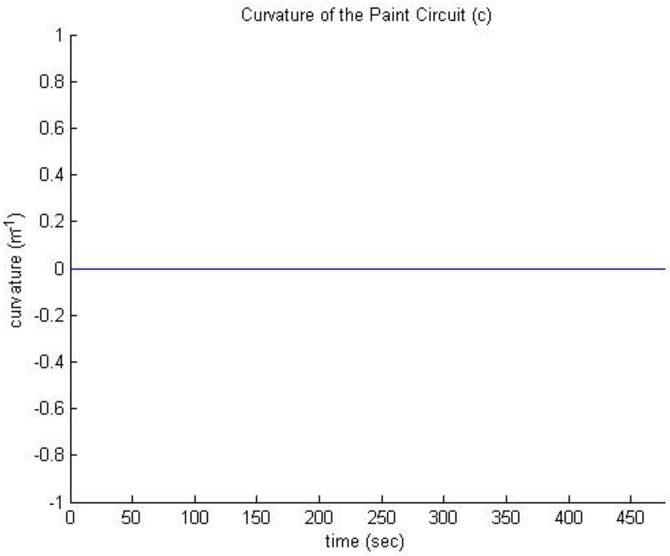


Figure 8.15: Curvature of the paint circuit ($c$). In blue, the estimated Curvature ($\hat{c}$).

## 8.3. Second Addition: The Theoretical Curvature Measure Subsystem

With the addition of the Theoretical Curvature Measure Subsystem, the circuit curvature is estimated even if there is not previous knowledge of the Paint Circuit or the Magnet Circuit. The performance of the Position Over Circuit

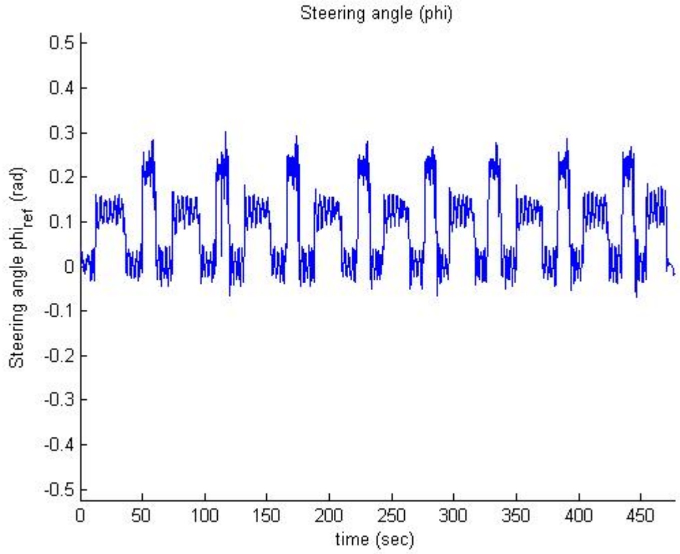


Figure 8.16: Steering wheel angle generated by the old High-Level Controller ($\phi_{ref}$).

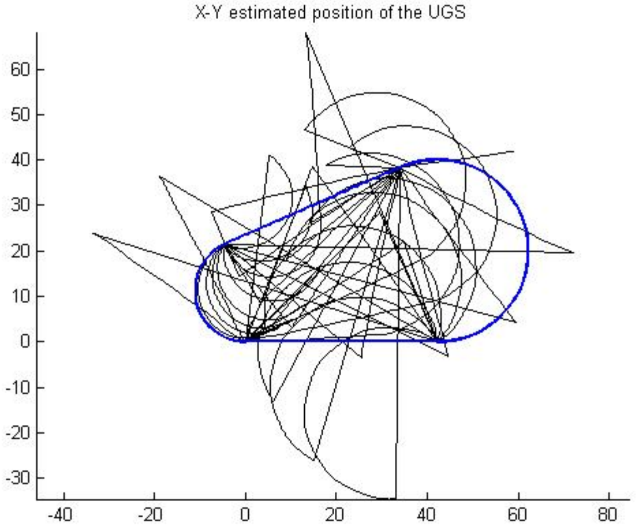


Figure 8.17: Position X-Y of the UGS. In black, the estimated position ($\hat{x}$, $\hat{y}$). In blue, the position of the real Paint Circuit.

Estimator will be improved.

In the **first test** (figures 8.18 to 8.24), the UGS is moving with a similar speed than in the previous test (figure 8.18). Now the curvature is estimated, and it look similar to the real curvature of the circuit (figure 8.22). It increase the performance of the proposed High-Level Control System, and allow the UGS to follow the circuit without lose it. The distance is shown in figure 8.19 and the angle to the path in figure 8.20. In figure 8.21, the advance over the circuit can be shown. The X-Y Position estimation is shown in figure 8.24. The steering reference is shown in figure 8.23.

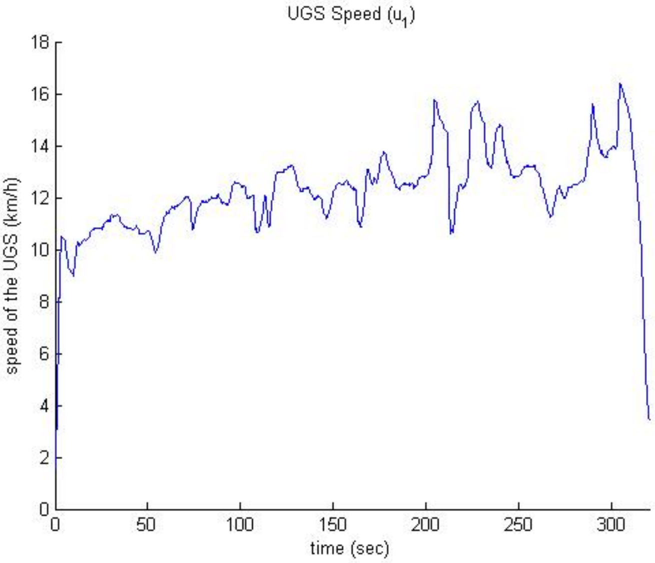


Figure 8.18: Measured Speed of the UGS ($\tilde{u}_1$).

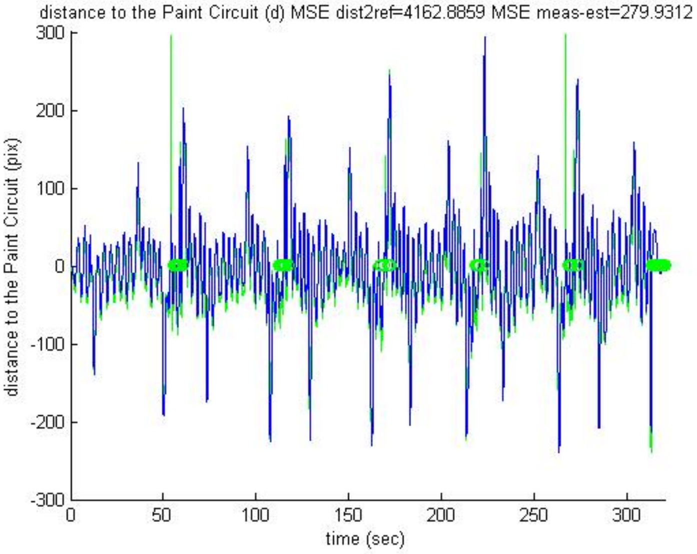


Figure 8.19: Distance of the UGS to the Paint Circuit (d). In green measured distance to the Paint Circuit ($\tilde{d}$). In blue estimated distance to the Paint Circuit ($\hat{d}$). Lost samples are represented by a green circle.

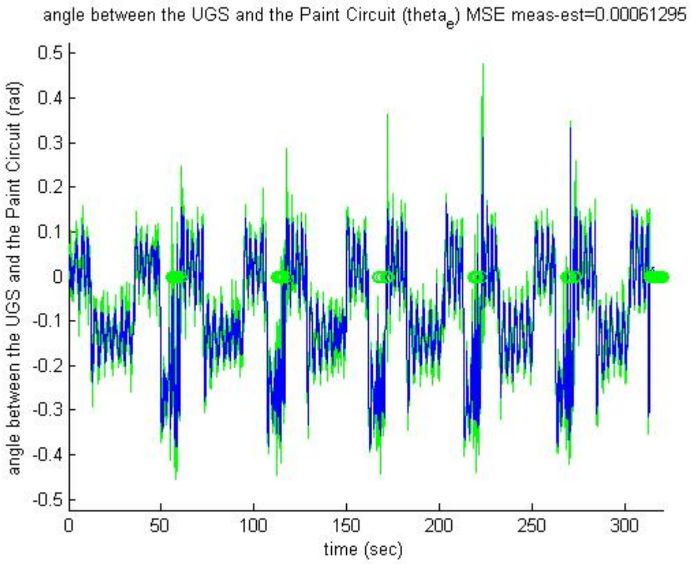


Figure 8.20: Angle $\theta_e$ between the UGS and the Paint Circuit. In green, measured angle ($\tilde{\theta}_e$). In blue estimated angle ($\hat{\theta}_e$). Lost samples are represented by a green circle.

