Robust and adaptive discrete-time control for autonomous off-road vehicles

Ph.D. Thesis

Fernando Benjamín Fernández Guzmán

Supervised by

Dr. José Antonio Cerrada Somolinos

Dr. Pedro Javier Herrera Caro

Programa de Doctorado en Ingeniería de Sistemas y Control

Dpto. Ingeniería de Software y Sistemas Informáticos

Escuela Técnica Superior de Ingeniería Informática

Universidad Nacional de Educación a Distancia

Madrid 2018

Robust and adaptive discrete-time control for autonomous off-road vehicles

Dissertation submitted to obtain the Ph.D. Degree by

Fernando Benjamín Fernández Guzmán

Supervised by

Dr. José Antonio Cerrada Somolinos

Dr. Pedro Javier Herrera Caro

Programa de Doctorado en Ingeniería de Sistemas y Control

Dpto. Ingeniería de Software y Sistemas Informáticos

Escuela Técnica Superior de Ingeniería Informática

Universidad Nacional de Educación a Distancia

Madrid 2018

To my wife and daughter.

Declaration

I hereby declare that except where specific reference is made to the work of others, the contents of this dissertation are original and have not been submitted in whole or in part for consideration for any other degree or qualification in this, or any other university. This dissertation is my own work and contains nothing which is the outcome of work done in collaboration with others, except as specified in the text and Acknowledgements.

Fernando Benjamín Fernández Guzmán

September 2018

Acknowledgements

I would like to thank AGCO for the provision of the robot and especially Thiemo Buchner and Benno Pichlmaier for their support in this research.

Special thanks to my supervisors for their time, patience, support, revisions, advises and motivation in this work. Special thanks also to Dr. Joseph Gross for his time and revisions.

This work has been developed with the help of the research projects DPI2013-44776-R and DPI2016-77677-P of the MICINN. It also belongs to the activities carried out within the framework of the research network CAM RoboCity2030 S2013/MIT-2748 of the Comunidad de Madrid.

Abstract

When designing a controller for autonomous off-road vehicles there are two main issues to consider apart from the vehicle dynamics: velocity and terrain irregularities. Those issues make the system to be time-varying, and whereas the first one is measurable, the effects of the terrain irregularities over the dynamic of the vehicle, can be very difficult to determine. Solutions to cover these issues could be very complex, time consuming and impractical to be implemented in an embedded system with limited resources. There are proposed solutions in the literature for the usage of Model Reference Adaptive Control (MRAC). Nevertheless, there is still room for improvement in terms of robustness, scope of the controller and complexity.

The research presented in this thesis consists of two different approaches: a *cascade self-tuning regulator* and a *robust self-tuning regulator*. The first one consists of two nested loops, the inner one contains the regulator for the yaw rate dynamics, and the outer one is a PID controller for the lateral position, managed by a gain scheduler. The second approach is only one robust controller that covers directly the lateral position. Both self-tuning regulators use an RST digital controller. The algorithm used in the first regulator is a Minimum Degree Pole Placement (MDPP), and for the second approach, a robust pole placement with regulation dynamics is used. Additionally, a gain scheduler is implemented to adjust the parameters of the regulation dynamics according to the measured forward speed. In both approaches, an online estimation of the yaw rate dynamics is proposed, using two

possible algorithms: *Recursive least-squares* and *Online gradient*.

The controllers proposed in this work were validated in simulated and real environments, and compared against a well known optimal controller used as a benchmark, with very satisfactory results. Also, they can be used for vehicles with different steering systems such as Skid-steering and Ackermann-steering. Furthermore, they are linear and have lightweight capabilities in terms of implementation in an ECU with limited resources, reducing by this the gap of the transition from research prototypes to commercial machines.

Keywords: Trajectory tracking; Path following; Robotics; Off-road vehicles; Autonomous vehicles; Guidance control; Robust digital control; RST Controller; Digital pole placement; Agricultural robotics; Skid-steering robot; Self-tuning regulator; Tractor-Implement system.

Resumen

A la hora de diseñar un controlador para vehículos todo terreno no tripulados existen dos aspectos a tener en cuenta, además de la propia dinámica del vehículo: la velocidad y las irregularidades del terreno. Estos aspectos hacen que el comportamiento del sistema varíe en el tiempo y, mientras que el primer aspecto puede ser medible, determinar cómo afectan las irregularidades del terreno a la dinámica del vehículo puede ser extremadamente difícil. Las soluciones que cubren esta problemática pueden ser muy complejas y requerir gran cantidad de recursos tanto de tiempo como de conocimientos, haciendo que su implementación en sistemas embebidos sea poco práctica. La literatura propone soluciones basadas en la implementación de controladores adaptativos con modelo de referencia (MRAC, por sus siglas en inglés). Sin embargo, aun existen aspectos a mejorar en términos de robustez, alcance del controlador y complejidad de diseño e implementación.

El trabajo de investigación desarrollado en esta tesis consiste en el análisis e implementación de dos tipos de controladores: un *regulador auto ajustable en cascada* y un *regulador auto ajustable robusto*. El primero consiste en dos lazos de control anidados, donde el lazo interno contiene el regulador auto ajustable para la velocidad de viraje y el lazo externo contiene un controlador tipo PID para la posición lateral del vehículo, el cual es gestionado por un planificador de ganancias para ajustar los parámetros del controlador según la velocidad del vehículo. El segundo controlador es un regulador auto ajustable robusto que controla directamente la posición lateral del vehículo. Ambos reguladores usan un controlador digital tipo RST, en el primero el algoritmo usado se basa en posicionamiento de polos de grado mínimo (MDPP, por sus siglas en inglés), mientras que en el segundo se basa en un posicionamiento de polos robusto basado en un modelo de regulación de segundo orden. Adicionalmente, este último contiene un

planificador de ganancias que ajusta los parámetros del modelo de regulación según la velocidad medida del vehículo. Para la identificación en tiempo real de la dinámica de viraje, en ambos controladores se han propuesto dos tipos de algoritmos: *Recursividad con mínimos-cuadrados* y *Gradiente en tiempo real*.

Las controladores propuestos en esta tesis fueron validados en entornos simulados y reales, y comparados favorablemente frente a un controlador de tipo óptimo ampliamente conocido. Además de utilizarse en robots agrícolas con dirección deslizante, pueden aplicarse en vehículos con diferente tipo de sistema de dirección, al ser los controladores lineales y "ligeros" en términos de programación, lo que favorece su implementación en sistemas embebidos pequeños, reduciendo así la brecha de transición de prototipos de investigación a vehículos comerciales.

Palabras clave: Robótica; Seguimiento de caminos; Seguimiento de trayectoria; Vehículos todoterreno; Vehículos autónomos; Control de navegación; Navegación automática; Control digital robusto; Controlador RST; Localización digital de polos; Robótica en la agricultura; Robots con dirección deslizante; Regulador auto-ajustable; Sistema tractor-apero.

Contents

List of Figures

List of Tables

List of Symbols

Acronyms / Abbreviations

ALV Autonomous Land Vehicle

ARMAX Auto-Regressive Moving Average with eXogenous input

ARX Auto-Regressive with eXogenous input

CCPP Continuous Curvature Path Planning

CPP Coverage Path Planning

CVA Canonical Variate Analysis

DARPA Defense Advanced Research Projects Agency

DOF Degrees of Freedom

ECU Electronic Control Unit

EIV Errors in variables

EKF Extended Kalman Filter

ELS Extended Least Squares

GL Generalized Least Squares

GNSS Global Navigation Satellite System

GPS Global Positioning System

IVAM Instrumental Variable with Auxiliary Model

LQG Linear Quadratic Gaussian

LQR Linear Quadratic Regulator

MDPP Minimum Degree Pole Placement

MDPP Minimum-degree Pole Placement

MIMO Multiple Input Multiple Output

MIT Massachusetts Institute of Technology

MOESP MIMO Output-Error State Space model identification

MPC Model Predictive Controller

MRAC Model Reference Adaptive Control

MRAS Model Reference Adaptive Systems

N4SID Numerical State Space Subspace System Identification

OEAFO Output Error with Adaptive Filtered Observations

OEEPM Output Error with Extended prediction Model

OEFC Output Error with Fixed Compensator

OEFO Output Error with Filtered Observations

PEM Prediction Error Methods

PID Proportional Integral Derivative

RLS Recursive Least Squares

RML Recursive Maximum Likelihood

RTK Real Time Kinematic

SCPP Simple Curvature Path Planning

SISO Simple Input Simple Output

STR Self Tuning Regulator

TSP Travelling Salesman Problem

WLAN Wireless Local Area Network

Controller Symbols

$\hat{\theta}$ Vector of process parameters

$\hat{x}$ State estimated variable

τ Time delay

φ Vector of past measurements of inputs and outputs

F State feedback gain

L State estimate gain

Q Tuning parameter of an optimal controller

R Parameter of control of a RST controller

R Tuning parameter of an optimal controller

r Reference output

S Parameter of control of a RST controller

S_{up} Input sensitivity function

S_{yb} Noise sensitivity function

S_{yp} Output sensitivity function

S_{yv} Disturbance sensitivity function

T Parameter of control of a RST controller

T_s Sampling time

u Control signal

x State variable

y System output

Subscripts

$_e$ Error

$_k$ Carrot point

$_{ss}$ Steady state

$_{up}$ Input to disturbances

$_{yb}$ Output to noise

$_{yp}$ Output to disturbances

$_{yv}$ Output to disturbances on the input

Vehicle Symbols

α_f Front slip angle

α_h Hitch slip angle

α_r Rear slip angle

β Orientation

δ Steering angle

$\dot{\beta}$ Angular speed

ν Course angle

$C_{\alpha f}$ Cornering stiffness at the front

$C_{\alpha h}$ Cornering stiffness at the hitch

$C_{\alpha r}$ Cornering stiffness at the rear

L_d Look ahead distance

R Radius

V_x Forward speed

V_y Lateral speed

x Position in x on a Cartesian coordinate system

y position in y on a Cartesian coordinate system

Chapter 1

Introduction

With an expected population growth of 30%, some sources forecast an increase of up to 9.6 billion inhabitants by 2050 (U.Nations, 2013). To be able to feed so many people, the agricultural processes play an important role since the load on the planet to produce enough food for everybody will increase as well. This means that our need for food is jeopardizing our planet and consequently, and ironically, our existence at the same time (Foley, 2017). For instance, by working the field with heavy machinery we increase the soil compaction and this will turn the field into a sterile area in the long run (Shah et al., 2017). There has been an increasing research interest on agricultural robotics and automation to help improve all different kind of agricultural processes and deal with the aforementioned and other agricultural problems. The goal is to use resources such as tools, machinery, water, fertilizer, working area, seeds, etc., to more efficiently increase yield without having to increase the working area. This is especially very important since the area of the planet that can be used for agricultural tasks is limited. Regarding field machinery, we can basically find two approaches to help solve some of those issues: The first one is to automate heavy machinery such as tractors, harvesters and working implements to make them even drive autonomously (Moorehead et al., 2012); the second one is to change and specialize the technology according to our needs and to use small machinery to complete these tasks (Auat Cheein and

Carelli, 2013; Jinlin and Liming, 2010; Kohanbash et al., 2012). Regarding the first approach, precision farming includes different solutions such as the dosage of fertilizers and pesticides by the use of e.g. section and rate control (Luck et al., 2010; Sharda et al., 2013). Regarding the second approach, there are different projects and concepts like the one presented by Blender et al. (2016), where the energy and seeds are rationalized.

The common ground in both approaches, and its relation to this text, is the need for autonomous driving. Agricultural robotics is a topic that has been gaining more and more attention and the necessity of developing different kinds of off-road autonomous vehicles for agricultural tasks has increased in the last couple of years.

1.1 Background

The autonomous land vehicle problem has been researched for decades and one of the first interesting results was made be MacAdam (1980, 1981). Later in the same decade, Carnegie Mellon University together with DARPA (Defense Advanced Research Projects Agency) and other institutions founded the Autonomous Land Vehicle project (ALV) (Kanade et al., 1986; Wallace et al., 1985), with the aim of building intelligent, vision-based mobile robots. The first results were brought to the field with ALVINN (An Autonomous Land Vehicle In a Neural Network) (Pomerleau, 1989).

In the same decade in the area of agriculture, vision guidance was also researched by Gerrish and Surbrook (1984); Gerrish et al. (1986) and Searcy and Reid (1986). And what seems to be the first adaptive approach for an agricultural tractor, was presented by Noh (1990), Noh and Erbach (1993) with a very good analysis of the state-of-the-art technologies and methods of that time. Later, parallel to the visual-based guidance results presented by Gerrish et al. (1997) and Ollis and Stentz (1996) from Carnegie

Mellon University, the advances in the Global Positioning System (GPS), brought also results in the area of autonomous GPS-based land machines (Michael O'Connor et al., 1996; O'Cornnor, 1997; Stombaugh et al., 1998).

Nowadays, autonomous driving vehicles have been gaining more and more interest, and a lot of research has been performed in the area of agricultural, and off-road machines (Bayar et al., 2016a; Bergerman et al., 2015; Blender et al., 2016; Mousazadeh, 2013; Vroegindeweij et al., 2018; Zhang et al., 2018). Also, even though the use of GNSS (Global Navigation Satellite System) for autonomous guidance is state of the art in modern land and agricultural machines, improvements in the design and implementation, as well as in the robustness of path following controllers, is still a challenge in all kind of ground vehicles (Amer et al., 2017).

1.2 Problem formulation

There are two main factors that can affect the dynamics of a land vehicle during its operation: speed and contact with the soil. Although these factors can be accounted for ground vehicles in general, off-road vehicles will be more affected by them. For instance, the changes in the soil for urban vehicles are less dramatic compared to agricultural vehicles, which should be able to drive through all different kind of terrains. An agricultural vehicle will be facing loose, slippery and wet earth which makes the driving a very difficult task. If that is not enough, it will carry different kind of implements or tools adding different dynamics to the vehicle, and changing even their weight on the go. Furthermore, the precision required to drive autonomously through the lanes, should be very high, so the vehicle does not destroy the crops. All this requires high accuracy controllers which should be able to adapt to the soil uncertainties and to the changes in the weight, the speed and the form of the vehicle

caused by the use of different implements. This complexity will affect not only the know-how needed to implement a solution, but also the processing power required in real-time, to be carried by the machines - not to mention the time needed to implement such a controller.

1.3 Motivation

As mentioned, one of the biggest challenges of automating an off-road vehicle is that the dynamics of the system are changing constantly. This is due to different factors such as driving in different soils, changes in the soil itself (which can not be predicted), changes in the driving speed, and changes in the load that the vehicle carries itself or through an implement. This makes the design of a controller a very difficult task, since finding a set of control parameters for every single situation is nearly impossible. A good tuned controller will need a lot of time and resources for its implementation in an off-road vehicle. For instance, tuning a classical PID controller for a tractor with an implement, will imply to perform the tuning in all different possible soils, covering all possible velocities and with all different combinations of front and rear implements for all the different situations. And all that has to be performed for the different types of vehicles with different steering systems such as Ackermann, 4-Wheeled and Skid-steering. Furthermore, the computational resources available in such vehicles are limited, and complex systems with heavy calculations are not always possible, especially when they have to be performed online and in real-time.