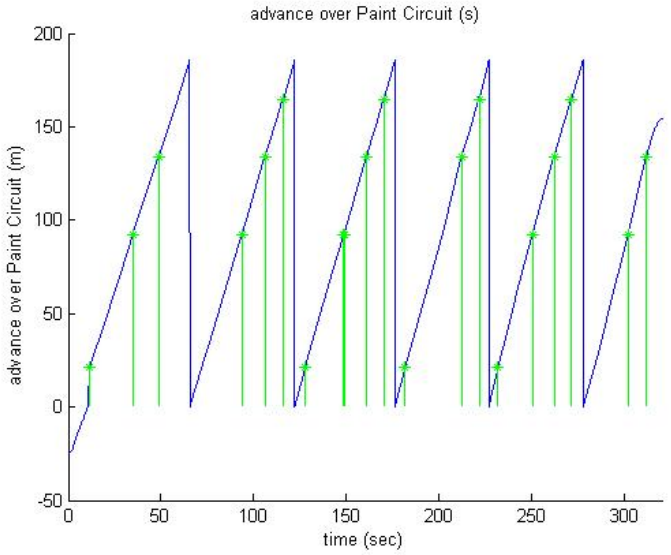


Figure 8.21: Advance over the Paint Circuit ($s$). In green, measured advance over the circuit ($\tilde{s}$); in blue, estimated advance over the circuit ($\hat{d}$).

In the **second test** (figures 8.25 to 8.31), the UGS speed is increased (see figure 8.25). It is shown that the performance of the actual High-Level Control System is increased respect to the old one (distance, angle to the path and estimated advance over the circuit are shown in figures 8.26, 8.27 and 8.28) thanks to the estimation of the curvature (figure 8.29). The steering wheel reference is shown in figure 8.30. Finally, the estimated X-Y position of the UGS can be

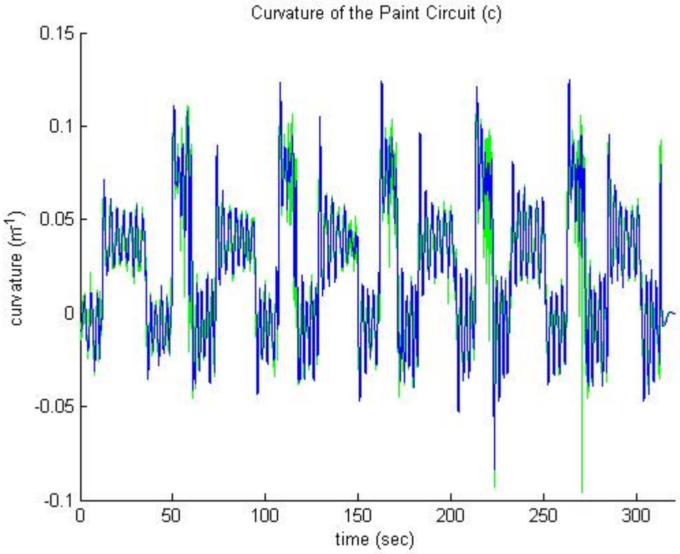


Figure 8.22: Curvature of the paint circuit ($c$). In green the theoretical curvature measured ($\tilde{c}_{teor}$). In blue, the estimated Curvature ($\hat{c}$).

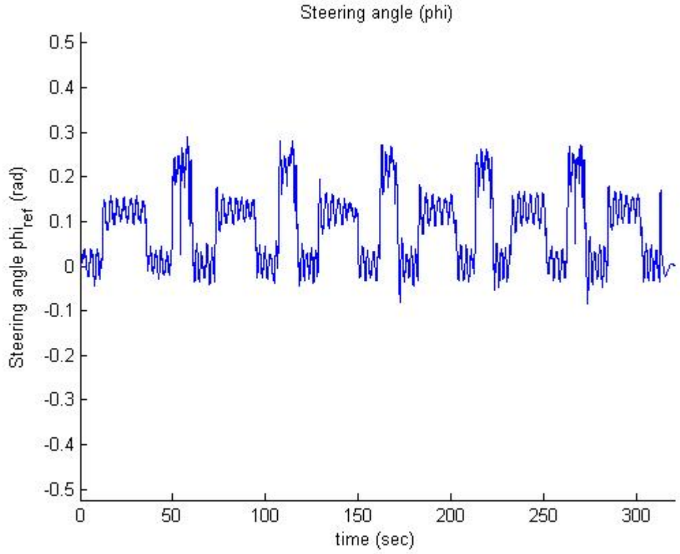


Figure 8.23: Steering wheel angle generated by the old High-Level Controller ($\phi_{ref}$).

shown in figure 8.31 but it is still bad.

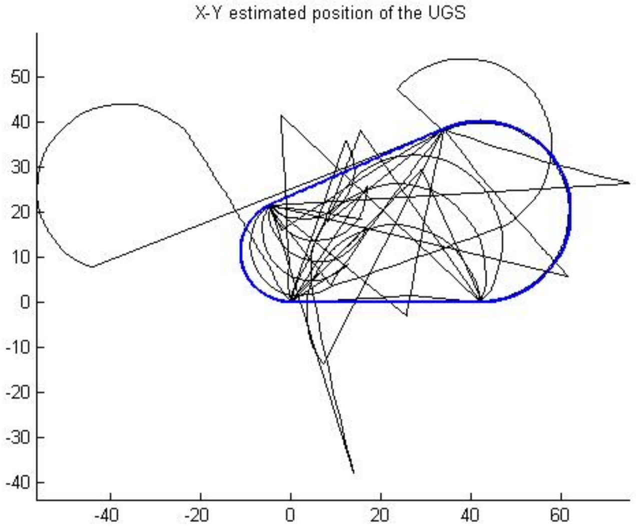


Figure 8.24: Position X-Y of the UGS. In black, the estimated position ($\hat{x}$, $\hat{y}$). In blue, the position of the real Paint Circuit.

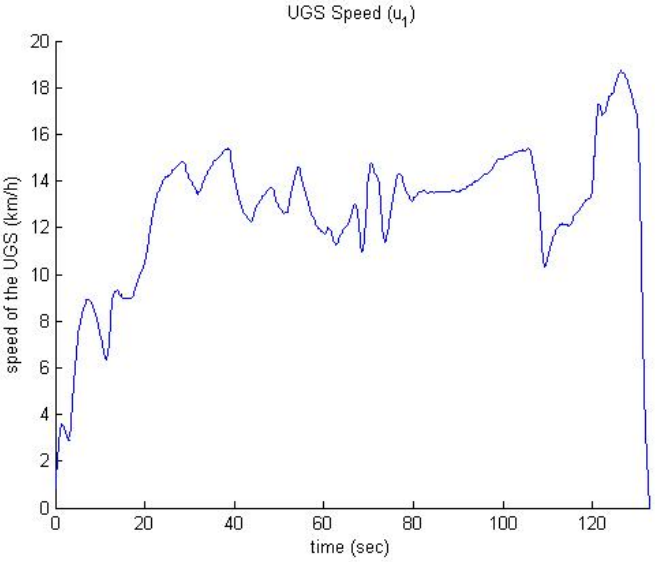


Figure 8.25: Measured Speed of the UGS ($\tilde{u}_1$).

## 8.4. Third Addition: The whole Curvature Estimator

In this section, the whole State Estimator is implemented. The last subsystem to add is the whole Curvature Estimator, that uses the Paint Circuit information to estimate the curvature of this circuit.

Thanks to the implementation of the whole State Estimator, the performance

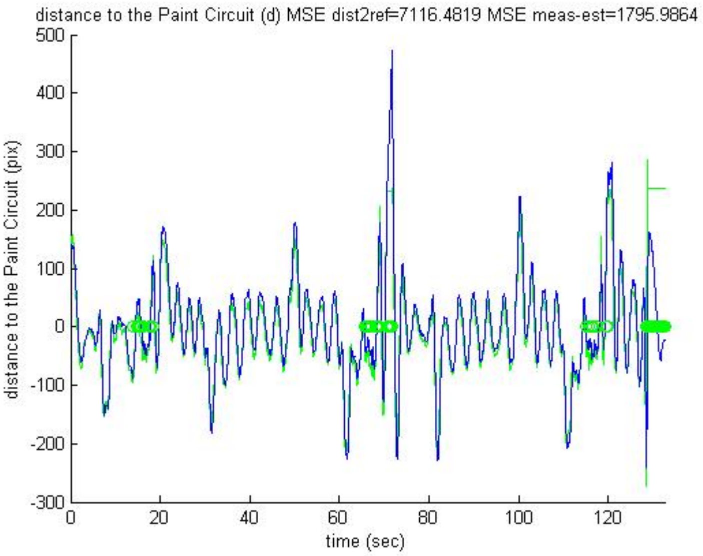


Figure 8.26: Distance of the UGS to the Paint Circuit (d). In green measured distance to the Paint Circuit ($\tilde{d}$). In blue estimated distance to the Paint Circuit ($\hat{d}$). Lost samples are represented by a green circle.

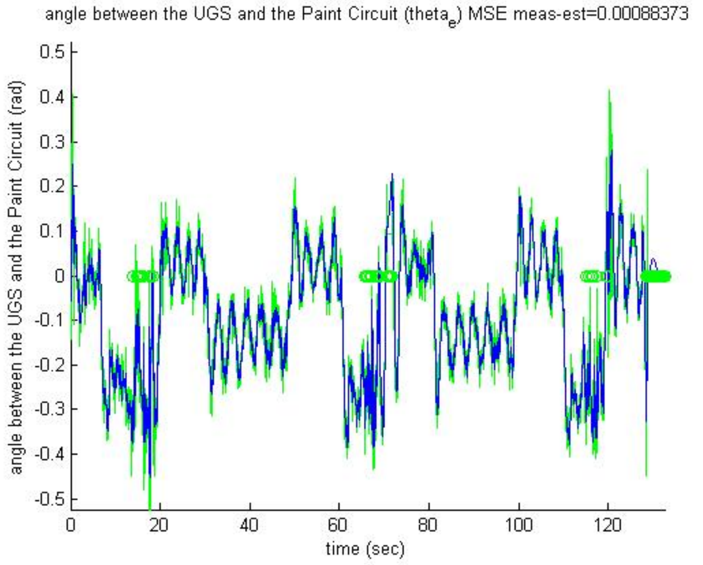


Figure 8.27: Angle $\theta_e$ between the UGS and the Paint Circuit. In green, measured angle ($\tilde{\theta}_e$). In blue estimated angle ($\hat{\theta}_e$). Lost samples are represented by a green circle.

of the proposed High-Level Control System is highly increased. In this test (figures 8.32 to 8.38) a really high speed is achieved, near $37km/h$, (see figure 8.32) without lose the circuit. Also a lots of laps could be executed correctly (see advance over the circuit in figure 8.35).

The good performance of the State Estimator was demonstrated even if the

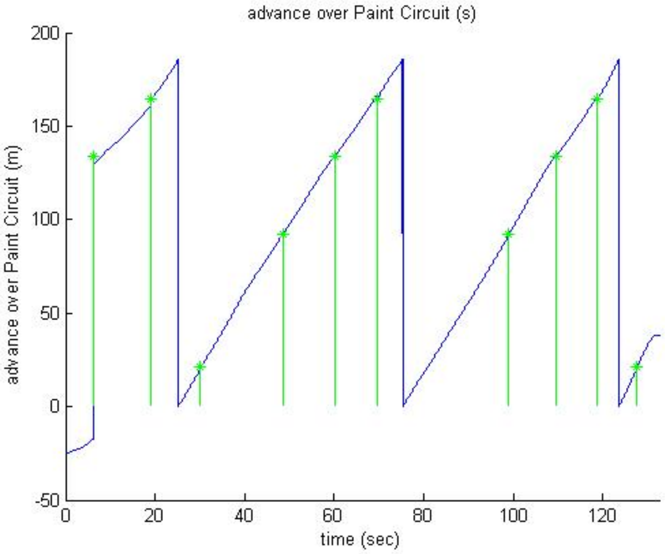


Figure 8.28: Advance over the Paint Circuit ($s$). In green, measured advance over the circuit ($\tilde{s}$); in blue, estimated advance over the circuit ($\hat{d}$).

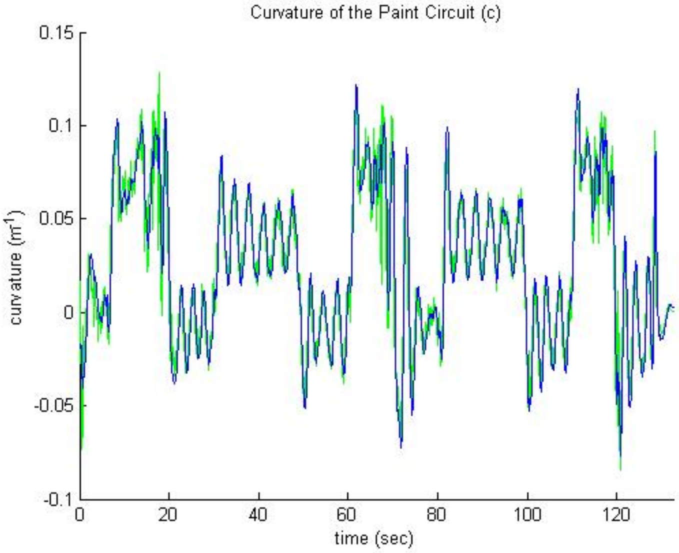


Figure 8.29: Curvature of the paint circuit ($c$). In green the theoretical curvature measured ($\tilde{c}_{teor}$). In blue, the estimated Curvature ($\hat{c}$).

initial state of the UGS is not the initial state set on the State Estimator (see figure 8.35, when a code is read and a new advance over the circuit is read, the state is updated correctly without cause any problem in the curvature estimation, figure 8.36).

In figure 8.33 the distance to the circuit is shown; in figure 8.34 the angle to the path; and in figure 8.38 the X-Y position of the UGS. Finally, figure 8.37

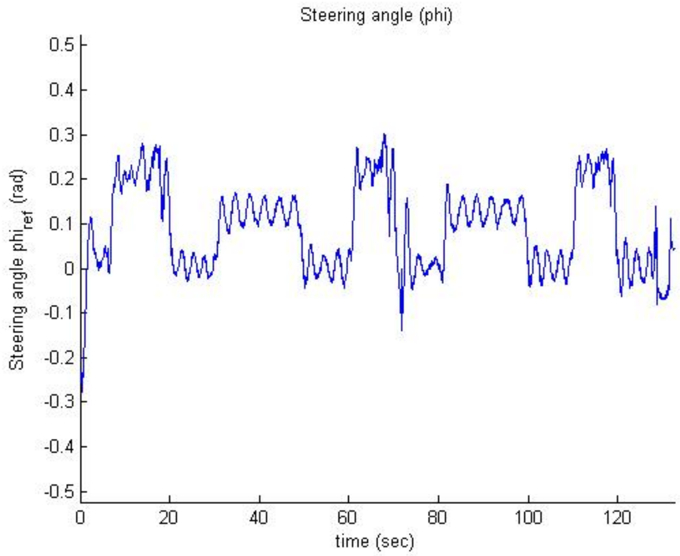


Figure 8.30: Steering wheel angle generated by the old High-Level Controller ($\phi_{ref}$).

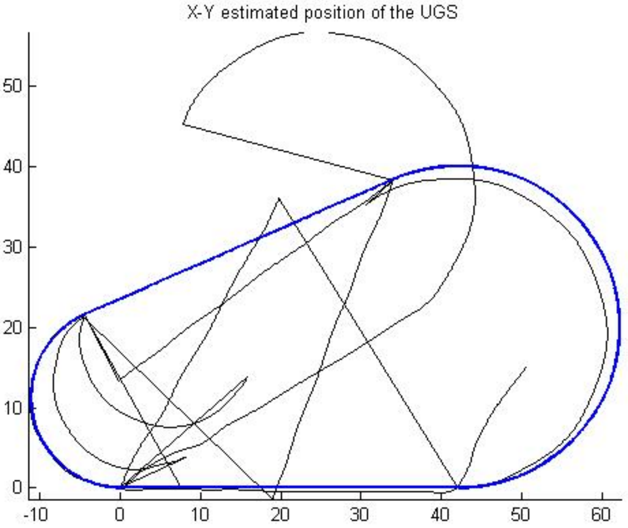


Figure 8.31: Position X-Y of the UGS. In black, the estimated position ($\hat{x}$, $\hat{y}$). In blue, the position of the real Paint Circuit.

shows the steering reference.

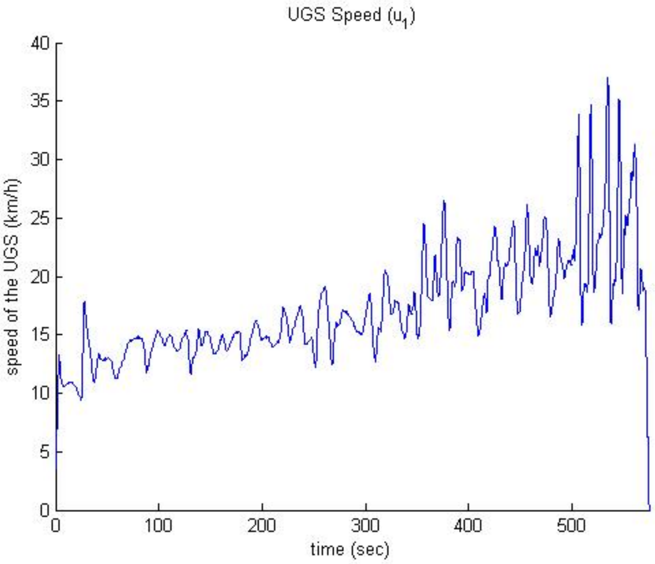


Figure 8.32: Measured Speed of the UGS ($\tilde{u}_1$).