Some solutions that specifically address this problem, have been proposed for the case of a tractor with an implement (Derrick and Bevly, 2008; Derrick et al., 2008; Derrick and Bevly, 2009; Gartley and Bevly, 2008). Nevertheless, the solution here for the changes in the speed is not ideal, nor its implementation in dif-

ferent types of steering systems, and it is also time consuming. Controllers for specific types of vehicles can be found in the literature, but they are either complex or need a lot of computing resources for real-time calculations (see Ch. 2). Therefore, the motivation of this research is to find a controller that is capable of covering for changes in the vehicle dynamics, terrain uncertainties, which is also lightweight, easy to implement in an embedded system with limited resources, and finally, that is flexible enough to be implemented in different types of steering systems with small time investment for tuning.

1.4 Objective of this work

The general objective of this work is to find a controller, able to cover for changes in the dynamics of an autonomous off-road vehicle. Specifically a digital controller should be addressed to make its implementation in a real-time embedded system easier. Also, to limit the resources needed by the processor, the controller should be lightweight and no heavy calculations should be performed. The main challenges that should be covered by the controller, are the changes in speed, weight and terrain irregularities. The controller should also be constructed in such a general way, that different kind of steering systems can be controlled by the same approach. Finally, the implementation and tuning of the controller should be easy enough to save time and resources.

1.5 Methodology

At the beginning of this work, a literature review is performed in order to find available solutions. With the literature review, different control approaches were proposed. Since the implementation and testing of the approaches require a lot of resources in terms of machinery and field tests, different types of simulations were performed using Matlab, C and C++ libraries. With the

help of the simulations, some preliminary results were addressed in order to then implement the best approaches in a real vehicle. Different tests were performed in different terrains to be able to compare the proposed solutions. A preliminary comparison of results was performed visually in order to get a feeling of the behaviors of the vehicle according to the soil. Finally, the errors of the lateral position were calculated in order to compare the results and conclude which approach dropped the best performance.

1.6 Contribution of this work

The main contribution of this work is the design of a digital, robust and adaptive path following controller, for autonomous off-road and land vehicles. This approach focuses, on the one hand, on adaptive control to cover for changes in the dynamics of the vehicle, and on the other hand, it is supported by a robust design to cover for uncertainties in the model and in the contact with the soil. The algorithms proposed here are also lightweight and based on digital control, to facilitate its implementation in a real-time embedded system with limited resources. This requirement will help to reduce the gap between a platform used mainly for research and an end-product ready to market due to different reasons. First, because by limiting the technology inside the machine, the cost will be reduced making the platform more affordable for the end user. Second, to reduce resources related to its design and implementation. And finally, the aim is also to find an approach that could be implemented in different machines and steering systems by tuning only a few parameters.

Some results obtained during this research can also be found in the following publications:

1. In Fernandez et al. (2016), the simulation used for comparing different vehicles and steering systems is explained. This simulation was fully developed under the scope of this re-

search, to be able to analyse the effect of different approaches in a real-time environment. Also, it is implemented in an interactive manner to be able to make changes on the go, not only to tune and check different approaches in the same environment, but also to be able to manipulate the uncertainties and manually change the vehicle dynamics in real-time. Furthermore, the way that the vehicles are simulated, as well as the navigation path and the controller, is very close to the real platform in order to reduce the gap between the simulation and the real environment. Another advantage of the simulation is to save time and resources, since it is possible to test the algorithms first there, before having to invest the time in moving machinery and equipment to the field. Finally, the weather plays an important role as well, since during winter or in raining season, it is very difficult to test the vehicles on the field.

2. In Fernandez et al. (2018a), the first results of using basic adaptive control in a tractor with varying hitch forces is introduced. Some analysis of the effects produced by a combination of changes in the speed and in the hitch forces to the behavior of the vehicle is also presented. Here, the yaw rate is controlled by a self-tuning regulator. Nevertheless, improvements to adapt the lateral position, as well as robustness in the design, are still missing.

3. In Fernandez et al. (2018b) a robust design is presented with the advantage of covering the yaw rate and the lateral position in the same controller. A design methodology is presented together with some analysis of the robustness of the system. The method is implemented on a skid-steering agricultural robot, and the results gathered from the field test are very satisfactory.

In the publications mentioned above, only a partial representation of the systems is introduced. Therefore, these methods

are presented with all their details in Chapter 3. Also, an extended analysis of results can be found in Chapter 4. Finally, for the benchmark, an observer-based optimal controller is used. This controller was selected, since some good results were found in the literature for the application of optimal control to the path-following problem (see Sec. 2.2.8). Furthermore, some improvements were done to this controller by the introduction of an observer and the online calculation of the Riccati equation. Even though this observer-based type of optimal controller is very well known, no publications were found related to the application of this controller to problem of path following. Therefore, the results related to this controller presented Chapter 4, are also very interesting.

1.7 Outline of the thesis

This document intends to present the scope of this research including an analysis of the problem, proposed solutions and an analysis of simulation and field tests, with the respective chapter distribution as follows:

Chapter 2 presents first a general and abstract representation of a vehicle and its positioning system, focused on the problem of autonomous driving. For this, three factors are important to take into account: *Navigation path*, *Control strategy* and *Lateral dynamics*. Based on that, a literal review follows, covering these three factors in order to better understand the needs and direction to be taken for this research. At the end of this chapter, a summary of the literature review and a conclusion about the theory and methods to be used for these research are presented.

Chapter 3 dives into the methods proposed to solve the problem presented in Sec. 1.2. This chapter is divided first into three sections presenting three different control approaches: *Cascade self-tuning regulator*, *Robust self-tuning regulator* and *Optimal*

digital control, where the latest is used as benchmark. At the end of the chapter, a comparison of the different controllers is presented.

Chapter 4 introduces first two different vehicles used for the simulations and tests of the proposed methods: *Tractor-Implement model* and a *Skid-steering robot*. After that, the following sections present respectively the simulations and results of the three methods mentioned in Chapter 3. At the end of the chapter, a comparison of results is presented.

Chapter 5 concludes finally with a discussion of the results and an introduction of improvements of the proposed solution as well as possible research directions for future work.

Chapter 2

State of the art

Figure 2.1 illustrates a very general representation of the positioning and motion system of a ground vehicle. Here, the inputs are the desired speed and curvature. On the one hand, the *Forward Dynamics* bring the vehicle to a desired speed in the driving direction (V_x), which is also influenced by the desired curvature. On the other hand, the *Steering System* brings the vehicle to a desired radius of curvature. After the curvature and forward speed are reached, the *Lateral Dynamics* move the vehicle with its respective yaw rate $\dot{\beta}$ and lateral speed V_y. Finally, integrating the forward speed, the lateral speed and the yaw rate, we can find the vehicle's position, x and y, in the heading direction β in a Cartesian coordinate system. As it can be seen, the *Forward Dynamics* as well as the *Lateral Dynamics* are time varying and the main factor for that variation is the contact between the wheels and the ground, especially if the ground has a lot of irregularities as is the case for off-road vehicles. The block called *Steering System* represents the different constellations that can be found in a vehicle such as Ackermann-steering, 4-Wheel-steering, Skid-steering and Articulated-steering among other less common. It should be taken into account, that for differential steering systems such as the Skid-steering, the *Forward Dynamics* are directly related and depend on the steering and vice-versa. Whereas for the Ackermann-steering, the *Forward Dynamics* and the *Steering System* are not dependent on each other. For instance, the

radius of curvature of an Ackermann-vehicle will only depend on the steering angle (and the slip angle produce by the contact to the ground), almost regardless of the vehicle's speed. Whereas the radius of curvature in a stationary skid-system will always be infinity and one will need to produce some speed in both right and left tracks of the vehicle, to produce forward speed and a radius of curvature.

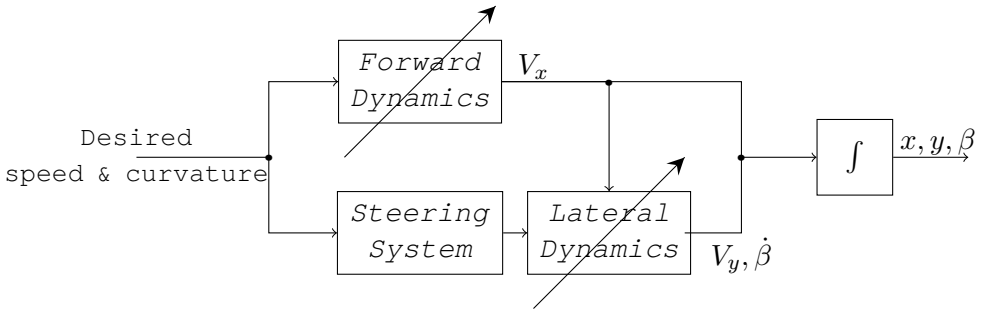


Figure 2.1 General positioning representation of an off-road vehicle.

The problem of controlling the position of a moving off-road vehicle consists mainly in dealing with the changes in the forward dynamics and in the lateral dynamics, where the last ones depend themselves on the forward dynamics. Therefore, for the sake of simplicity we will deal only with the lateral dynamics and assume that controlling the vehicle to reach a desired forward speed is already covered. Now our problem reduces to achieving some desired lateral dynamics, which depend on the forward dynamics, on the steering system and on the contact between the wheels and the ground, with the aim of following a path in an autonomous manner.

The autonomous off-road vehicle problem can be addressed into three parts. The first part concerns the reference used for navigation, i.e. the way-path or navigation path used as a reference for the vehicle to be followed. The second part is related to the control approach or scheme to be implemented, i.e. how to bring the vehicle from a current position to a desired one inside the navigation path. The last part concerns the model that will represent the vehicle behaviour (equation of motion) and which

will be used by the control scheme for a correct navigation. This equation of motion could be represented either by the kinematics of the vehicle, by its dynamics or by an estimation without real physical meaning. It is necessary to know what work has been done in this areas to be able to contribute with a new approach. For that, the following sections present first a literature review about the state of the art in how to solve this autonomous off-road vehicle problem. Then, a summary of the state of the art will be introduced in order to help to understand the research direction followed in this work. Finally, a theoretical basis for this research will be presented.

2.1 Navigation path

According to De Luca et al. (1998), the type of motion of an autonomous terrain vehicle can be divided into three main strategies: *Point-to-point motion*, *Path following* and *Trajectory tracking*. In *point-to-point*, the vehicle only aims to arrive at a goal point from an initial one regardless of the geometry between them. For *path following*, the vehicle is expected to follow a path, i.e. apart from reaching a goal point, the contour between the initial and end point should be specified and followed. Finally in *trajectory tracking*, the vehicle is not only intended to follow the path, but the velocity of the system is also controlled which implies that for each point, an arrival time is expected. As aforementioned at the beginning of this chapter, we will be dealing only with the lateral dynamics and assume that the vehicle is able to achieve a desired forward speed. Therefore, *trajectory tracking* will be out of the scope of this research. That leave us with *path following*, since the contour between points should also be followed by the autonomous off-road vehicle. How the way-points are connected, could be divided into continuous and discontinuous. A discontinuous path, connects the navigation points with straight lines as it is illustrated at the left side of Figure 2.2. A continuous path will smooth the connections

around the navigation points as it is depicted at the right side of the same figure, where for example, the angles of entering and leaving each point are specified. In this example a very simple rule of thumb is followed, i.e. 90^o and 270^o between the first two points and 90^o and 180^o between the second and the third point. Nevertheless, the aim is to specify the curves according to the vehicle's physical characteristics and limitations, since the control signal will represent the curvature radius. A very common technique for smoothing the contour at the connection points with circles was first presented by Dubins (1957), but there are other methods like splines, Bèzier curves or simple curvature path planning (SCPP) (Scheuer and Fraichard, 1997). Barton (2001), presents a summary and literature review of different methods for continuous curvature path planing (CCPP). Also newer methods like Voronoi diagrams and Fermat's spirals can be found in Candeloro et al. (2013).

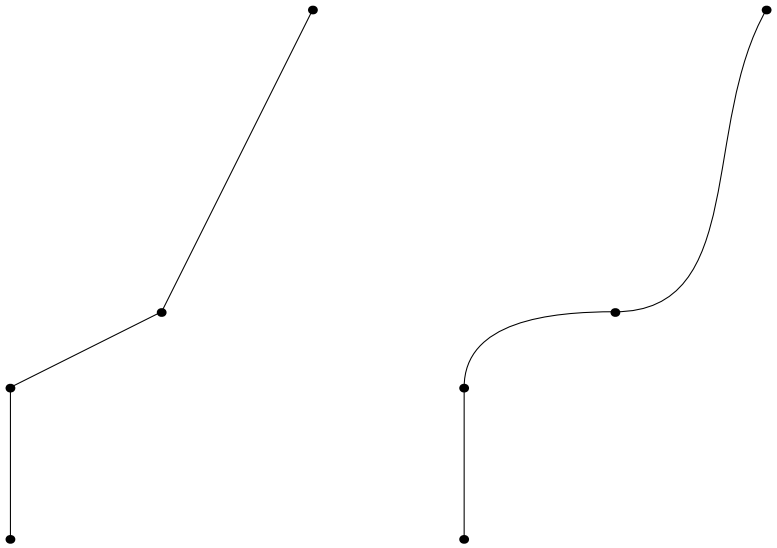


Figure 2.2 Discontinuous path at the left and continuous path at the right.

Apart from the approach where the user places the desired navigation points manually, an automatic way of distributing the way-points is also a very extended topic of research known as Coverage Path Planning (CPP), which can be compared to the travelling salesman problem (TSP). The first interesting survey on

CPP was presented by Choset (2001) and followed by Galceran and Carreras (2013) with some interesting updates such as multi-robot CPP.

2.2 Control strategies

The control strategy to be used for an autonomous vehicle is based on its behaviour and can be represented either by the geometry of the vehicle, by the vehicle kinematics or by its dynamics. For systems where the dynamics are not known, geometric and kinematic approaches should be enough, especially if the tracking strategy is based on *point-to-point*. For *path following* and *trajectory tracking* approaches, kinematic or dynamic models can be used. Figure 2.3 presents a distribution of control schemes according to the vehicle model based (Amer et al., 2017).