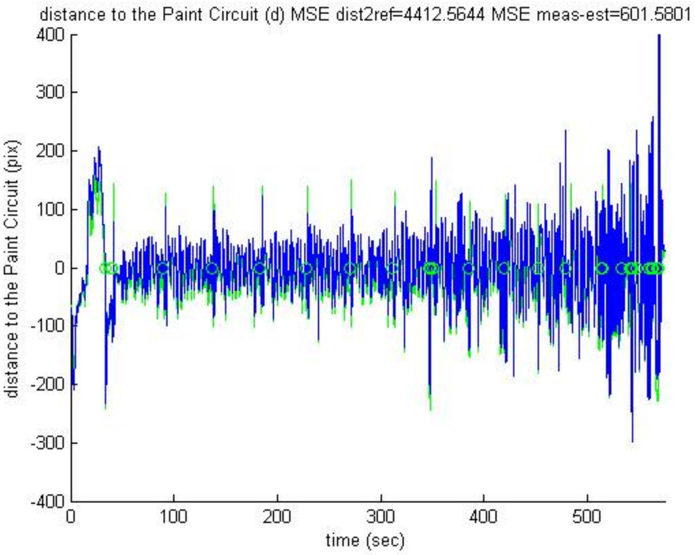


Figure 8.33: Distance of the UGS to the Paint Circuit (d). In green measured distance to the Paint Circuit ($\tilde{d}$). In blue estimated distance to the Paint Circuit ($\hat{d}$). Lost samples are represented by a green circle.

## 8.5. Last Addition: The Feed-Forward Controller

Finally, the Feed Forward Controller is added to the State Estimator to complete the proposed High-Level Control System that was tested (the Feed-Back controller could not be tested).

In the **first test** (figures 8.39 to 8.45) shows a really long race (more than 700 seconds) with a very high number of laps (16) and with variable medium-high

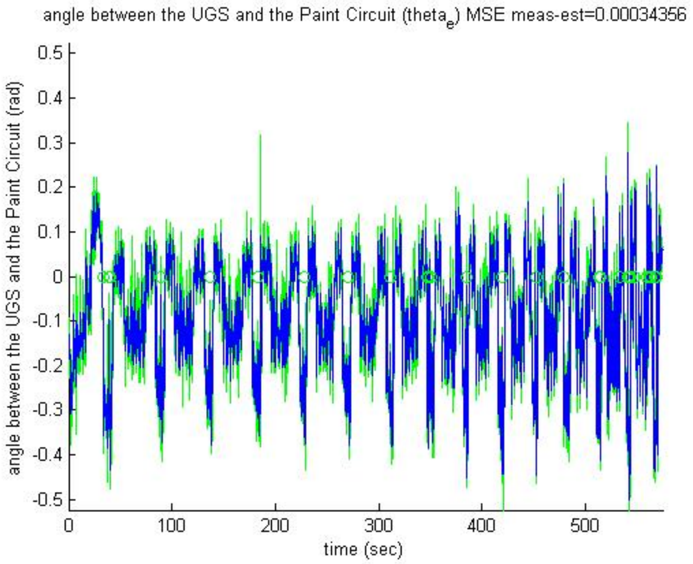


Figure 8.34: Angle $\theta_e$ between the UGS and the Paint Circuit. In green, measured angle ($\tilde{\theta}_e$). In blue estimated angle ($\hat{\theta}_e$). Lost samples are represented by a green circle.

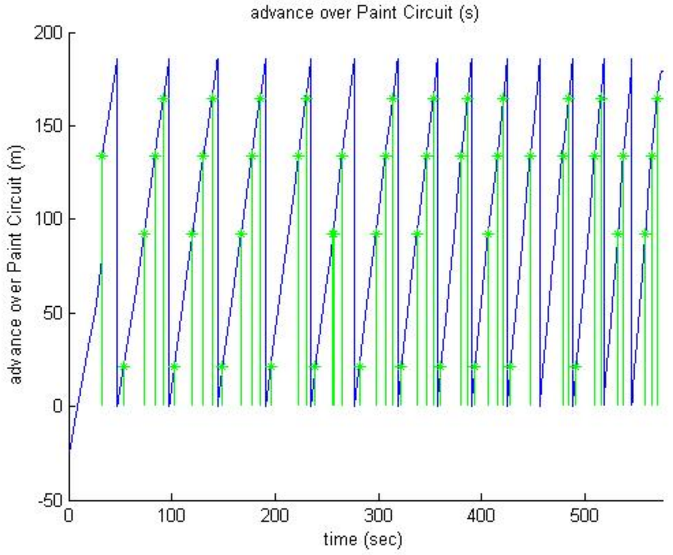


Figure 8.35: Advance over the Paint Circuit ($s$). In green, measured advance over the circuit ($\tilde{s}$); in blue, estimated advance over the circuit ($\hat{d}$).

speed (between 10 and 25 km/h). The performance of the High-Level Controller was really improved, following the path with any problem, even with lost measurement samples.

In figures 8.40, 8.41, 8.42, the distance, the angle and the advance over the circuit is shown. Figure 8.43 shows the curvature of the circuit and figure 8.45 the X-Y position of the UGS. Finally, the steering wheel reference is shown in

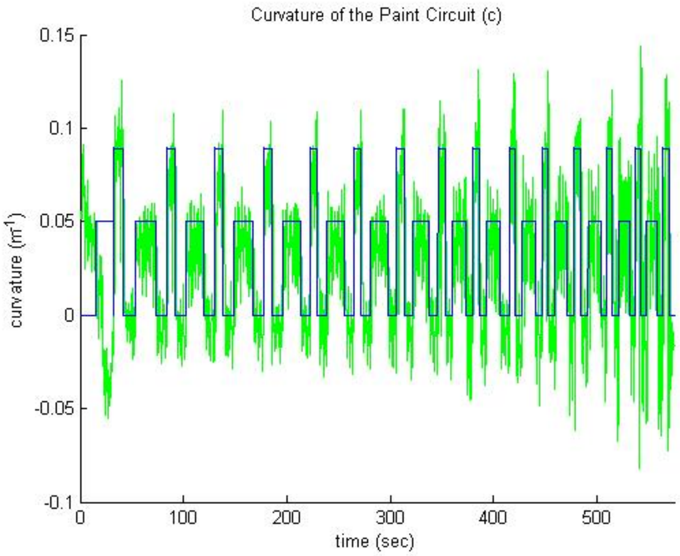


Figure 8.36: Curvature of the paint circuit ($c$). In green the theoretical curvature measured ($\tilde{c}_{teor}$). In blue, the estimated Curvature ($\hat{c}$).

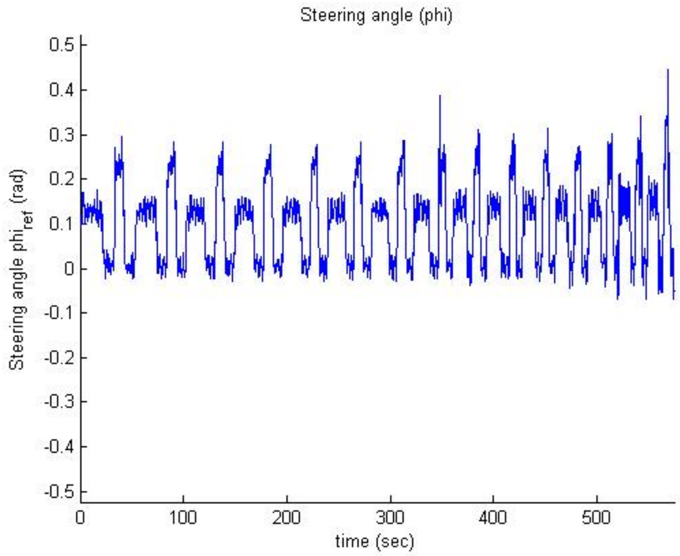


Figure 8.37: Steering wheel angle generated by the old High-Level Controller ($\phi_{ref}$).

figure 8.44.

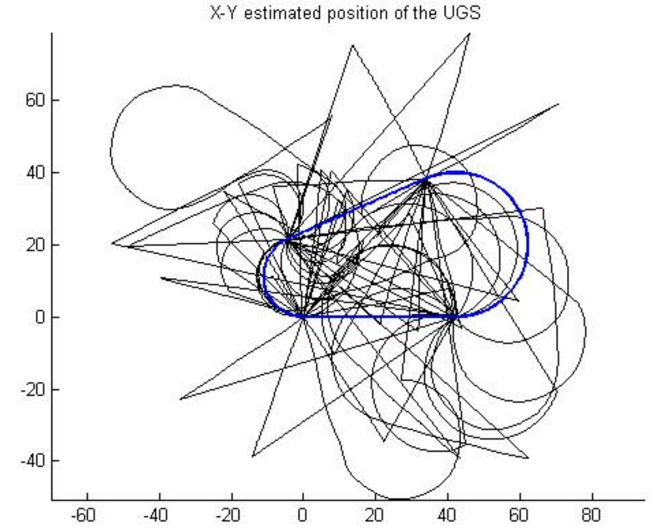


Figure 8.38: Position X-Y of the UGS. In black, the estimated position ($\hat{x}$, $\hat{y}$). In blue, the position of the real Paint Circuit.

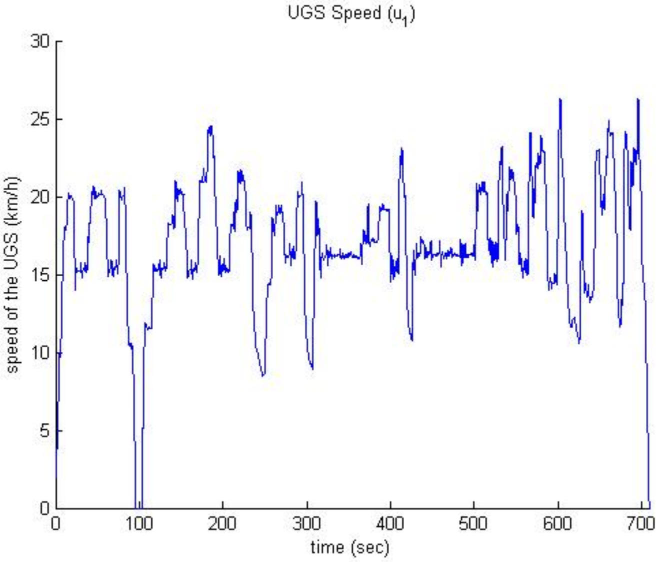


Figure 8.39: Measured Speed of the UGS ($\tilde{u}_1$).

The **second test** (figures 8.46 to 8.52) demonstrates that a high-speed, 45 km/h (figure 8.46) can achieve with this proposed High-Level Controller without the lost of the path even if some samples are lost.

In figures 8.40, 8.41, 8.42, the distance, the angle and the advance over the circuit is shown. Figure 8.43 shows the curvature of the circuit and figure 8.45 the X-Y position of the UGS. Finally, the steering wheel reference is shown in

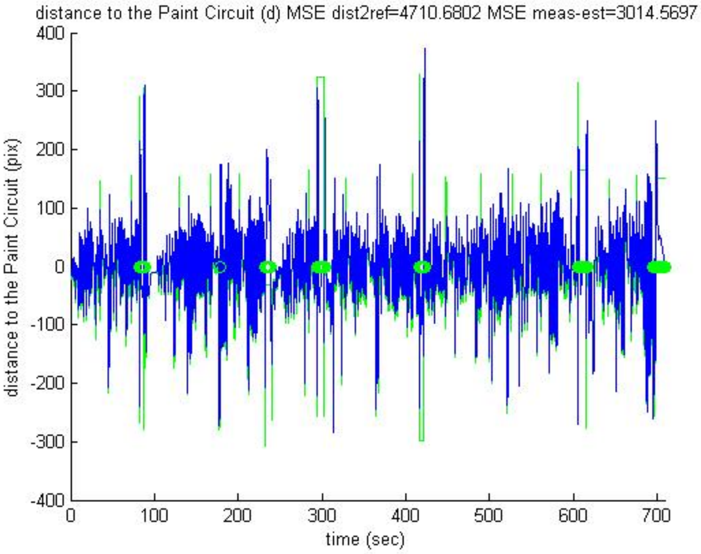


Figure 8.40: Distance of the UGS to the Paint Circuit (d). In green measured distance to the Paint Circuit ($\tilde{d}$). In blue estimated distance to the Paint Circuit ($\hat{d}$). Lost samples are represented by a green circle.

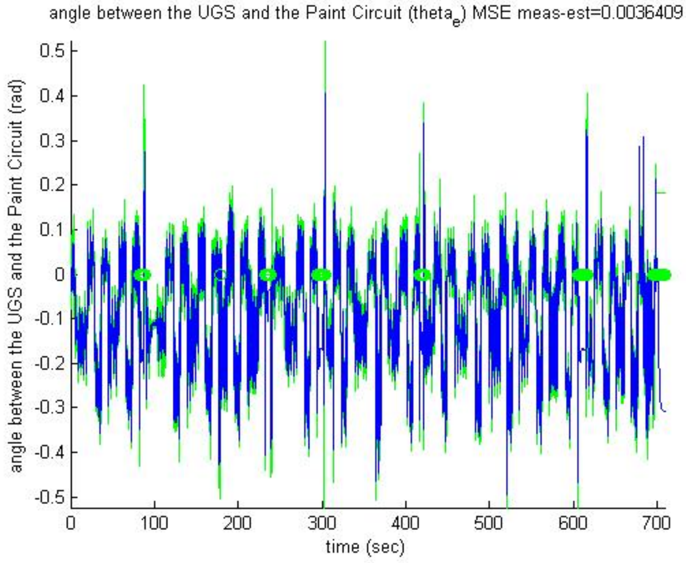


Figure 8.41: Angle $\theta_e$ between the UGS and the Paint Circuit. In green, measured angle ($\tilde{\theta}_e$). In blue estimated angle ($\hat{\theta}_e$). Lost samples are represented by a green circle.

figure 8.44.

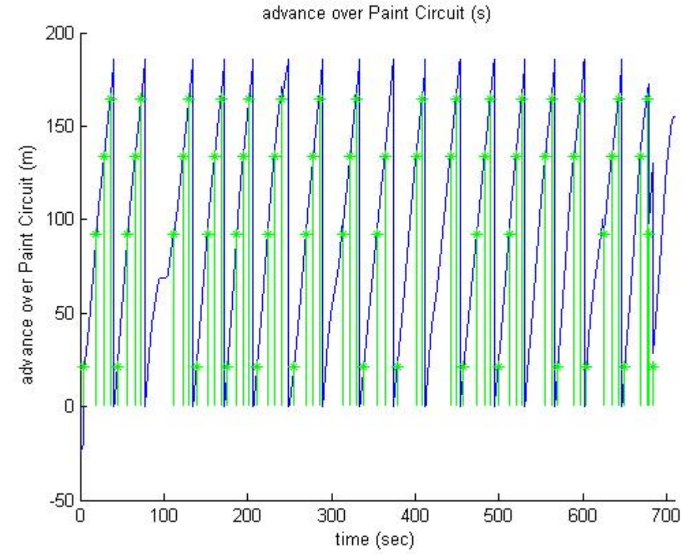


Figure 8.42: Advance over the Paint Circuit ($s$). In green, measured advance over the circuit ($\tilde{s}$); in blue, estimated advance over the circuit ($\hat{d}$).

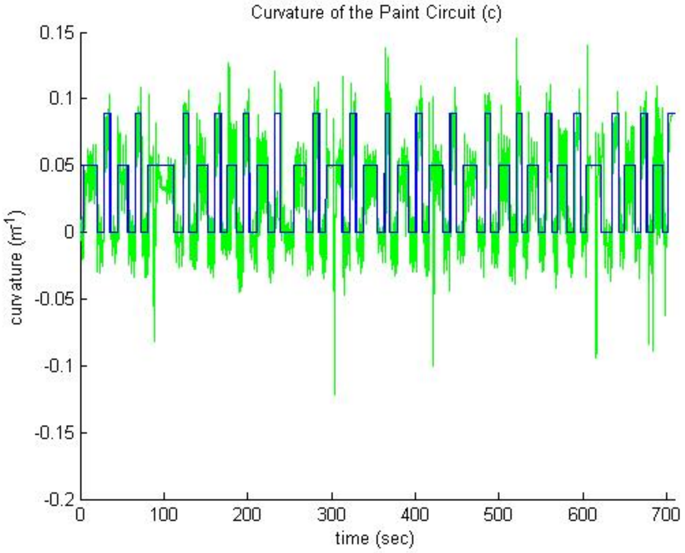


Figure 8.43: Curvature of the paint circuit ($c$). In green the theoretical curvature measured ($\tilde{c}_{teor}$). In blue, the estimated Curvature ($\hat{c}$).

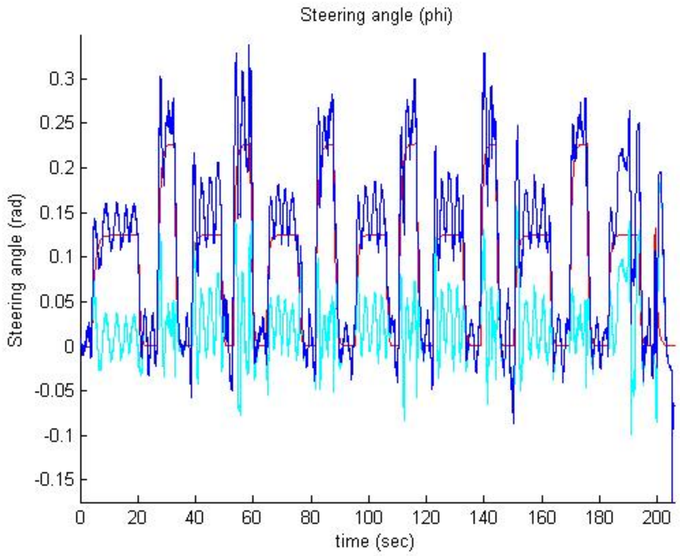


Figure 8.44: Steering wheel angle generated by the old High-Level Controller. In cyan, the output of the Feed-Back controller ($\Delta\phi$). In red, the output of the Feed-Forward Controller ($\phi_{lin}$). In blue, the addition of the Feed-Back and Feed-Forward controllers, that is the steering wheel reference ($\phi_{ref} = \Delta\phi + \phi_{lin}$).