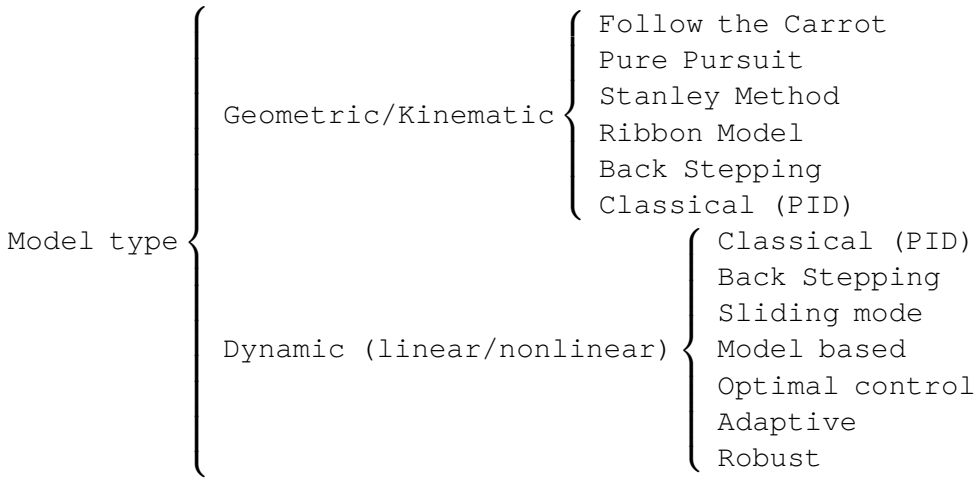


Figure 2.3 Controller according to the model type.

Although pure pursuit has been used as a standard benchmark to validate new controllers, optimal controllers are very popular in modern control theory and a study by Sharp (2005) presents excellent path following performance, also in the presence of changes in the vehicle dynamics. Nevertheless, if the system is time varying, an adaptive controller could be more attractive. The following subsections present more details and a literature review of each control scheme.

2.2.1 Follow the carrot

This is the simplest path following controller and the name comes from the idea of guiding a donkey by dangling a carrot in front of it. First Rankin et al. (1996) and later Barton (2001) present some applications and details about this method, which basically consists of adjusting the steering angle, by using a look-ahead distance, as design parameter to find the next following point on the way-path (the carrot). It can be deduced from Figure 2.4, the difference between the orientation of vehicle and the orientation of the carrot point as follows:

$$\beta_e = \beta_k - \beta. \tag{2.1}$$

Using a proportional control law K, we can then find the control signal

$$\delta = K \cdot \beta_e, \tag{2.2}$$

where δ is the desired steering angle. This method can be directly applied to an Ackermann-steering vehicle. For its usage in other steering system such as Skid-steering, the equivalent curvature has to be calculated.

Some improvements can be done by extending the control law to include an integral and a derivative part as proposed by Amer et al. (2017). The behaviour of the method will depend on the look-ahead distance: if it is very small, the vehicle will tend to oscillate, and if it is too big, it will cut edges. This method can be extended to include the cross-track error in the control law having a MISO system (Barton, 2001).

2.2.2 Pure pursuit

Pure pursuit is an extension of the follow-the-carrot method. In this case, the curvature between the vehicle and the carrot point can be directly calculated. According to Barton (2001), the method presents better performance and less oscillations than

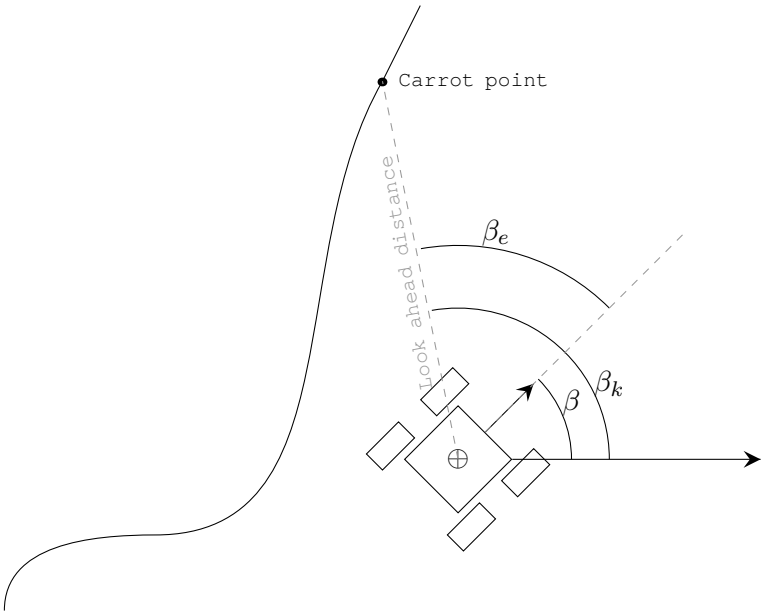


Figure 2.4 Follow the carrot.

follow-the-carrot. This method is one of the widest used methods (Amer et al., 2017) and can be first found in Wallace et al. (1985) and then with more details and some first results in Amidi (1990) and Coulter (1992). Very interesting results comparing pure pursuit against other popular methods can be found in Wit et al. (2004) and Snider (2009). In Figure 2.5, it can be seen that the calculation of the curvature radius R can be done by applying the law of sines and cosines as follows (Snider, 2009):

$$\frac{L_d}{\sin(2\beta_e)} = \frac{R}{\sin(\frac{\pi}{2} - \beta_e)}, \tag{2.3}$$

which can also be rewritten as

$$\frac{L_d}{2\sin(\beta_e)\cos(\beta_e)} = \frac{R}{\cos(\beta_e)}, \tag{2.4}$$

and solving for R we have

$$R = \frac{L_d}{2\sin(\beta_e)}. \tag{2.5}$$

For controlling a wider range of vehicles, one can use curvature as the control signal, which is the inverse of the radius $(1/R)$. If the steering angle δ is to be used as the control sig-

nal, some calculations have to be done based on the geometry and measurements of the vehicle to be controlled. As previously mentioned, this method is very popular and has been widely applied. For more details on how and where it has been applied, the reader is recommended to refer to Amer et al. (2017).

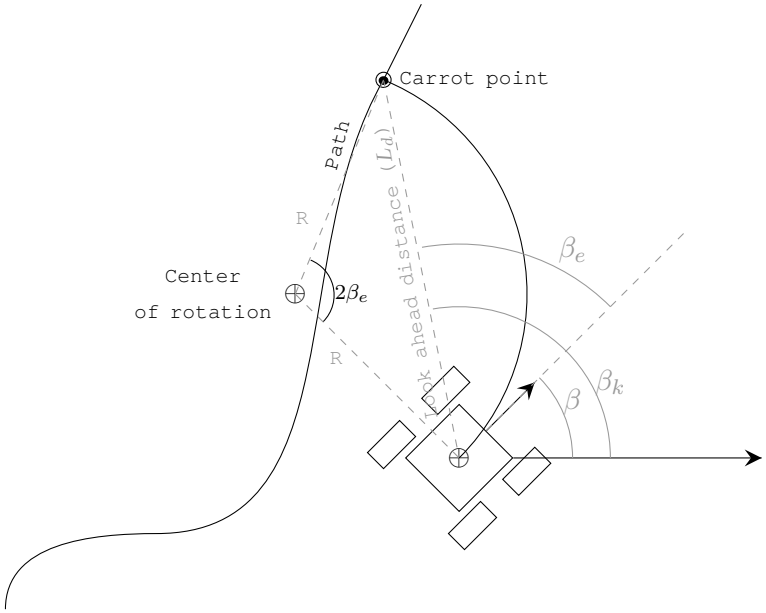


Figure 2.5 Pure pursuit.

2.2.3 Stanley

A general geometric representation of the Stanley method can be seen in Figure 2.6. Its name comes from the robot that won the 2005 DARPA Grand Challenge, Stanley (Thrun et al., 2006). It was developed by Stanford University for its autonomous Ackermann-steering robot and the idea is based on the cross-track error and the orientation error, which is calculated as follows:

$$\beta_e = \beta_p - \beta. \tag{2.6}$$

Since this method was developed for an Ackermann-steered vehicle, the steering angle is a nonlinear function of the cross-track error e and the heading error β_e:

$$\delta(t) = \beta_e(t) + \tan^{-1}\left(\frac{k \cdot e(t)}{V_x(t)}\right), \tag{2.7}$$

where k is a design parameter. It can be seen from Eq. (2.7) that as the cross-track increases, so does the steering angle towards the path. On the other hand, assuming the vehicle drives along the path, the cross-track is zero and the steering angle equals the heading error, i.e. the steering angle adjusts to the heading error. The original detailed explanation of the steering algorithm can be found in Hoffmann et al. (2007). Figure 2.6 illustrates a possible representation for a general vehicle (not necessarily Ackermann-steering), where a transformation from the steering angle to the radius of curvature should be done if a different steering system is to be used.

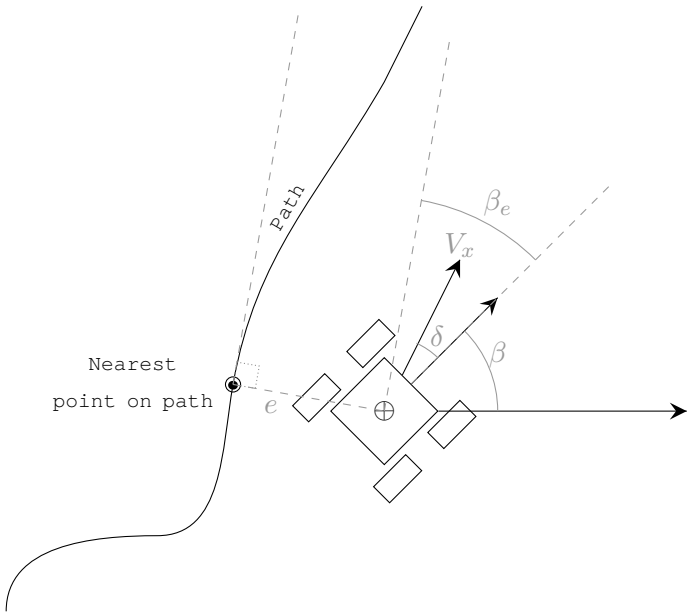


Figure 2.6 Stanley method.

2.2.4 Other basic algorithms

Vector pursuit is a bit more complex than *Follow the carrot* and *Pure pursuit*. It was first explained in Wit (2000) and the author also presented some very interesting results in Wit et al. (2004), where it can be seen that it performs remarkably better, especially for look-ahead distances where follow the carrot and pure pursuit are unstable. The only disadvantage that can be seen in the results, is that in steady state, vector pursue seems to have a steady state error in the "U" path and in the figure-eight

path. Nevertheless, it is discussed here, that at higher speed, vector pursuit will remain stable compared to follow-the-carrot and pure pursuit. This method is based on screw theory and it also takes into account arriving at the way-point with the correct orientation and curvature, which is not considered in the previous methods. Nevertheless, it does not seem to be as popular and there is not much information and results in the literature to be found.

Follow the past is another pursuit method based on a look-ahead distance and can be found in Hellström and Ringdahl (2006). It uses the recording behaviours of the vehicle about steering and orientation to solve the problem of cutting corners. It focuses on forest vehicles which use articulated steering, and defines three independent behaviours: *Turn towards the recorded orientation*; *Mimic the recorded steering angle*; *Move towards the path*.

A Lyapunov based method is presented in Kanayama et al. (1990) for controlling non-holonomic vehicles using as input a reference posture $[x_r(t) \quad y_r(t) \quad \theta_r(t)]^T$ and a reference velocity $[v_r(t) \quad \omega_r(t)]^T$, which are functions of time. This control approach aims to achieve desired linear and rotational speeds $[v(t) \quad \omega(t)]^T$. This makes the method more suitable for a *trajectory tracking* design which, as already mentioned, is a more complex problem.

Ribbon model method was first introduced in Sun et al. (2012) and later in Chen et al. (2014). The basic idea is that the performance of the path-following controller depends on the model and therefore, a new vehicle-road model called "ribbon" was developed. It considers the road width and the vehicle geometry. Based on this model, a steering controller and a speed controller were designed. The combination of model and controller allows to adapt to different kind of environments. Its name comes from the idea that the designed path to be followed is a "ribbon" with left and right edges delimiting the road. Here the idea is to

keep the vehicle inside of this "ribbon" and depending on the road conditions, the speed is adjusted.

Back stepping is similar to the Lyapunov based method since the aim is to bring the tracking errors $[e_1 \quad e_2 \quad e_3]^T$ to zero, as it presented by Ye (2008) and Hima et al. (2011). In those examples, it is used for a differential steering vehicle, where

$$\begin{bmatrix} e_1 \\ e_2 \\ e_3 \end{bmatrix} = \begin{bmatrix} \cos\beta & \sin\beta & 0 \\ -\sin\beta & \cos\beta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_r - x \\ y_r - y \\ \beta_r - \beta \end{bmatrix} \tag{2.8}$$

and the control signal takes the form

$$\begin{bmatrix} v \\ \omega \end{bmatrix} = \begin{bmatrix} v_d + \tilde{v} \\ \omega_d + \tilde{\omega} \end{bmatrix}. \tag{2.9}$$

Here, $\tilde{v}$ and $\tilde{\omega}$ are defined as virtual control errors and

$$\begin{bmatrix} v_d \\ \omega_d \end{bmatrix} = \begin{bmatrix} v_r \cos(e_3) + k_1(v_r, \omega_r) e_1 \\ \omega_r + k_y v_r e_2 + k_2(v_r, \omega_r) \sin(e_3) \end{bmatrix}, \tag{2.10}$$

where k_y is constant and k_1 and k_2 are functions of the reference linear and angular velocities v_r and ω_r (Fierro and Lewis, 1995; Jiangdagger and Nijmeijer, 1997). This method can be found also in agricultural applications (Fang et al., 2005).

Sliding mode treats the control signals as discontinuous functions, similar to switching control, where the signal slides from one boundary to another. Therefore the name, sliding mode. This makes the controller suitable for nonlinear systems. Nevertheless, a drawback could appear when generating high frequency control signals since the stress in the actuator increases and that could lead to failure or shorter lifetime. This technique is very popular and has being applied in numerous cases for the path following problem (Aithal and Janardhanan, 2013; Arslan and Temeltas, 2011; Canale et al., 2009; Dagci et al., 2003; Kigezi et al., 2015; Martin et al., 2013; Raffo et al., 2009; Solea and

Cernega, 2015; Solea and Nunes, 2007; Wang et al., 2008; Yang and Kim, 1999; Zhou et al., 2005), including agriculture vehicles (Li and Hu, 2014; Matveev et al., 2013).

Other less popular methods can be found in the literature such as a *Kinematic-based model* (De Luca et al., 1998; Snider, 2009); *Future prediction control* (Zakaria et al., 2013, 2012); *Linear interpolation* (Scaglia et al., 2010, 2015) and *Virtual force* (Rossetter, 2003);

2.2.5 Classical control: PID

Figure 2.7 presents a general representation for a cascade controller of the lateral position. Here, each controller can take the form of a PID, or a combination of P, I and D. This basic configuration, as such, and with some modifications can be found in different applications (Amer et al., 2016; Derrick and Bevly, 2008; Derrick et al., 2008; Derrick and Bevly, 2009; Gartley and Bevly, 2008; Kayacan et al., 2015; Ping et al., 2010). A transfer function of a PID controller will consist of three parts, the proportional (P), integral (I) and the derivative (D) part

$$G_{PID} = k_p + \frac{k_y}{s} + k_d \cdot s, \tag{2.11}$$

where k_p, k_y and k_d are the parameter to be tuned.