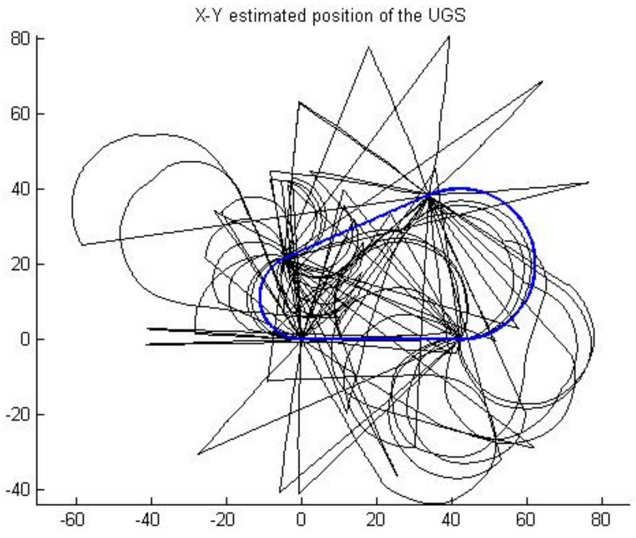


Figure 8.45: Position X-Y of the UGS. In black, the estimated position ($\hat{x}$, $\hat{y}$). In blue, the position of the real Paint Circuit.

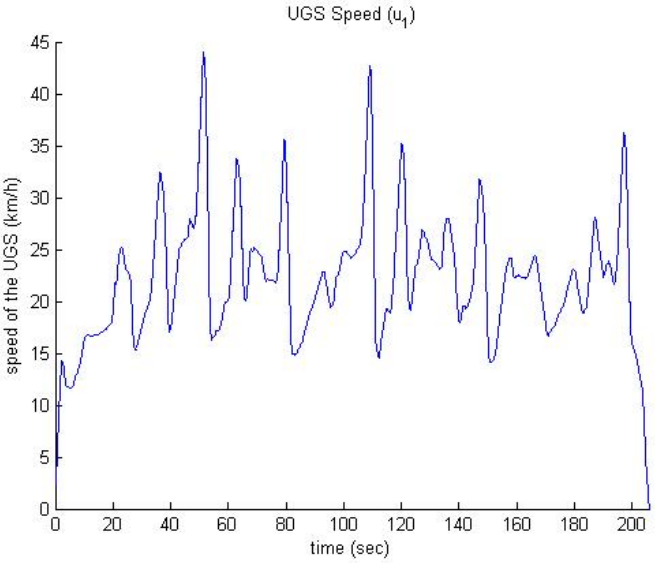


Figure 8.46: Measured Speed of the UGS ($\tilde{u}_1$).

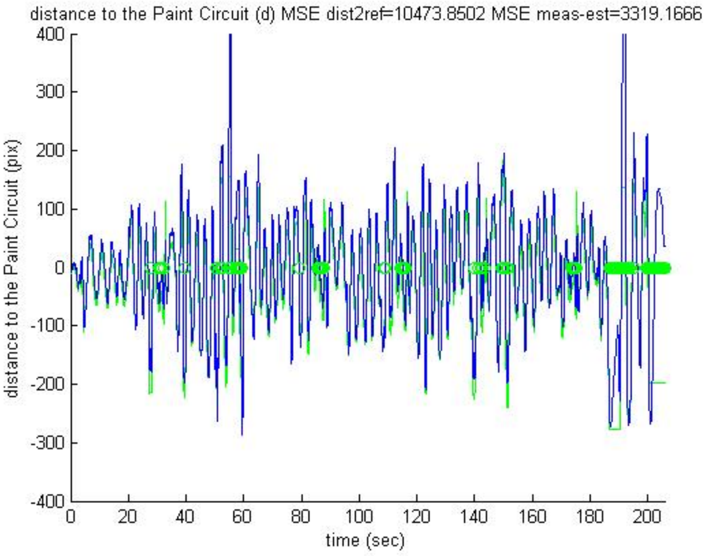


Figure 8.47: Distance of the UGS to the Paint Circuit (d). In green measured distance to the Paint Circuit ($\tilde{d}$). In blue estimated distance to the Paint Circuit ($\hat{d}$). Lost samples are represented by a green circle.

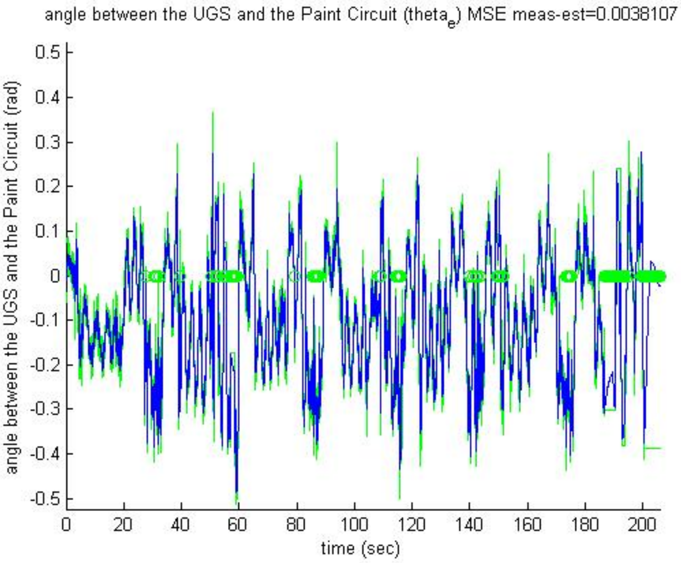


Figure 8.48: Angle $\theta_e$ between the UGS and the Paint Circuit. In green, measured angle ($\tilde{\theta}_e$). In blue estimated angle ($\hat{\theta}_e$). Lost samples are represented by a green circle.

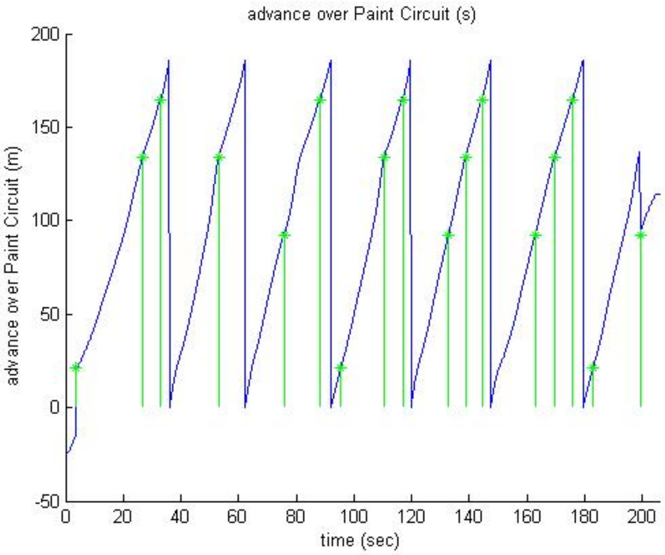


Figure 8.49: Advance over the Paint Circuit ($s$). In green, measured advance over the circuit ($\tilde{s}$); in blue, estimated advance over the circuit ($\hat{d}$).

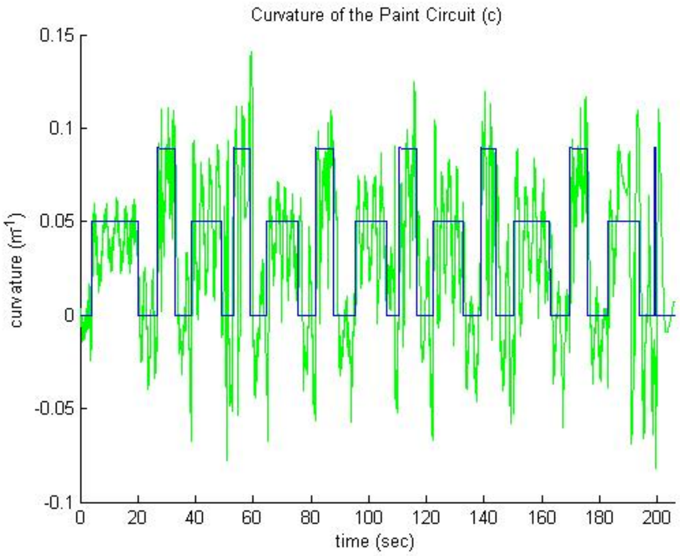


Figure 8.50: Curvature of the paint circuit ($c$). In green the theoretical curvature measured ($\tilde{c}_{teor}$). In blue, the estimated Curvature ($\hat{c}$).

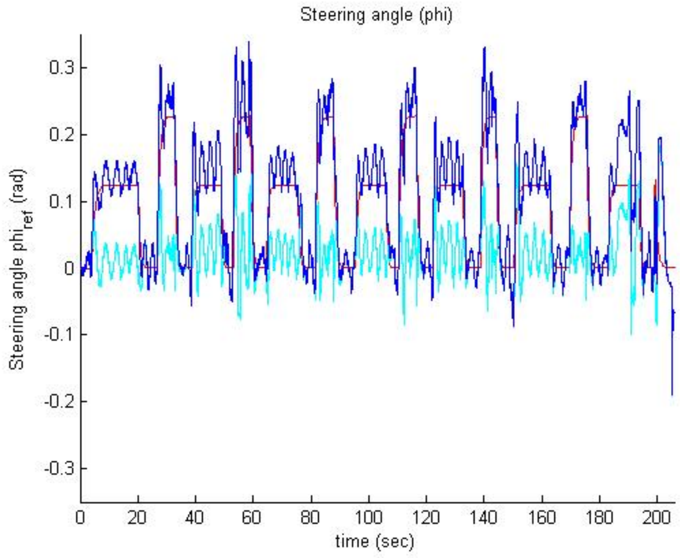


Figure 8.51: Steering wheel angle generated by the old High-Level Controller. In cyan, the output of the Feed-Back controller ($\Delta\phi$). In red, the output of the Feed-Forward Controller ($\phi_{lin}$). In blue, the addition of the Feed-Back and Feed-Forward controllers, that is the steering wheel reference ($\phi_{ref} = \Delta\phi + \phi_{lin}$).

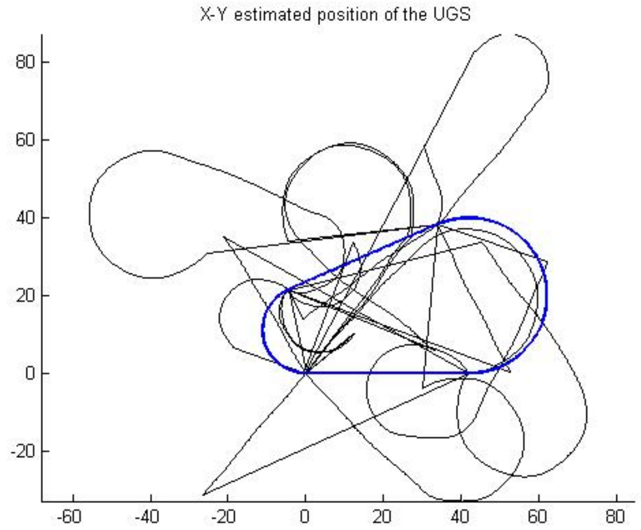


Figure 8.52: Position X-Y of the UGS. In black, the estimated position ($\hat{x}$, $\hat{y}$). In blue, the position of the real Paint Circuit.

# Chapter 9

# Conclusion and Future Work

With this Master Work, a new High-Level Control System was proposed for the path following of two car-like UGS Prototypes, one based on a Citroën C3 and the other based on a Iveco Eurocargo.

The objective was the increase of the performance of a previous controller. The goals were the increase of the speed of the UGS without the lost of the path; the increase of the safety in the path following task, minimizing the effect of the path lost and trying to avoid it; the increase of the weight of the UGS Platform; and finally the improvement of the UGS passengers.

The proposed High-Level Control System is composed by two main subsystems, a very complex State Estimator and a Controller. The State Estimator is in charged of providing to the Controller with the best values of the state of the UGS. For this really complex task a multi-system scheme is proposed, using for this task the whole information that is available. The main task of the Controller is, using the state of the UGS, generate the best reference of the steering wheel to achieve the path following problem.

This proposed High-Level Control System was tested thanks lots of simulations, demonstrating that the objective of the Master Work was theoretically satisfied.

Finally, this High-Level Control System was implemented in the UGS Platform. Because of economical problems of the contractor (SIEMENS, S.A.), the Project ends before it expected, and the whole proposed High-Level Control System never could be implemented in the two available platforms. This Control System was implemented only in the Citroën Platform, and the Feed-Back controller could not be tested. However, tests done with only the State Estimator and the Feed-Forward controller shows really good results that satisfy the objective of the Master Work.

As the Project ends before it expected, there are a lot of remaining work to do. The Feed-Back controller needs to be tested in the Citroën C3 Platform. A better Wheel Position Estimator and a Speed Estimator would be developed. Also, a Wheel dynamic compensator would be developed.

Other remaining work, is the implementation of the proposed High-Level Control System in the Iveco Platform.

Other possible future work is the improvement of the X-Y Position Estimator by using GSP to update the state more often than now. Also this X-Y Position Estimator could be used to close the loop as other UGS done.

Finally an improvement of the Very-High Level Control System Also could improve the performance of the High-Level Controller, by reporting it if any failure occurs.

# Appendix A

# Definitions and control problems

According to ISO 8373, mobile robot is defined as one robot that contains everything you need for your driving and movement (power, control and navigation system) [(Antonio Barrientos, 2007)]. It is sometimes designated with the acronym AGS (Autonomous Guided Vehicle).

Mobile robots can be classified according to the means by which to move, thereby placing ground mobile robots (called UGSs Unmanned Ground Systems), aerial mobile robots (known as UASs Unmanned Aerial Systems) or underwater mobile robot (identified as AUS Autonomous Underwater Systems).

Within the UGSs, we can distinguish land mobile robots with wheels (wheeled mobile robots) of those who have a different system of propulsion such as chains or legs [(Carelli, 2009)].

A brief classification of the ground mobile robots with wheels is by its structure, as proposed in [(Siciliano, 2008)]:

- Robot Type (3.0) or omni-mobile robot.
- Robot Type (2.0) or unicycle type robot
- Robot Type (2.1).
- Robot Type (1.1) or Ackermann type robot, which is the configuration of conventional cars for human use.
- Robot Type (1.2).

Another possible classification of mobile robots in general (and the land mobile robots with wheels in particular), is in response to restrictions on mobility [(J.M. de la Cruz, 2001)], existing:

- Robots holonomic that can reach any point in configuration space from any point, regardless of the path.
- Robots not holonomic, which can reach any point in configuration space from any point, but the trajectory is crucial. This is the case for conventional cars for human use.

The case that applies to this Master Work is a non-holonomic wheeled Ackermann type UGS.

The three generic control problems subject to multiple researches in recent times of the wheeled UGS are [(Siciliano, 2008)]:

- Path following.
- Stabilization of trajectories.
- Stabilization of fixed postures.

The control problem to be addressed in this Master Work is the path following, whose goal is, given a curve in the plane, a longitudinal velocity of the robot chassis, and a point P of interest linked to the chassis, have that point of interest follow the curve when the UGS is moving at the specified speed.

# Appendix B

# Asynchronous Extended Kalman Filter

The Kalman Filter (KF) was first proposed in (Kalman, 1960) to predict and filter linear systems. It was widely used since that day and some improvements were done, like the Extended Kalman Filter (EKF) for non-linear systems (as it is summarized in (Greg Welch, 1995)).

To better understand the EKF a summarizing is done:

We start by the knowledge of the state space model of the process:

$$\vec{x}(k+1) = \vec{f}(\vec{x}(k), \vec{u}(k)) \tag{B.1}$$

Where the $\vec{x}(k)$ is the n-dimensional state of the system at k and $\vec{v}(k)$ is the m-dimensional vector of inputs of the system at k. The Jacobians of this model are: $F_x = [\frac{\partial f_i}{\partial x_j}]_{n*n}$ and $F_u = [\frac{\partial f_i}{\partial u_k}]_{n*m}$

The observation model of the process is:

$$\vec{y}(k) = \vec{h}(\vec{x}(k)) \tag{B.2}$$

Where $\vec{y}(k)$ is the p-dimensional number of outputs (measurements) of the system. The Jacobian of this model is $H_x = [\frac{\partial h_i}{\partial x_j}]_{p*n}$

As the real system is stochastic, there are lots of random variables, and a reformulation of the problem is needed:

State Space Model of the Process:

$$\hat{\vec{x}}(k+1) = \vec{f}(\hat{\vec{x}}(k), \tilde{\vec{u}}(k)) + \vec{v}_x(k) \tag{B.3}$$

Where:

- $\hat{\vec{x}}(k)$ is the n-dimensional estimated state at k. This is a random variable that can be approached by a $\hat{\vec{x}}(k) \to N(\vec{x}(k), P(k))$. P(k) is the $n*n$-dimensional matrix of covariances of the state.
- $\tilde{\vec{u}}(k)$ are the m-dimensional measured inputs of the system. It is a random variable $\tilde{\vec{u}}(k) \to N(\vec{u}(k), Q(k))$, where Q(k) is the $m*m$-dimensional matrix of covariances of the input of the system.
- $\vec{v}_x(k)$ is a n-dimensional white noise vector of the model, that follows $\vec{v}_x(k) \to N(\vec{0}, M(k))$, been M(k), the $n*n$-dimensional matrix of covariances of the state space model.

Observation Model of the process:

$$\tilde{\vec{y}}(k) = \vec{y}(k) + \vec{w}(k) \tag{B.4}$$
$$\hat{\vec{y}}(k) = \vec{h}(\hat{\vec{x}}(k)) \tag{B.5}$$

Where:

- $\vec{w}(k)$ is a p-dimensional white noise vector of the observation model, that follows $\vec{w}(k) \to N(\vec{0}, R(k))$, been R(k) the $p*p$-dimensional matrix of covariances of the observation model.
- $\tilde{\vec{y}}(k)$ are the p-dimensional measured outputs of the system. It is a random variable $\tilde{\vec{y}}(k) \to N(\vec{y}(k), R(k))$.
- $\hat{\vec{y}}(k)$ are the p-dimensional estimated outputs of the system. It is a random variable $\hat{\vec{y}}(k) \to N(\vec{y}(k), H_x(k) * P(k) * H_x^t(k))$.