In Figure 2.7, the most inner loop represents the steering system, which for a standard tractor, could be formed by a hydraulic valve with dead band and saturation. The controller consists of a lookup table and an integral action "I" (Derrick and Bevly, 2009; Gartley, 2005). The following loop in the cascade system represents the yaw rate which can be generated by a kinematic or a dynamic model. In any case, a proportional action should be enough to control the yaw rate. Finally, the outer loop will cover the lateral position or cross-track, which is unstable since applying a constant steering angle will keep the vehicle turning. For

this case a PID action is used, although only a proportional and an integral action should be sufficient.

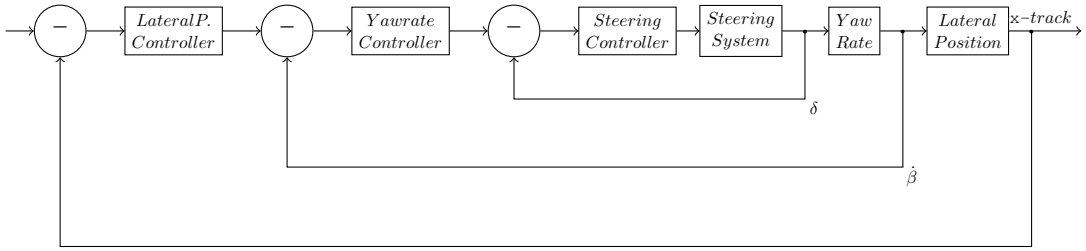


Figure 2.7 Block diagram of a cascade controller for the lateral position of a vehicle.

As already mentioned, the dynamics of the vehicle will change depending on the forward speed. This will be reflected in the DC gain, which will increase as the speed increases. For high speed, the proportional action of the yaw rate controller will not be enough to achieve zero steady state as the DC gain will be very high. In order to maintain zero steady state, a feed-forward gain has to be included, as it is the case in the studies of Barton (2001), Gartley (2005) and Snider (2009). This can be represented in Figure 2.8.

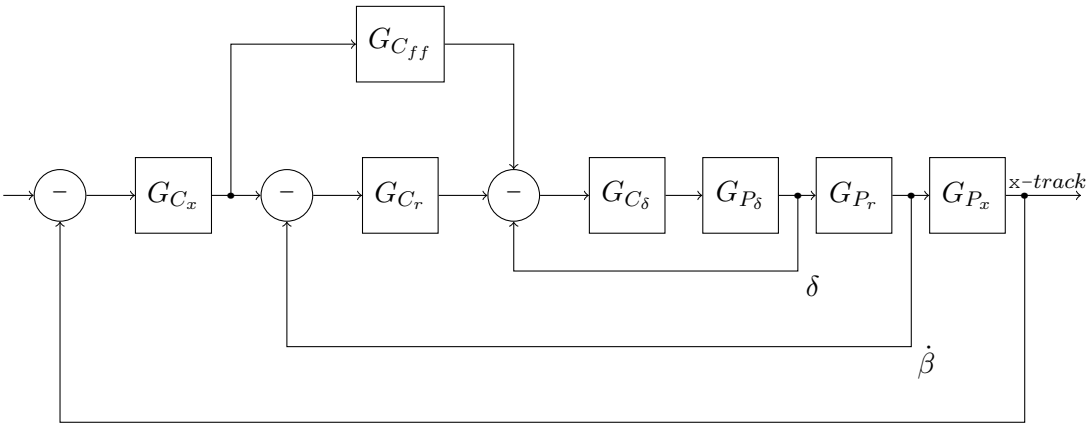


Figure 2.8 Block diagram of a cascade controller with feed-forward for the lateral position of a vehicle.

Despite the simplicity of the method, one of the drawbacks of using this approach in a fast driving vehicle, is that the parameters correspond only to one configuration of forward speed and terrain interaction. Changes in the operating conditions will require tuning a new set of parameters. This is especially

expensive if one would like to tune different parameter for a wide range of operating conditions.

2.2.6 Model predictive control (MPC)

Model predictive controllers can be found in linear and nonlinear forms. Backman et al. (2012) presents some results for agricultural application in its nonlinear form. It also presents a short review of MPC for tractor-trailer systems. One of the drawbacks of MPC is its high demand for real-time calculations in order to find reliable predictions. Especially for high speeds such as 20 m/s (∼70 km/h), the calculations have to take place faster to achieve accurate predictions. Perhaps desktop processors could make performing such high frequency calculations possible, nevertheless, current commercial embedded systems still have limited resources varying from 100 Mhz to a max of 450 Mhz. This computational problem seems to be approached in works such as Keviczky et al. (2006), Falcone et al. (2007), Klančar and Škrjanc (2007), Kühne (2005), Kuhne et al. (2005). Finally, even though in Amer et al. (2017) is claimed that the solution to the high demand for calculation resources could be the use of metaheuristics (Merabti et al., 2016; Xue et al., 2017), guaranteeing stability appears to be an issue (Backman et al., 2012).

2.2.7 Other nonlinear controllers

Fuzzy logic and artificial neural networks play also an important role in the problems of path following and trajectory tracking. Kayacan et al. (2012), Cao et al. (2017) and Zhu et al. (2005) present some approaches for the agricultural applications and Zhang et al. (2013) for off-road vehicles. There are also other interesting approaches to be found in the literature for ground vehicles such as the ones presented by Yoo et al. (2006), Ahmed and Petrov (2015), Faress et al. (2005), Wai and Liu (2009) and Liao et al. (2006).

Although Kayacan et al. (2012) presents very interesting results and claims to use a computational simple method, the complexity of those approaches as well as the resources needed are not really suitable for its implementation in commercial embedded systems with limited resources.

2.2.8 Modern control: Optimal pole placement

The method of pole placement is one of the most simplest control strategies. Figure 2.9 depicts a state feedback system, where the aim is to place the poles in such a way that the closed loop stays stable. The big question is where the poles should be placed to guaranty a trade-off between performance and control effort, and the solution to that is called optimal control. This problem is also referred to as the Linear Quadratic Gaussian (LQG) problem and is based on minimizing the cost function of the form

$$V = \int_0^\infty (y^T(t)y(t) + u^T(t)Ru(t))dt, \qquad R > 0 \tag{2.12}$$

where the optimal control law that minimizes this function takes the form

$$u_{opt} = F \cdot x. \tag{2.13}$$

Here, R is a tuning parameter and x are the states of our system. More details on how to solve this problem in a digital form can be found in Ogata (1995).

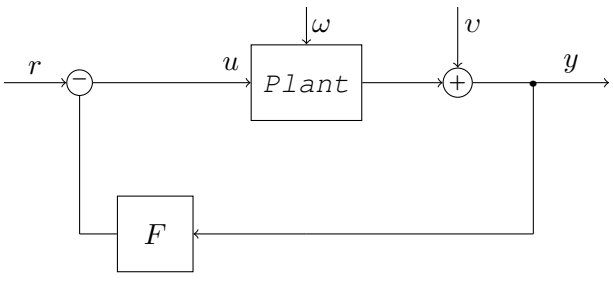


Figure 2.9 Closed-loop system with state feedback.

Regarding the path following problem, Snider (2009) presented some results for the use of optimal control for autonomous car-like robots. These results demonstrate that an optimal con-

troller with a feed-forward to compensate for increasing gain with the speed (as mentioned in the PID method in Section 2.2.5) and a past-observations model to address overshoot, performs better than other more basic controllers (pure pursuit). The past-observations model is based on optimal preview control and detailed application for the path following problem can be found in Sharp (2005), where some results also demonstrate good robustness against changes in the dynamics of the vehicle. Alternatively, to approach the changes in speed, Sharp (2012) makes use of gain scheduling to adapt this approach to different gains in a motorcycle application. In addition to performing better at higher speeds, this controller also covers a wider range of driving scenarios. Snider (2009) explains that this method performance is similar to the MPC but consumes less computational resources, although it requires past-observations that have to be processed. Nevertheless one of its disadvantages, besides that the overshoot is not completely reduced, is that it is not very robust to speed changes. To reduce computational resources and make the system more robust to rapid changes and measurement noise, the use of an observer that works as a filter can be applied. A general representation of such a system with an observer and a state feedback can be found in Figure 2.10, where ω is the process noise, v is the measurement noise and $\hat{x}$ are the state estimations.

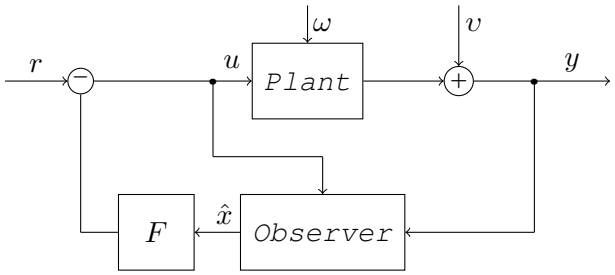


Figure 2.10 Closed-loop system with state feedback and an observer.

2.2.9 Adaptive control

Adaptive control is a well known branch of Control theory and a very good overview and summary of different approaches can be

found in Tao (2014). Also a very extensive book, which focuses on digital systems for adaptive and robust control can be found in Landau et al. (2011). One can find adaptive control designs for linear and nonlinear systems, and each of them can take a deterministic approach (using models without considering noise) or a stochastic approach introducing measurement noise in the models. Also regarding how the control parameters are updated, we can have direct methods or indirect methods where e.g. the control parameters are updated through an estimation process. Regarding the parameter adaptation, the estimation can be designed with a gradient algorithm or with a least-squares algorithm.

Some examples of direct methods could be *Gain scheduling* and *Model Reference Adaptive Systems* (MRAS). Examples of indirect methods could be *Self Tuning Regulators* (STR) and *Dual Control*. Although those classifications were covered in a general way, Åström and Wittenmark (1994) and Tao (2014) explained that both methods (STR and MRAS) could take both forms (direct and indirect).

Gain scheduling can be considered one of the simples ways of adaptive control. As it is depicted in Figure 2.11, in a system where the *Process* variations depend on the *Environment*, the parameters of the *Controller* can be adjusted with the *Gain Scheduler* as long as the changes in the *Environment* are measurable. For instance, the *Gain Scheduler* could take the form of a look-up table or the form of a function $f(.)$ where . represents the *Environment* variable to be measured. The aim is to produce a control signal u that brings the process output y to a desired $reference$ value. This approach is well known in aviation, where the environment variable that changes the *Process* is the altitude (Åström and Wittenmark, 1994). Also in the case of steering systems, there are different publications where this scheme is used to control the steering valve of heavy machines (Baslamisli et al., 2007; Derrick and Bevly, 2009).

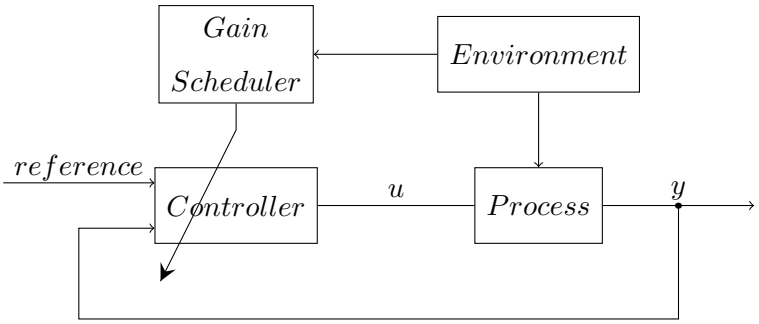


Figure 2.11 Gain scheduling.

A Model Reference Adaptive System (MRAS) is displayed in Figure 2.12 and depends on a *Reference Model* to measure how well the *Process* converges to the desired reference. These deviations are measured by the *Adaptation Mechanism*, which is the key of the solution and it is normally based on a parameter adjustment called the MIT (Massachusetts Institute of Technology) rule:

$$\frac{d\theta}{dt} = -\gamma e \frac{\partial e}{\partial \theta}, \tag{2.14}$$

where e is the error between the *Process* output and the *Reference Model* output, θ represents the control parameters and γ is the adaptation rate. This can be seen as a gradient method with the aim of minimizing the error. This scheme of adaptive control has been widely researched by Derrick and Bevly (2008), Derrick and Bevly (2009), Derrick et al. (2008) and Gartley and Bevly (2008) in the area of agricultural machines showing good results. Here, the adaptive controller is used to adjust the yaw rate feed-forward gain in a cascade system as the one illustrated in Figure 2.8.

Self Tuning Regulator (STR) is another very popular adaptive system and its main difference to MRAS is that it can be based either on pole placement design or on a minimum variance design. According to Landau et al. (2011), in its digital form the pole placement design requires no common factors between poles (A) and zeros (B), or if there are common factors, they should be stable (inside the unit circle). The advantage of this method is

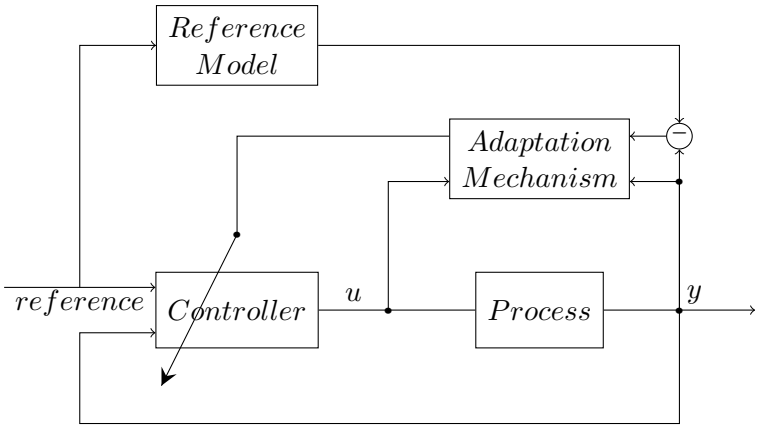


Figure 2.12 Model Reference Adaptive Control (MRAC).

that the zeros are not required to be asymptotically stable. It was already stated that the STR can take a direct form. This is the case when using a stochastic model of the form $A \cdot y = B \cdot u + C \cdot e$, where e is random measurement noise. By making the desired closed loop poles $P = C$, the system takes the form of a minimum variance controller, which is similar to the MRAC. For the minimum variance design, either the zeros are required to be asymptotically stable, or the polynomials $Q(q^{-1})$ and $P(q^{-1})$ and the scalar λ in its combination $\lambda QA + PB$ is asymptotically stable. The general form of the STR is illustrated in Figure 2.13, where it can be appreciated that the *Estimator* is in charge of finding an approximation of the *Process*, which is time varying. The estimation should fulfill the requirements (above outlined for pole placement or minimum variance) and, if that is the case, the *Controller Design* will be in charge of finding the corresponding control parameters according to the user specifications. Similar to the MRAS, u is the control signal so the output y converges to a desired $reference$. As it can be seen, the *Controller* block can take different forms according the user design. One advantage of STR over MRAS is that the zeros does not have to be stable, which is not the case for MRAS. Nevertheless, for a real physical systems the zeros are not always stable. Another advantage of STR over MRAS is that, through pole placement, the user can define desired closed loop poles based on a second order system. An example of a self-tuning controller for a tractor was presented by Noh and

Erbach (1993), based on his thesis (Noh, 1990). This approach uses a stochastic model based on a minimum variance design which, as previously mentioned, is very similar to the MRAC. Although the idea is very interesting, it was designed as a controller for the lateral position, which in theory, could contain unstable zeros. The simulation used for testing, is based on a model that approximates the lateral position producing always stable zeros. In a real system, this could be a problem, and the solution might be to use pole placement instead of minimum variance. Another problem arises by using a 2nd order system for the lateral position where, in a tractor with an hydraulic steering system and tire relaxation, the order of the system might be up to 5.

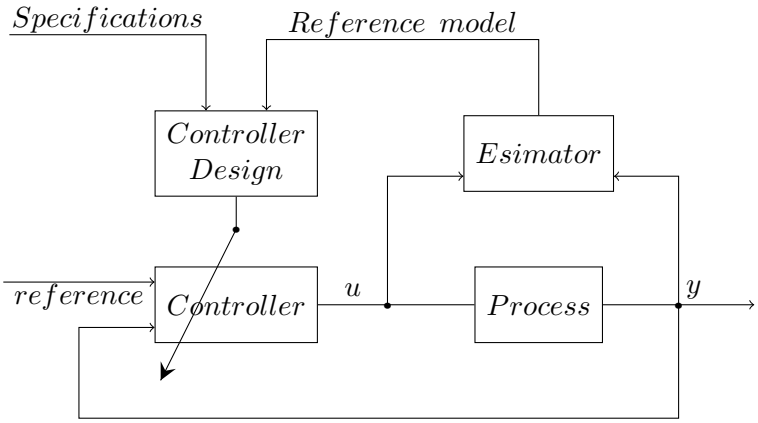


Figure 2.13 Self-tuning regulator.

Dual control was mentioned by Åström and Wittenmark (1994) as an approach that, in contrast to MRAS and RST, takes into account parameter uncertainties. Even though Åström and Wittenmark (1994) claimed at that time that this approach is too complicated to be implemented in a practical case, the concept is worth to be mentioned. Figure 2.14 illustrates the idea, where the hyperstate is a nonlinear estimator of the probability distribution of the measured states. This will allow for rapid changes in the states, covering then the uncertainties. The problem with this method is the complexity of solving the hyperstate problem which should take place in real-time. The simplest example of the hy-

perstate could be a model represented by its mean and covariance as explained in Åström and Wittenmark (1994).

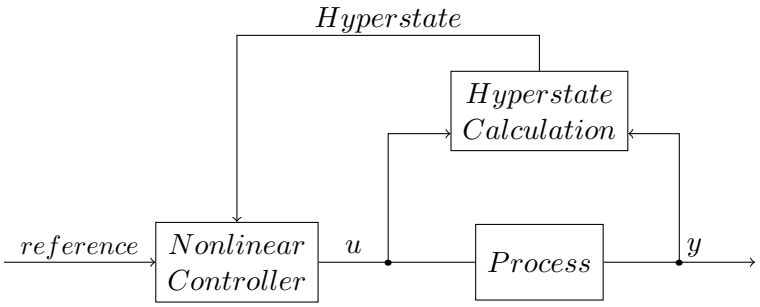


Figure 2.14 Dual control.

Other types of adaptive controllers have been presented in the literature (Landau et al., 2011) with examples such as *Multiple Model Adaptive Control*, *Adaptive Regulation*, *Feed-forward Compensation*, *Parameter Adaption Algorithms* and *Iterative Identification* which has been used in Liu and Alleyne (2014) for vehicle model identification. There are also special cases of adaptation applied to the autonomous trajectory tracking and path following problems (Chen, 2011; Das et al., 2006; Khatib et al., 2015; Li et al., 2015; Martins et al., 2008; Yoo, 2010). Interesting is for instance the work in Fukao et al. (2000), which used the backstepping strategy and compared a real robot with a virtual one tracking a desired trajectory. Here, the real model has unknown parameters such as wheel radius and width, that keep changing until the error between both robots is minimized. Also regarding backstepping, Hima et al. (2011) presented an adaptive controller law for passenger vehicles. For agricultural applications, Bayar et al. (2016b) developed a slippage estimator that changes the model on the go and adapts the controller. Lenain et al. (2003) presented a nonlinear adaptive controller for tractors in the presence of sliding.

2.2.10 Robust control

There are situation where it is really difficult to model non linearities and uncertainties. In those cases, the design of the

controller should some how take these changes into account to make the controller robust. The idea is to have a nominal model and a family of models to cover for possible changes and based on that, design a controller that covers the family of models. For that, two main approaches can be applied: *Additive model uncertainty* (see Fig. 2.15) and *Multiplicative model uncertainty* (see Fig.2.16). For the case of additive uncertainties we can see from Figure 2.15 that the family of models can be represented by

$$G'(z^{-1}) = Process(z^{-1}) + \delta(z^{-1})W_a(z^{-1}), \tag{2.15}$$

where $Process$ is the nominal plant, δ and W_a are stable transfer functions and $\|\delta(z^{-1})\|_\infty \leq 1$.

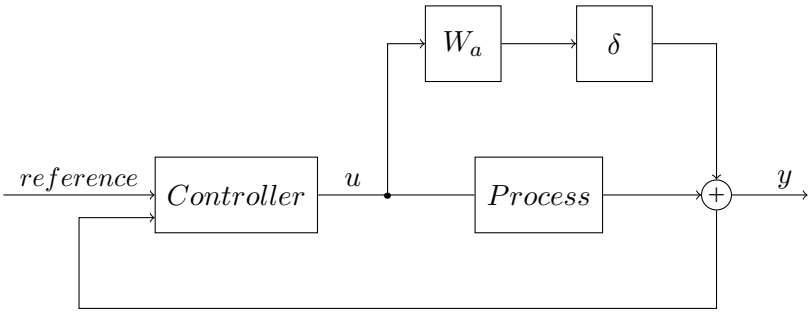


Figure 2.15 Additive model uncertainty.

For the case of multiplicative uncertainties we can see from Figure 2.16 the that family of models can be represented by

$$G'(z^{-1}) = Process(z^{-1})\left[1 + \delta(z^{-1})W_m(z^{-1})\right]. \tag{2.16}$$

Here again $Process$ is the nominal plant, δ and W_a are stable transfer functions and $\|\delta(z^{-1})\|_\infty \leq 1$.

Landau et al. (2011) presents another type of uncertainties called *Feed-back uncertainties*, which take the form

$$G'(z^{-1}) = \frac{Process(z^{-1})}{1 + \delta(z^{-1})W_r(z^{-1})}. \tag{2.17}$$

There are robust solutions to be found in the literature applied to the trajectory tracking and path following problems (Pan

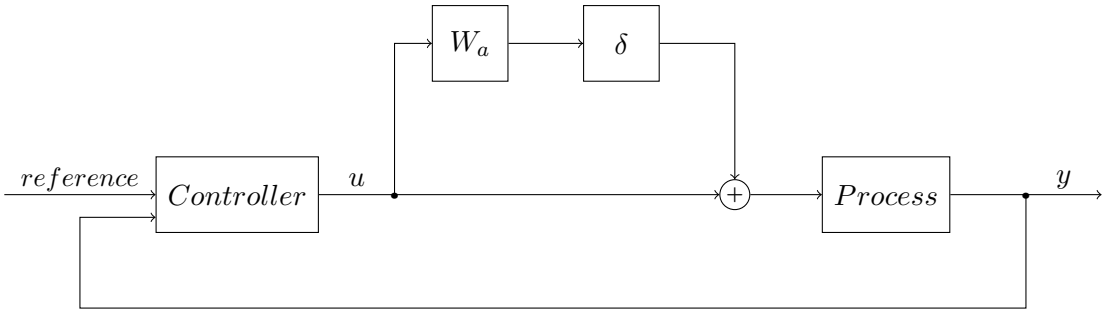


Figure 2.16 Multiplicative model uncertainty.

et al., 2014; Zhou et al., 2005). A couple of them have been mainly applied to nonholonomic and skid-steering robots(Arslan and Temeltas, 2011; Hwang et al., 2013; Inoue et al., 2013; Normey-Rico et al., 2001; Roy et al., 2015; Wai et al., 2010). Nevertheless, these solutions appear in its majority to be either complex, nonlinear and/or demand computational resources especially when combined with MPC (Bahadorian et al., 2012). They also appear to be applied to specific type of vehicles and for an indoor controlled environment.

2.3 Lateral dynamics

In the previous Subsection 2.2, different control schemes were mentioned. To be able to design and implement those control schemes, a model or representation of the vehicle is needed. For instance, Amer et al. (2017) divides the different types of vehicle models into four categories: *Geometric model*, *Kinematic model*, *Linear dynamic model* and *Nonlinear dynamic model*.

The geometric model considers only the dimensions of the vehicle as well as the geometry formed by the steering system in relation to the navigation points. This could be the case for Ackermann-steering but might not be the case for skid-steering where a differential speed is needed to steer the vehicle. The kinematic model can represent the vehicle only by its position and speed, including heading and yaw rate without considering the forces that produce the motion. Finally the dynamic models

(nonlinear and linear) extend the kinematic model and include the forces that produce the motion. The most precise approach that considers all the factor that affect the motion of the vehicle is the nonlinear model.

Now the challenge of using the same controller in different vehicles, is that each type of vehicle will have a different nonlinear model and different considerations have to be taken into account such as tire relaxation and non-holonomic [1] constrains. One way of simplifying the problem is to use a linear controller, which implies that the model of the vehicle to be used can also be linear.

In some cases the model can be linearised, as it is the case of a tractor-implement system presented in Section 4.1, whereas in other cases, such as in a skid-steered vehicle, it is not possible to linearise the model. A way of finding a linear model in those cases is through a system identification. This will require to measure the corresponding inputs and outputs of the system and in order to have an accurate model, the input signal should contain a reach range of frequencies, where the vehicle is expected to be stimulated in the real situations. There are also other situations where it is very helpful to find an identified model. For instance, in some cases the physical parameters such as the cornering stiffness or momentum of inertia are not available. Another advantage is that the identification can also take place online, which is very helpful for the cases of a time varying system. Furthermore, using system identification, the form of the equation can be represented in a desired order without the need of having parameters with a physical meaning.

[1] In a non-holonomic system, the controllable Degrees of Freedom (DOF) are less than the total DOF of the system itself (Amer et al., 2017). For instance, a ground vehicle will have three DOF in an Cartesian coordinate system (x, y and vehicle heading β), but only forward speed and steering angle can be controlled. Therefore, a ground vehicle can be considered as a non-holonomic system.

Landau and Zito (2006) presents two different types of identification algorithms: *Gradient* and *Least-squares*, where each of them have their off-line and online (real-time) version. Here, the online methods (recursive) are also presented in two sets: one related to the whitening of the prediction error and another related to the non-correlation of the observations vector. Regarding the first set, we have the following methods:

- Recursive Least Squares (RLS)
- Extended Least Squares (ELS)
- Generalized Least Squares (GL)
- Recursive Maximum Likelihood (RML)
- Output Error with Extended prediction Model (OEEPM)

And regarding the uncorrelation of observations, we have the following methods:

- Instrumental Variable with Auxiliary Model (IVAM)
- Output Error with Fixed Compensator (OEFC)
- Output Error with Filtered Observations (OEFO)
- Output Error with Adaptive Filtered Observations (OEAFO)

The methods mentioned above are used for Single Input Single Output systems (SISO). They identify the parameters of a transfer function of the type Auto-Regressive with eXogenous input (ARX). In presence of stochastic (random) disturbances, a transfer function of the type Auto-Regressive Moving Average with eXogenous input (ARMAX) can be used. One can say that the most popular method is the least-squares, and Åström and Wittenmark (1994) dedicates a whole chapter for the real-time parameter estimation using this method.

For finding Multiple Input Multiple Output systems (MIMO), one could use Subspace Identification Methods (SIM). Qin (2006) presents an overview of SIM, dividing them into open and closed loop systems. For open loop methods, the author pointed out the following methods:

- Canonical Variate Analysis (CVA) (Larimore, 1990)
- Numerical State Space Subspace System Identification (N4SID) (Overschee and Moor, 1994)
- Subspace Splitting (Jansson and Wahlberg, 1996)
- MIMO Output-Error State Space model identification (MOESP) (Verhaegen and Dewilde, 1992)

And for the closed loop methods, we could find the following representative ones:

- Prediction Error Methods (PEMs) (Forssell and Ljung, 1999)
- Errors in variables (EIV) (Chou and Verhaegen, 1997)

An iterative learning method for a dynamic model can be found in Liu and Alleyne (2014). Also regarding linear identification of off-road vehicles, Karkee and Steward (2011) applied a combination of PEM and Gradient optimization (Levenberg-Marquard) for the parameter estimation of a tractor with a single axle towed implement. Gartley and Bevly (2008) applied an Extended Kalman Filter(EKF), where the updated covariance matrix included the parameters of the model. This method can be comparable to the RLS presented in Åström and Wittenmark (1994). Kayacan et al. (2015) and Yi et al. (2009) also used the EKF for the estimation of the motion parameters. Iagnemma et al. (2004) applied a RLS algorithm for the estimation of terrain parameters. Finally, an optimization algorithm was used in Caldwell and Murphey (2011) for the estimation of the motion parameters of agricultural vehicles.

2.4 Summary of literature review

In the literature review, three main aspects were analysed to address the problem of autonomous off-road vehicles: **Navigation**, which describes the way and form a navigation path is going to be followed; **Control strategy**, which describes the method to be used to make the vehicle follow the navigation path; **Lateral dynamics**, which is needed by the control strategy in order to calculate appropriate control parameters.

Navigation could be divided into three different approaches: *Point-to-point* concerns only the arriving at each point, no matter how the vehicle moves from point to point; *Path following* concerns the way the vehicle drives between the points, connecting them in a smooth way (continuous curvature path) or in a simple way (discontinuous curvature path); *Trajectory tracking* not only concerns the way the vehicle drives, but also the timing to arrive at each way-point.