The recursive loop of the EKF is composed by:

1. Init of the loop: we have to define the initial state $\hat{\vec{x}}(0|0)$ and its initial matrix of covariances $P(0|0)$. Also all the other variances had to be set.
2. State Prediction: we have the measurements of the inputs $\tilde{\vec{u}}(k+1)$ and the last estimated state $\hat{\vec{x}}(k|k)$ with its matrix of covariances $P(k|k)$. We have to calculate the new predicted state $\hat{\vec{x}}(k+1|k)$ with the State Space Model, the new Jacobians of the State Space Model ($F_x(k+1)$ and $F_u(k+1)$) and the new covariances matrix $P(k+1|k)$.

$$\begin{aligned} \hat{\vec{x}}(k+1|k) &= \vec{f}(\hat{\vec{x}}(k|k), \tilde{\vec{u}}(k)) && \text{(B.6)} \\ P(k+1|k) &= F_x(k+1) \cdot P(k|k) \cdot F_x^t(k+1) + \\ &\quad + F_u(k+1) \cdot Q(k) \cdot F_u^t(k+1) + M(k+1) && \text{(B.7)} \end{aligned}$$

3. Outputs Prediction: using the estimated estate calculated before $\hat{\vec{x}}(k+1|k)$, we have to calculate the estimated value of the outputs of the system $\hat{\vec{y}}(k+1)$ and its Jacobian $H_x(k+1)$

$$\hat{\vec{y}}(k+1) = \vec{h}(\hat{\vec{x}}(k+1|k)) \tag{B.8}$$

4. Outputs Matching: using the measurements of the sensors of the outputs of the system $\tilde{\vec{y}}(k+1)$ we calculate the innovation of the output $\vec{v}_y(k+1)$ and its matrix of covariances $S(k+1)$ that is a $p*p$-dimensional matrix.

$$\begin{aligned} \vec{v}_y(k+1) &= \tilde{\vec{y}}(k+1) - \hat{\vec{y}}(k+1) && \text{(B.9)} \\ S(k+1) &= H_x(k+1)\cdot P(k+1|k)\cdot H_x^t(k+1) + R(k+1) && \text{(B.10)} \end{aligned}$$

5. State Correction: we calculate the Mahalanobis distance of the innovation $d(k+1) = \vec{v}_y^t \cdot S(k+1)\cdot \vec{v}_y$. If this distance is minor than a threshold, we are able to make the correction of the estimated state, calculating:

$$\begin{aligned} W(k+1) &= P(k+1|k)\cdot H_x^t(k+1)\cdot S^{-1}(k+1) && \text{(B.11)} \\ \hat{\vec{x}}(k+1|k+1) &= \hat{\vec{x}}(k+1|k) + W(k+1)\cdot \vec{v}_y(k+1) && \text{(B.12)} \\ P(k+1|k+1) &= P(k+1|k) - \\ & -W(k+1)\cdot H_x(k+1)\cdot P(k+1|k) && \text{(B.13)} \end{aligned}$$

6. Prepare the next step: You have to prepare the next step by making:

$$\begin{aligned} \hat{\vec{x}}(k|k) &= \hat{\vec{x}}(k+1|k+1) && \text{(B.14)} \\ P(k|k) &= P(k+1|k+1) && \text{(B.15)} \end{aligned}$$

For this Master Work, sometimes the outputs of the system are received at different samples of time, asynchronously. Some approaches are developed in (Arm, 2004), but for our Master Work a variation is developed:

We define the vector of outputs enabled $\vec{e}(k)$ in k that is p-dimensional vector with ones in the position of the outputs enabled and zeros in the position of the outputs non-enabled.

Now, a new observation model is defined:

$$\hat{\vec{y}}'(k+1) = diag(\vec{e}(k+1))\cdot \hat{\vec{y}}(k+1) \tag{B.16}$$

Where $diag(\vec{e}(k+1))$ is a $p*p$ matrix with every hole equal to zero but the diagonal that is the vector $\vec{e}(k+1)$.

And a new Jacobian of the observation model is also defined:

$$H_x'(k+1) = diag(\vec{e})\cdot H_x(k+1) \tag{B.17}$$

Now, some things changes:

- In the Outputs Matching step, the outputs used are $\tilde{\vec{y}}'(k+1) = diag(\vec{e}) \cdot \tilde{\vec{y}}(k+1)$ and therefore:

$$\begin{aligned}
\vec{v}_y{}'(k+1) &= \vec{\tilde{y}}'(k+1) - \vec{\hat{y}}'(k+1) && \text{(B.18)}\\
S'(k+1) &= H_x'(k+1) \cdot P(k+1|k) \cdot (H_x'(k+1))^t + \\
&\quad +R'(k+1) && \text{(B.19)}
\end{aligned}$$

  Where $R'(k+1) = diag(\vec{e}) \cdot R(k+1)$

- In the State Correction step, there are the following changes:

$$\begin{aligned}
W'(k+1) &= P(k+1|k) \cdot (H_x'(k+1))^t \cdot (S'(k+1))^+ && \text{(B.20)}\\
\hat{\vec{x}}'(k+1|k+1) &= \hat{\vec{x}}(k+1|k) + W'(k+1) \cdot \vec{v}_y'(k+1) && \text{(B.21)}\\
P'(k+1|k+1) &= P(k+1|k) - \\
&\quad -W'(k+1) \cdot H_x'(k+1) \cdot P(k+1|k) && \text{(B.22)}
\end{aligned}$$

  Where $(S'(k+1))^+$ is the pseudo-inverse of $S'(k+1)$.

# Bibliography

(2004). *SLAM Based on Kalman Filter for Multi-rate Fusion of Laser and Encoder Measurements*. Procedings of 2004 IEEE/ESJ International Conference on Intelligent Robots and Systems.

(2010). Google inc. http://googleblog.blogspot.com.es/2010/10/what-were-driving-at.html.

(2012). Darpa challenge. http://www.darpagrandchallenge.com.

(2012). Programa autopia. http://www.car.upm-csic.es/autopia/.

(2012). Sartre project. http://www.sartreproject.eu/en/Sidor/default.aspx.

(2012). Vislab, university of parma. http://viac.vislab.it/.

Antonio Barrientos, e. a. (2007). *Fundamentos de Robotica*. McGrawHill.

Bertozzi, M., Broggi, A., Conte, G., Fascioli, A., and Fascioli, R. (1998). Vision-based automated vehicle guidance: the experience of the argo vehicle.

Broggi, A., Bertozzi, M., and Fascioli, A. (2000). Architectural issues on vision-based automatic vehicle guidance: the experience of the argo project. *Real-Time Imaging*, 6(4):313–324.

Carelli, R. (2009). Conferencia: Control de robots moviles. Conferencia en la ETSII-UPM.

de la Salud, O. M. (2009). *Informe sobre la Situacion Mundial de la Seguridad Vial: es hora de pasar a la accion*. Organizacion Mundial de la Salud (OMS), Switzerland, 1st edition.

de Trafico, D. G. http://www.dgt.es/.

Francisco Aparicio, e. a. (2008). *Ingenieria del Transporte*. CIE DOSAT 2000.

Greg Welch, G. B. (1995). An introduction to the kalman filter. Technical report, University of North Carolina at Chapel Hill.

I. Mondragon, M. Olivares-Mendez, P. C. C. M. and Mejias, L. (2010). Unmanned aerial vehicles uavs attitude, height, motion estimation and control using visual systems. *Autonomous Robots*, 29:17 – 34.

J. L. Sanchez-Lopez, P. Campoy, M. A. O.-M. I. M.-B. D. G.-G. (2011). Adaptive control system based on linear control theory for the path-following problem of a car-like mobile robot. In *IFAC PID'12*.

J.M. de la Cruz, A.M. Sanchez Perez, F. R. H. (2001). *Mecanica Analitica*. Seccion de Publicaciones de la ETSII-UPM.

Kalman, R. E. (1960). A new approach to linear filtering and prediction problems. *Transactions of the ASME - Journal of Basic Engineering*, 82(D):35–45.

M. Montemerlo, J. Becker, S. B. H. D. D. D. S. E.-D. H. T. H. G. H. B. H. D. J. S. K. D. L. A. L. J. L. J. M. D. O. J. P. I. P. A. P. M. P. G. S. D. S. A. V. and Thrun, S. (2008). Junior: The stanford entry in the urban challenge. *Journal of Field Robotics*.

R. Behringer, N. M. (1998). Autonomous road vehicle guidance from autobahnen to narrow curves. In Transactions, I., editor, *Robotics and Automation*, volume 14, page 810 : 815.

S. Thrun, M. Montemerlo, H. D. D. S. A. A. J. D.-P. F. J. G. M. H. G. H. K. L. C. O. M. P. V. P. P. S. S. S. C. D. L.-E. J. C. K. C. M. C. R. J. v. N. E. J. P. A. G. B. B. D. S. E. A. K. A. N. and Mahoney, P. (2006). Winning the darpa grand challenge. *Journal of Field Robotics*.

Shladover, S. (2007). Path at 20 history and major milestones. In Transactions, I., editor, *Intelligent Transportation Systems*, volume 8.

Siciliano, K. (2008). *Handbook of Robotics*. Springer.

y Empleo, P. (2010). El tiempo de desplazamiento al trabajo en espana. http://paroyempleo.blogspot.com/2010/08/el-tiempo-de-desplazamiento-al-trabajo.html.