Control strategy presented a summary of different schemes for the navigation problem (*point-to-point, path following* and *trajectory tracking*). These strategies can be used for three different vehicle representations: *Geometric, Kinematic* and *Dynamic*, where each of them could be linear or nonlinear. Some strategies such as PID can be applied to all three representations, whereas others are only suitable for one or two of them. The results presented by Snider (2009) and Barton (2001) explained that the basic schemes such as *Pure pursuit*, *Stanley* and *Kinematic based*, have some drawbacks such as oscillations, bad robustness against disturbances and discontinuous paths, cutting corners and steady state error at high speeds. In Snider (2009), the best performance was thrown by the *Preview optimal control* with fair to good results in all assessments. Regarding Skid-steering, where the maneuver is a result of a difference between the angular velocities of the right and left wheels, there are different methods that deal with its modeling and representation (Al-Milli et al.,

2010; Caldwell and Murphey, 2011; Guo and Peng, 2013; Maclaurin, 2011; Martínez et al., 2005; Wang et al., 2015; Xueyuan et al., 2013; Yi et al., 2009) but as already mentioned in Section 2.2.10, due to the complexity of the system dynamics the solution to the control problem is also complex and non-linear (Arslan and Temeltas, 2011; Caracciolo et al., 1999; Inoue et al., 2013; Jun et al., 2014; Pazderski and Kozłowski, 2008; Pazderskit et al., 2004; Tchoń et al., 2015; Yi et al., 2007). Another way to deal with this complexity is the approach of *Adaptive control*. For instance, for the problem of off-road vehicles where the changes in the terrain and speed introduce more uncertainties, the answer appeared to be adaptive control. For example, for Ackermann-steering different research has been performed, more specifically for a tractor with varying hitch forces where Gartley and Bevly (2008) have done an analysis of a tractor-implement system. Derrick and Bevly (2008, 2009) as well as Derrick et al. (2008), proposed a MRAS for the yaw rate dynamics. An adaptive and predictive nonlinear controller for off-road vehicles is presented in Lenain et al. (2007). Even though the results using *Adaptive control* are very satisfactory, one of the main drawbacks of MRAS is that the zeros have to be stable. Therefore, this approach was used only for the yaw rate gain and a cascade PID has to be used for the lateral position. Another limitations of *Adaptive control* is its lack of robustness to noise and disturbances. Also according to Landau et al. (2011), only in especial applications the design assumptions that the digital plant will have stable zeros will work (delays bigger than half sampling period will produce unstable zeros).

Lateral dynamics concerns the way the vehicle behaves. As a consequence of the different control strategies, the way the vehicle is represented is of key importance. This can be done geometrically, by the kinematics of the vehicle or by its dynamics. Also, a system identification could be performed in case some model parameters are not known a priori. Another advantage

of system identification, is that the parameters of the model to be identified do not necessarily need a physical meaning. For system identification, two main approaches can be used: *Least-squares algorithms* or *Gradient algorithms*, where both can take an off-line form or an on-line (recursive) form. For instance, if an adaptive controller is to be used, an online identification of the model has to be performed to account for changes in the dynamics of the vehicle.

2.5 Theoretical basis of this research

Recalling the problem formulation of Section 1.2, it addresses the need for a controller that is capable of covering changes in the vehicle dynamics and terrain uncertainties, which is also lightweight and easy to implement in an embedded system with limited resources and finally, that is flexible enough to be implemented universally in different types of steering systems.

With that in mind, it can be concluded after the literature review in the previous subsections, that the best approach for the autonomous off-road vehicle is the implementation of an adaptive controller. Ideally, a *trajectory tracking* strategy will facilitate the error reduction, especially in the turning points, since it has to consider vehicle constraints for the design of proper forward speed and turning curvature. Nevertheless, as already mentioned at the beginning of this chapter, for the sake of simplicity and since the aim of this research is to focus on the control strategy, the approach to follow in this research for the navigation is a **Discontinuous Curvature Path**, and the ability to control a desired forward speed V_x will be assumed. As for the model to be used, the question remains open if a static model (either kinematic or dynamic) will perform better than an online estimation. For the last case, a comparison between lest-squares and gradient algorithms could be very interesting.

Regarding the adaptive control scheme, MRAS was already tested with successful results. Nevertheless, STR could bring advantages over MRAS such as the ability of working with unstable zeros and the use of pole placement for the design of a desired closed loop response.

On the other hand, the lack of robustness in the adaptive control scheme is also explained in the literature review (Sec. 2.4). Therefore, a robust approach is also considered for this research. Also, Amer et al. (2017) concludes in its summary, the importance of robust controllers to be able to effectively navigate various types of paths. A well known approach to solve the robust control problem in the domain of frequency, is by shaping some sensitivity functions through weights that enclose the family of models as presented in Section 2.2.10, with the help of some norms like the H_∞ and H_2 as constraints. There are also other methods such as linear matrix inequalities, to express both H_∞ and H_2 constraints. One problem here could be the difficulty to enclose a family of models since the uncertainties of the terrain can not be known a priori. Another problem that could be faced, apart from the complexity of the approach, is the fact that these methods still have to be translated to a digital domain for its implementation in an embedded system with limited resources. One solution could be to work directly in the digital domain. For that, the robust pole placement approach presented by Landau and Zito (2006) can be used. Also, with this approach it is possible to shape the sensitivity functions without the definition of the family of models beforehand.

Consequently, a combination of adaptive and robust control could be the ideal solution for the autonomous off-road vehicle problem. Parallel to that, the work in Snider (2009) presented very good results for the method called *preview optimal control*, with minor problems of overshooting. As already mentioned in Section 2.2.8, these overshooting problems could be solved by the

implementation of an observer in the optimal controller. This improvement in the optimal controller will be presented in Chapter 3, to be used as a benchmark for its comparison with the STR solutions proposed in this research.

Chapter 3

Adaptive, robust and optimal discrete-time control

This chapter presents the three methods proposed in this research for the control of the lateral position of off-road vehicles. The first section presents a straightforward way of implementing an adaptive RST controller. The system consists of two nested closed loops where the inner loop controls the yaw rate dynamics with the help of an estimator. The outer loop controls the lateral position in feedback with a PID controller. The second section extends the RST controller by working directly with the lateral position and by considering tracking and regulation performance as well as by introducing a robust design. The third section presents an observer-based optimal controller chosen as a benchmark for this research due to its filtering capabilities and to its energy efficiency. Also, it is a well-known controller used for autonomous vehicles (Snider, 2009)

3.1 Cascade self-tuning regulator

In this section, a nested cascade system for controlling the lateral position of an off-road vehicle is presented. The inner loop consists of a self-tuning regulator to control the yaw rate dynamics with the help of an estimator. The outer loops consist of a PID managed by a gain scheduler to control the lateral posi-

tion. This solution is the starting point of this research and is inspired by the method presented by Derrick and Bevly (2008), which is also based on a cascade system. The main difference lies in the inner loop, where a MRAC is used instead. Even though the MRAC showed good results, this method requires that the zeros of the system are stable and this can restrict the scope of the solution. Also, an exact model of the vehicle as reference is needed and that might not always be the case. Furthermore, although the MRAC is based on a cost function to find an optimal solution, the user has no influence in changing the behaviour of the vehicle. Therefore, there are three main solutions covered by the system proposed in this section: 1. A pole placement method for the yaw rate dynamics that allows the user to change the desired behavior of the vehicle; 2. A recursive system identification to find changes in the model in real-time; 3. A gain scheduler for managing the PID parameters used in the outer loop for the lateral position.

This solution is illustrated in Figure 3.1. As already mentioned, it is based on a gain scheduler and a self-tuning regulator, which is made up of a recursive least-squares parameter identification algorithm for the yaw rate dynamics $\dot{\beta}$ combined with a minimum-degree pole placement (MDPP) method for changing the parameters of a digital RST controller in real time. The MDPP is computed by solving the Diophantine equation (also called Bezout polynomial) for the desired closed-loop reference 2nd order model based on ω_m and ζ_m. Here, $r(t)$ is the reference point for a desire lateral position $y(t)$ and a classical PID controller is to be used. Furthermore, since the lateral position depends on the forward speed, a *Gain Scheduler* is used as a look-up table to apply the PID parameters that better fit the desired system response.

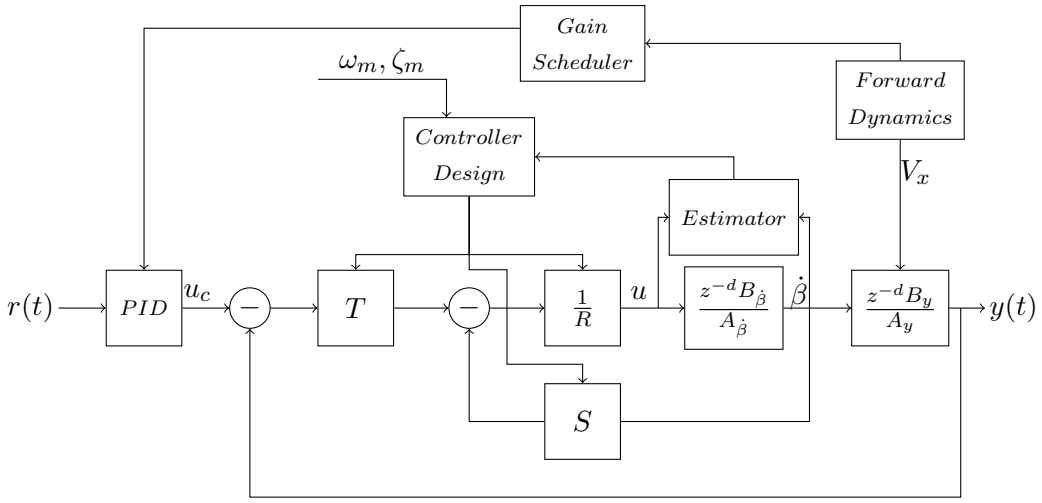


Figure 3.1 Block diagram of an indirect self-tuning regulator for terrain vehicles.

3.1.1 Gain scheduler

Since the motion dynamics of a vehicle depend on the forward speed, one will need to work with non-linear models for controlling the lateral position. On the other hand, a *Gain Scheduler* controller works well when one can find measurable variables that can be correlated to changes in the process, and in the majority of the cases, measuring the forward speed of a vehicle is possible with high accuracy. Therefore, one can correlate the forward speed to the PID control parameters of the lateral position. Therefore, one can make use of a PID look-up table for the desired range of velocities. Alternatively, if one can say in general that the gain of the closed loop system of a vehicle is inversely proportional to its speed, one can then use the same proportion where the parameters P, I and D are a function of the forward speed. Nevertheless, it is a common approach to tune the parameters manually and the question arises what should be the nominal conditions of the vehicle, such as speed and terrain, for the tuning process. Based on those decisions, one should assess in a real environment if a linear correlation between the control parameters and the forward speed will deliver satisfactory results, otherwise all parameters should be adjusted individually for each different scenario. As mentioned before, this task

is not trivial since e.g. for tuning a tractor to adapt well to different terrains and implements a lot of resources will be needed, not only time but also the preparation and transport of the machinery itself to proceed with the tests in different terrains.

3.1.2 Estimator

Assuming that the yaw rate defined by $\frac{z^{-d}B_{\dot{\beta}}}{A_{\dot{\beta}}}$ in Figure 3.1 can be modeled with a second order system, we can use the next pulse transfer function as the model to be identified by the estimator:

$$G_{\dot{\beta}}(z^{-1}) = \frac{b_0 + b_1 z^{-1}}{1 + a_1 z^{-1} + a_2 z^{-2}}. \tag{3.1}$$

For an online estimation of the process parameters, we can start with the following performance cost function:

$$V(\theta) = \frac{1}{2}\sum_{i=1}^{t} \lambda^{t-i}(\dot{\beta}(i) - \varphi^T(i)\hat{\theta})^2, \tag{3.2}$$

where

$$\hat{\theta} = [-a_1 \quad -a_2 \quad b_0 \quad b_1]^T \tag{3.3}$$

and

$$\varphi = \begin{bmatrix} \dot{\beta}(i-1) \\ \dot{\beta}(i-2) \\ u(i-1) \\ u(i-2) \end{bmatrix}. \tag{3.4}$$

The objective is to find the parameters that minimize the performance index

$$\hat{\theta} = \arg\min_{\theta} V(\theta), \tag{3.5}$$

through an iterative search of the form:

$$\begin{aligned} \hat{\theta}(t) &= \hat{\theta}(t-1) + K(t)(y(t) - \varphi^T(t)\hat{\theta}(t-1)) \\ K(t) &= P(t-1)\varphi(t)(\lambda + \varphi^T(t)P(t-1)\varphi(t))^{-1} \\ P(t) &= (I - K(t)\varphi^T(t))P(t-1)/\lambda, \end{aligned} \tag{3.6}$$

which is a recursive, least-squares method with exponential forgetting, where $0 < \lambda \leq 1$ is the forgetting factor (Åström and Wittenmark, 1994). The flow diagram of the recursive estimation can be seen in Figure 3.2. The initial conditions for the estimation vector and the measurements vector can be zero ($\hat{\theta}(t-1) = 0;\ \varphi(t) = 0$) and the initial condition for $K(t)$ and $P(t)$ can take the following unit value:

$$\begin{aligned} K(t-1) &= \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix} \\ P(t-1) &= \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}. \end{aligned} \tag{3.7}$$

After each iteration, the parameters of the model are identified and if the error produced by those parameters is under a desired limit (e.g. 1 rad/s), then the identified parameters can be used for the *Controller Design*.

3.1.3 Controller design

The controller of the form

$$R(z^{-1})u(k) = T(z^{-1})u_c(k) - S(z^{-1})y(k) \tag{3.8}$$

is calculated by the "Controller design" using the minimum-degree pole placement of Algorithm 1, which was first described by Åström and Wittenmark (1994). The aim is to have a desired closed-loop response based on the following second order reference model defined by ω_m and ζ_m:

$$\frac{y(z^{-1})}{u(z^{-1})} = \frac{b_{m0}z^{-1}}{1 + a_{m1}z^{-1} + a_{m2}z^{-2}}. \tag{3.9}$$

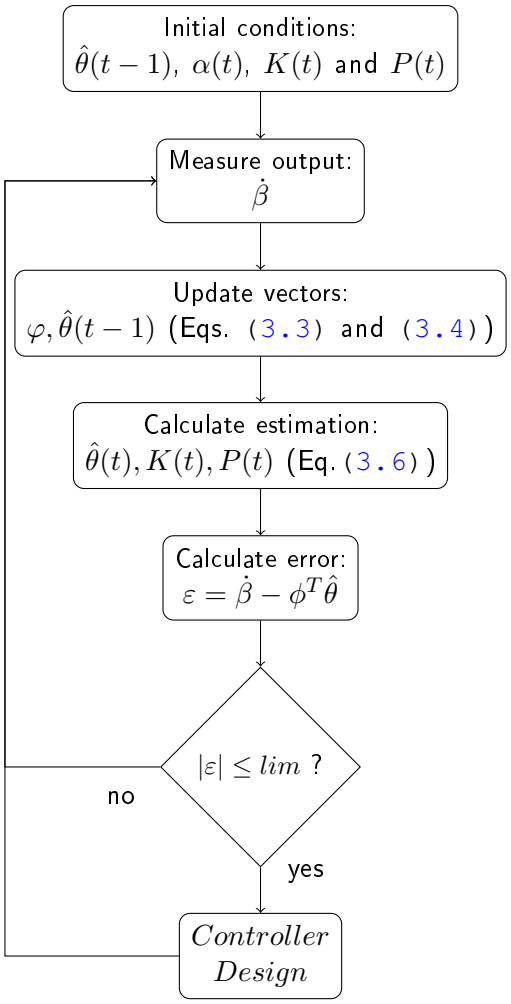


Figure 3.2 Flow diagram for the recursive least-squares estimator.

This method can be used for different vehicles (of different sizes). Nevertheless, we should first find the desired and realistic response for the vehicle that we are planning to control.

Algorithm 1: Minimum-degree pole placement (MDPP) with self-tuning regulator

Require: Polynomials A, B

Ensure: Closed loop polynomials Am, Bm and Ao

0 Online estimation of polynomials A and B.

1 Check compatibility conditions:

$$degAm = degA$$
$$degBm = degB$$
$$degAo = degA - degB^{+} - 1$$
$$B_m = B^{-}B'm$$

2 Factor B as $B = B^{+}B^{-}$, where B^{+} is monic.

3 Find the solution R' and S with $degS < degA$ from $AR' + B^{-}S = A_o A_m$.

4 Form $R = R'B^{+}$ and $T = A_o B'_m$, and compute the control signal from the control law $Ru = Tu_c - Sy$.

To apply the minimum-degree pole placement self-tuning algorithm, we first identify the polynomials A and B by the estimator as described in the flow diagram 3.2. Checking the compatibility condition one sees that $degAm = degA = 2$; $degBm = degB = 1$; $degA_o = degA - degB^+ - 1 = 1$ which implies that $degB^+ = 0$. Therefore factoring B makes $B^+ = 1$ and $B^{-1} = B = b_0 + b_1 z^{-1}$ and the reduced Diophantine equation is as follows:

$$(1 + a_{m_1} z^{-1} + a_{m_2} z^{-2})(1 + a_0 z^{-1}) = (1 + a_1 z^{-1} + a_2 z^{-2})(1 + r_1 z^{-1}) + (b_0 + b_1 z^{-1})(s_0 + s_1 z^{-1}). \quad (3.10)$$

Finally, the following equations system presents the solution to R, S and T for each cycle, to produce the control signal u in real-time for the given reference set point u_c:

$$\begin{aligned}
r_1 =& \frac{b_1}{b_0} + \frac{(b_1^2 - a_{m1} b_0 b_1 + a_{m2} b_0^2)(-b_1 + a_o b_0)}{b_0 (b_1^2 - a_1 b_0 b_1 + a_2 b_0^2} \\
s_0 =& \frac{b_1 (a_0 a_{m1} - a_2 - a_{m1} a_1 + a_1^2 + a_{m1} - a_1 a_o)}{b_1^2 - a_1 b_0 b_1 + a_2 b_0^2} \\
&+ \frac{b_0 (a_{m1} a_2 - a_1 a_2 - a_o a_{m2} + a_o a_2)}{b_1^2 - a_1 b_0 b_1 + a_2 b_0^2} \\
s_1 =& \frac{b_1 (a_1 a_2 - a_{m1} a_2 + a_o a_{m2} - a_o a_2)}{b_1^2 - a_1 b_0 b_1 + a_2 b_0^2} \\
&+ \frac{b_0 (a_2 a_{m2} - a_2^2 - a_o a_{m2} a_1 + a_o a_2 a_{m1})}{b_1^2 - a_1 b_0 b_1 + a_2 b_0^2} \\
T =& \beta (1 + a_o z^{-1}) \\
\beta =& \frac{1 + a_{m1} + a_{m2}}{b_0 + b_1}.
\end{aligned} \quad (3.11)$$

3.2 Robust self-tuning regulator

Figure 3.3 illustrates an evolution and more robust version of the first solution presented in the previous Section 3.1 (Fig. 3.1). This solution is proposed for two main reasons. First, the usage of only one controller to avoid a nested system. And second, to be able to introduce a robust design to the pole placement to cover for uncertainties in the terrain. Since two of the main factors that affect the system are the contact with the soil and the forward speed, this approach divides the vehicle model into two: yaw

rate dynamics $\frac{z^{-d}B_{\dot{\beta}}}{A_{\dot{\beta}}}$ and lateral position $\frac{z^{-d}B_y}{A_y}$. The first one is related to the changes in the soil, which are calculated online with the *Estimator*, and the second one is related to the forward speed. The *Estimator* calculates in real-time the parameters of the yaw rate characteristic and passes them to the *Controller Design* which knowing the forward speed V_x constructs an estimation of the whole plant $\frac{z^{-d}B}{A}$ and, in contrast to the approach of Section 3.1, calculates the control parameters for the whole system. Other intrinsic advantages come along with this method. For instance, tuning a PID for all the range of working speeds is time consuming. In contrast, this system requires fewer parameters to be tuned for one nominal speed, an they can be linearly adjusted by the *Gain scheduler* as a function of the vehicle speed.

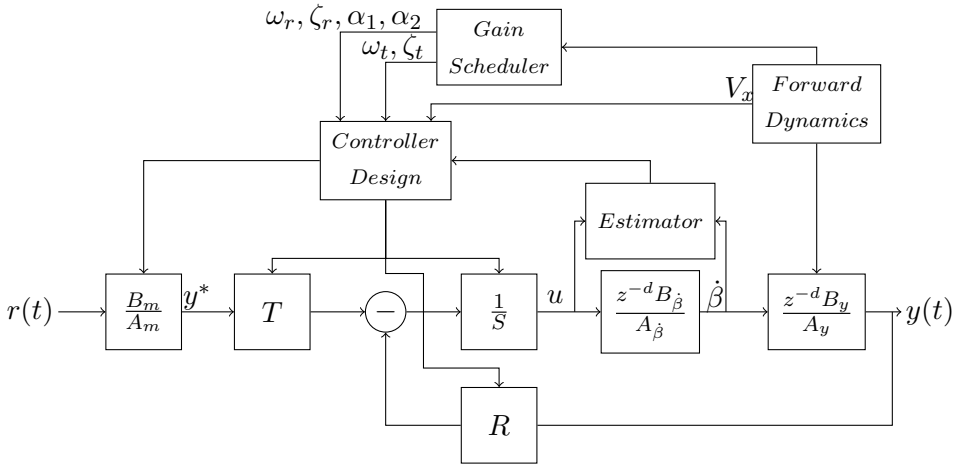


Figure 3.3 Block diagram of a robust and adaptive digital controller for terrain vehicles.

To be able to find the control parameters R, S and T, the *Controller Design* needs the dominant and auxiliary poles P_D and P_F to solve the Bezout polynomial of Eq. (3.27). For that, the *Gain Scheduler* delivers ω_r and ζ_r for the regulation dynamics used in P_D and α_1 and α_2 used for P_F. This system also introduces some tracking dynamics defined by $\frac{B_m}{A_m}$ based on the parameters ω_t and ζ_t which depend on the forward dynamics and are also delivered by the *Gain Scheduler*.

3.2.1 Gain scheduler

For the system depicted in Figure 3.3, the *Gain Scheduler* will have two tasks. The first one will be in charge of changing the tracking dynamics, which can be represented with a second order system based on ω_t and ζ_t. For instance, a system with $\omega_t = 3$ and $\zeta = 1$ will respond to a unit step achieving the steady state in approximately 3 s. For a terrain vehicle this means changing to a lane that is 1 m to the left in less than 3 s. With a forward speed of 0.25 m/s the vehicle will need 75 cm to change the lane which is quiet acceptable. Now driving at 1 m/s, in 3 s the vehicle will need 3 m to change the lane instead of 0.75 m, which might not be desirable. Therefore, in this case it is more important to specify the maximum desired forward distance to change to a lane 1 m away and based on that, use different tracking dynamics depending on the forward speed. Another factor to consider is the physical limitation of the vehicle. For instance, a differential steering will be in theory able to turn with a zero radius, whereas in an Ackermann-steering, the maximum turning radius will be limited by the maximum steering angle and the distance between axes. In summary, one option could be to choose $\zeta_t = 1$ to avoid overshooting and make the natural frequency a function of the forward speed $\omega_t = f(V_x)$.

The second task will be in charge of changing the regulation dynamics of the system depending on the forward speed, which are related to the reaction of the system against measurement noise and disturbances. For instance, high values of ω_t at slow speeds will make the system very reactive and introduce oscillations. On the other hand, small values of ω_r at higher speeds will increase the error of the system since the system will not correct its position fast enough. Also here the physical limitations of the system should be considered. A slow hydraulic steering system will not react fast enough to high values of ω_r. The auxiliary poles play also an important role in the regulation dynamics and

its function will be described in more detail in the following sections. Also in this case, one could choose $\zeta_r = 1$ to avoid overshooting and make the natural frequency a function of the forward speed $\omega_r = f(V_x)$. Finally, for the sake of simplicity, one could pick a fixed value for α_1 and α_2 between -0.05 and -0.5.

3.2.2 Estimator

Section 3.1.2 presented a recursive least-squares method with exponential forgetting for an online estimation of the process. Here, we present another method using gradient search. Figure 3.4 illustrates the general approach of the gradient algorithm. After measuring the output of the plant, an estimation is calculated with the input vector of past input and outputs measurements (φ) and the model parameters vector ($\hat{\theta}$). Then, the error ε is calculated and used by the gradient algorithm which updates the model parameters. Since we want to adapt the model parameters to real-time changes in the system dynamics, this loop is repeated endlessly.

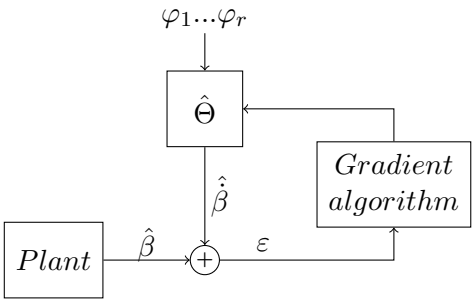


Figure 3.4 Gradient-based algorithm for the online estimation of the vehicle's yaw rate parameters.

To know if the parameters are adapting in the right direction, this method is based on a performance index of the form

$$V(\theta) = \frac{1}{2N}\sum_{k=p}^{N}(\dot{\beta}(k) - \hat{\dot{\beta}}(k|k-1,\theta))^2 = \frac{1}{2N}\sum_{k=p}^{N}\varepsilon^2(k,\theta), \tag{3.12}$$

where the estimation can be calculated by means of a linear predictor model as follows:

$$\hat{\dot{\beta}}(k|k-1) = -a_1\dot{\beta}(k-1) - \ldots - a_n\dot{\beta}(k-n) + b_0u(k-d) + \ldots + b_mu(k-d-m). \quad (3.13)$$

Recalling the objective of the least-squares method (Eq. (3.5)), we can see that here the aim of finding the parameters that minimize our cost function is the same

$$\hat{\theta} = \arg\min_{\theta} V(\theta), \quad (3.5)$$

with the difference of using a different iterative search of the form

$$\theta(l+1) = \theta(l) + \alpha f(l). \quad (3.14)$$

Here l represents the iteration step, $f(l) \in \Re^{np}$ is called the *search direction* and α is the *learning rate*. The combination of $\alpha f(l)$ could take different forms: *Steepest descent* when $f(l)$ is replaced by the negative gradient described by Eq. (3.15); *Newton method* when $\alpha f(l)$ is replaced by the Hessian (Eq. (3.16)) times the negative of the gradient; *Levenberg-Marquardt* is a combination of both *Steepest descent* and *Newton*. Those are some of the same methods used for training artificial neural networks with the difference that here only one artificial neuron is used, which means that no back-propagation of the error is needed. Also, instead of using batch training, we are using online training.

$$\nabla V(\theta) = \left[\frac{\partial V}{\partial \theta_1} \ldots \frac{\partial V}{\partial \theta_{np}}\right]^T. \quad (3.15)$$

$$\nabla^2 V(\theta) = \begin{bmatrix} \frac{\partial^2 V}{\partial \theta_1^2} & \frac{\partial^2 V}{\partial \theta_1 \partial \theta_2} & \cdots & \frac{\partial^2 V}{\partial \theta_1 \partial \theta_{np}} \\ \frac{\partial^2 V}{\partial \theta_2 \partial \theta_1} & \frac{\partial^2 V}{\partial \theta_2^2} & \cdots & \frac{\partial^2 V}{\partial \theta_2 \partial \theta_{np}} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial^2 V}{\partial \theta_{np} \partial \theta_1} & \frac{\partial^2 V}{\partial \theta_{np} \partial \theta_2} & \cdots & \frac{\partial^2 V}{\partial \theta_{np}^2} \end{bmatrix}. \quad (3.16)$$

Each i^{th} element of the gradient of the performance index ($\nabla V(\theta)$) takes then the following form:

$$\frac{\partial V(\theta)}{\partial \theta_i} = \frac{1}{2N}\sum_{k=1}^{N}\frac{\partial(\dot{\beta}(i)-\hat{\dot{\beta}}(k|\theta))^2}{\partial\theta_i} = -\frac{1}{N}\sum_{k=1}^{N}\frac{\partial\hat{y}(k|\theta)}{\partial\theta_i}\varepsilon(k|\theta), \tag{3.17}$$

where N is the number of measurements in a batch. Nevertheless, since we are aiming for online identification, $N = 1$ and so we have that

$$\frac{\partial V(\theta)}{\partial \theta} = \frac{\partial(\dot{\beta}(i)-\hat{\dot{\beta}}(k|\theta))^2}{\partial\theta_i} = -\frac{\partial\hat{\dot{\beta}}(k|\theta)}{\partial\theta_i}\varepsilon(k|\theta). \tag{3.18}$$

Rewriting (3.13) we have that

$$\hat{\dot{\beta}}(k|\theta) = \theta_i \cdot \varphi, \tag{3.19}$$

where

$$\theta = [-a_1 \cdots a_n \quad b_0 \cdots b_m] \tag{3.20}$$

and

$$\varphi = \begin{bmatrix} \dot{\beta}(k-1) \\ \vdots \\ \dot{\beta}(k-n) \\ u(k-d) \\ \vdots \\ u(k-d-m) \end{bmatrix}. \tag{3.21}$$

Applying the chain rule to Eq. (3.17) to find the gradient, we have that

$$\frac{\partial\hat{\dot{\beta}}(k|\theta)}{\partial\theta} = \varphi(k). \tag{3.22}$$

Rewriting Eq. 3.14 for the steepest descent method we have that

$$\theta(k+1) = \theta(k) - \alpha\frac{\partial V}{\partial\theta}\bigg|_k. \tag{3.23}$$

Substituting Eq. (3.22) into (3.18) and the result into Eq. (3.23) we obtain the following online iteration where one epoch has one

iteration $(l \mapsto k)$:

$$\theta(k+1) = \theta(k) + \alpha \cdot \varphi(k)' \cdot \varepsilon(k). \tag{3.24}$$

Figure 3.5 shows a flow diagram to implement the algorithm into the embedded ECU of the vehicle. We need first to set the initial conditions of the model parameters and pick a learning rate α, which should be as small as possible ($\alpha << 1$) to avoid calculating unstable parameters due to picks and measurement noise due to changes in the terrain. For the initial conditions, the model parameters could be first calculated in a simulation, so there are only few iterations needed on the real environment. In the second step, the yaw rate should be measured by means of e.g. an Inertial Measurement Unit (IMU). In the following steps, the input vector with past measurements of inputs and output should be constructed and multiplied by the model parameters to find an output estimation in order to be able to measure the error produced by the model parameters to be used. If the error produced by using the last updated model parameters is under certain limit (e.g. under 1 m), we can use it to update the parameters vector in the next cycle, otherwise the results are ignored and the model parameters are calculated with the new measurements. The updated model parameters are then used by the *Controller Design* to update the control parameters, making the system adaptive to changes in the system dynamics.

3.2.3 Pole placement: Regulation and tracking

A canonical digital controller called RST is illustrated in Figure 3.6. This structure allows one to impose different dynamics by obtaining the polynomials R and S in order to satisfy the desired *regulation performance*. T introduces a *tracking performance* that filters at the same time a desired trajectory $y^*(t+d+1)$ from a tracking model system $\frac{B_m}{A_m}$.

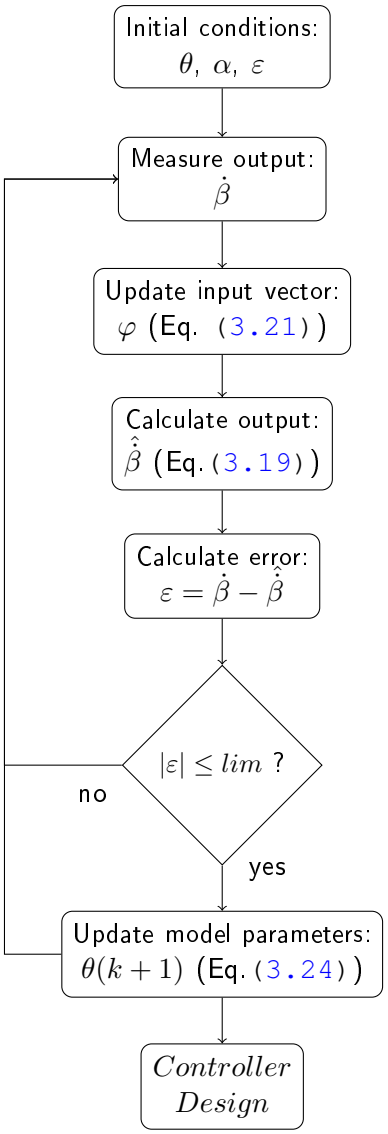


Figure 3.5 Flow diagram for the online gradient-algorithm estimator.

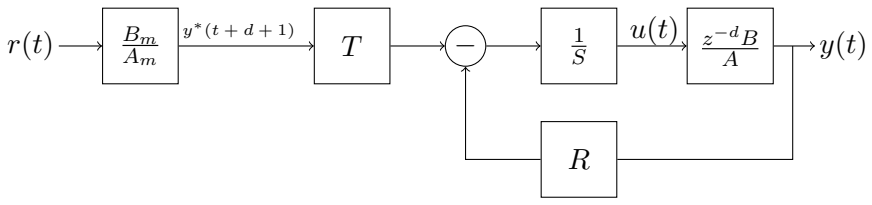


Figure 3.6 Digital canonical controller for tracking and regulation.

The process to be controlled is represented by $\frac{z^{-d}B}{A}$ and the closed loop function from the desired trajectory y^* to the output y is represented in Eq. (3.25).

$$H_{CL}(z^{-1}) = \frac{z^{-d}B(z^{-1})T(z^{-1})}{A(z^{-1})S(z^{-1})+z^{-d}B(z^{-1})R(z^{-1})} , \qquad (3.25)$$

where d is the time delay and

$$\begin{aligned} A(z^{-1}) &= 1 + a_1 z^{-1} + ... + a_{n_A} z^{-n_A} \\ B(z^{-1}) &= b_1 z^{-1} + b_2 z^{-2} + ... + b_{n_B} z^{-n_B} \\ S(z^{-1}) &= 1 + s_1 z^{-1} + ... + s_{n_S} z^{-n_S} \\ R(z^{-1}) &= r_0 + r_1 z^{-1} + ... + r_{n_R} z^{-n_R}. \end{aligned} \tag{3.26}$$

3.2.3.1 Regulation

To compute the coefficients R and S of the digital controller we solve a Bezout polynomial equation of the form

$$P(z^{-1}) = P_D(z^{-1})P_F(z^{-1}) = A(z^{-1})S(z^{-1}) + z^{-d}B(z^{-1})R(z^{-1}). \tag{3.27}$$

The characteristic polynomial P contains dominant and auxiliary poles. The dominant poles P_D are chosen from the digitalization of a second-order system defined by ω_0 and ζ. The digital auxiliary poles P_F improve the robustness of the controller and are normally smaller (faster) than the real part of the dominant poles. Typical values for the auxiliary poles are $-0.05 \leq \alpha_1 \leq -0.5$ and α_2 either equals 0 or $\alpha_2 = \alpha_1$. By defining the characteristic equation

$$P(z^{-1}) = 1 + p_1 z^{-1} + ... + p_{n_P} z^{-n_P}, \tag{3.28}$$

we obtain R and S by solving

$$x = M^{-1} p, \tag{3.29}$$

where

$$\begin{aligned} x^T &= [1, s_1, ..., s_{n_S}, r_0, ..., r_{n_R}] \\ p^T &= [1, p_1, ..., p_{n_P}, 0, ..., 0] \end{aligned} \tag{3.30}$$

and

$$M = \begin{bmatrix} 1 & 0 & ... & 0 & 0 & ... & ... & 0 \\ a_1 & 1 & & . & b'_1 & & & \\ a_2 & & & 0 & b'_2 & & & b'_1 \\ & & & 1 & . & & & b'_2 \\ & & & a_1 & . & & & . \\ a_{n_A} & & & a_2 & b'_{n_B} & & & . \\ 0 & & & . & 0 & . & . & . \\ 0 & ... & 0 & a_{n_A} & 0 & 0 & 0 & b'_{n_B} \end{bmatrix}. \quad (3.31)$$

Here, $b'_i = 0$ *for* $i = 0, 1, ..., d$ and $b'_i = b_{i-d}$ *for* $i \geq d+1$ and:

$$\begin{aligned} n_A &= degA(z^{-1}) \\ n_B &= degB(z^{-1}) \\ n_S &= degS(z^{-1}) = n_B + d - 1 \\ n_R &= degR(z^{-1}) = n_A - 1 \\ n_P &= degP(z^{-1}) \leq n_A + n_B + d - 1. \end{aligned} \quad (3.32)$$

3.2.3.2 Tracking

The reference model $H_m = \frac{B_m}{A_m}$ can be used to have an output y that follows a desired trajectory y^* each time a reference r is changed. This tracking model H_m can take the form of a second order system with desired ω_0 and ζ. This leads to choose $T(z^{-1})$ to have a unit static gain between y^* and y and to compensate between regulation dynamics defined by $P(z^{-1})$ and tracking dynamics defined by the poles of the reference model (Am):

$$T(z^{-1}) = G * P(z^{-1}), \quad (3.33)$$

where

$$G = \begin{cases} 1/B(1) & if \quad B(1) \neq 0 \\ 1 & if \quad B(1) = 0. \end{cases} \quad (3.34)$$

Having R, S, T and a desired trajectory y^* we obtain a control law of the form

$$S(z^{-1})u(t) + R(z^{-1})y(t) = T(z^{-1})y^*(t + d + 1). \quad (3.35)$$

3.2.4 Robust pole placement design

As already mentioned, we can say in a general way that the uncertainties in the dynamics of an off-road vehicle are related to the changes in speed and soil irregularities. Regarding the changes in speed, a gain scheduler system parametrization can be used. Regarding soil irregularities, one solution is to design a system that considers these uncertainties due to nonlinearities and time varying elements. Here we present a methodology for the design and implementation of a robust digital RST controller using pole placement. There are automatic methods for finding the parameters and shaping the sensitivity functions using convex optimization or H_∞ optimization (Landau and Karimi, 1998; Langer and Constantinescu, 1999; Langer and Landau, 1999). Nevertheless, this section presents a method to control the lateral position of an off-road vehicle by manually shaping the sensitivity functions (Landau and Zito, 2006) due to its simplicity compared to the automatic methods. This also allows for more flexibility in the design and a deep understanding of how the system reacts to changes in the controller design. Besides, one has also the flexibility to manually change the shaping that best fits different terrains. In order to consider uncertainties in our control design, the structure of Figure 3.6 can be extended into Figure 3.7 where $p(t)$ are disturbances, $b(t)$ is noise and $v(t)$ disturbances on the plant input.

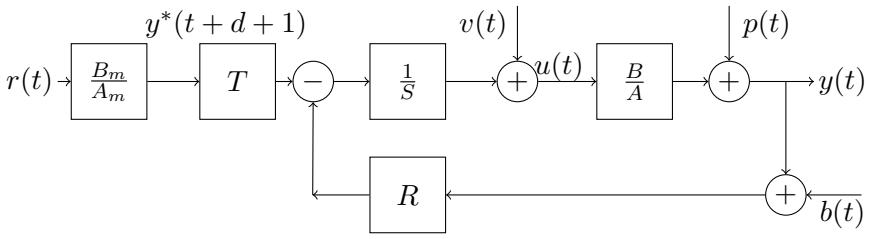


Figure 3.7 Digital canonical controller with disturbances and noise.

We can obtain then the following *output, input, noise*, and *disturbance sensitivity functions* respectively:

$$\begin{aligned} S_{yp}(z^{-1}) &= \frac{A(z^{-1})S(z^{-1})}{A(z^{-1})S(z^{-1}) + B(z^{-1})R(z^{-1})} \\ S_{up}(z^{-1}) &= \frac{-A(z^{-1})R(z^{-1})}{A(z^{-1})S(z^{-1}) + B(z^{-1})R(z^{-1})} \\ S_{yb}(z^{-1}) &= \frac{-B(z^{-1})R(z^{-1})}{A(z^{-1})S(z^{-1}) + B(z^{-1})R(z^{-1})} \\ S_{yv}(z^{-1}) &= \frac{B(z^{-1})S(z^{-1})}{A(z^{-1})S(z^{-1}) + B(z^{-1})R(z^{-1})}. \end{aligned} \tag{3.36}$$

As we can see, the common denominator happens to be the characteristic equation of the closed loop system of Eq. (3.25). The *noise sensitivity function* S_{yb} with a negative sign is also known in the literature as the *complementary sensitivity function* and from Eq. (3.36) we see that $S_{yp} - S_{yb} = 1$. Furthermore, the robustness of the closed loop system can be evaluated by the distance of the critical point $[-1, j0]$ to the Nyquist plot of the nominal plant and by the frequency characteristics of the sensitivity functions (Eq. (3.36)). The elements that help to evaluate this robustness are the following four margins: *gain, phase, delay* and *modulus margin*.

Especially the modulus margin is very important because it defines the maximum admissible value of the *output sensitivity function* S_{yp} and defines the tolerance with respect to nonlinear or time varying elements that may belong to the system. A good modulus margin also guarantees good values for the gain and phase margins. The modulus margin ΔM is the inverse of the maximum of the *output sensitivity function* S_{yp}. Typical values for a good modulus margin are $\Delta M \geq 0.5(-6dB)$.

Therefore, we use these margins to shape the frequency characteristics of the sensitivity functions S_{yp} and S_{up} and guarantee both robustness and closed loop performance. To be able to shape the sensitivity functions using pole placement, we have to introduce fixed parts to the Bezout polynomial equation (Eq. (3.27))

as follows:

$$P(z^{-1}) = A(z^{-1})H_S(z^{-1})S'(z^{-1}) + z^{-d}B(z^{-1})H_R(z^{-1})R'(z^{-1}). \quad (3.37)$$

The solution to Eq. (3.37) is found in the same way Eq. (3.27) is solved with Eq. (3.29), after replacing $A(z^{-1})$ by $A'(z^{-1}) = A(z^{-1})H_S(z^{-1})$ and $B(z^{-1})$ by $B'(z^{-1}) = B(z^{-1})H_R(z^{-1})$ with the following new conditions:

$$\begin{aligned} n_P &= degP(z^{-1}) \leq n_A + n_{H_S} + n_B + n_{H_R} + d - 1 \\ n'_S &= degS'(z^{-1}) = n_B + n_{H_R} + d - 1 \\ n'_R &= degR'(z^{-1}) = n_A + n_{H_S} - 1. \end{aligned} \quad (3.38)$$

3.2.4.1 Shaping the *input sensitivity function* S_{up}

The *input sensitivity function* reflects the actuator stress, therefore, we should limit its magnitude at high frequencies. By plotting the inverse of its magnitude $|S_{up}(e^{-j\omega})|$, we check and impose and upper bound maximum by imposing the prefixed value $H_R(z^{-1})$ presented in Eq. (3.39), which e.g. makes $|S_{up}| = 0$ at $0.5f_s$ by choosing $\beta = 1$, where f_s is the sampling frequency.

$$H_R(z^{-1}) = 1 + \beta z^{-1}; \qquad 0 < \beta \leq 1. \quad (3.39)$$

3.2.4.2 Shaping the *noise sensitivity function* S_{yb}

The *noise sensitivity function* is related to the delay margin $\Delta\tau$ and considering $\Delta\tau = T_s$ we impose a frequency template of a digital integrator of the form

$$|S_{yb}^{-1}(z^{-1})|dB < -20log|1 - z^{-1}|; \quad z = e^{jw}; \qquad 0 \leq \omega \leq \pi. \quad (3.40)$$

3.2.4.3 Shaping the output *sensitivity function* S_{yp}

We already mentioned a typical value for a good modulus margin $(\Delta M \geq 0.5(-6dB))$. Imposing that to the *output sensitivity function* we have that

$$|S_{yp}(z^{-1})| \leq -\Delta M; \quad z = e^{jw}; \qquad 0 \leq \omega \leq \pi. \quad (3.41)$$

Furthermore, we can use the complementary function defined by $S_{yp}(z^{-1}) = 1 + S_{yb}(z^{-1})$ and with the use of triangle inequality impose an upper and lower delay margin defined by the following equation:

$$1 - |S_{yb}(z^{-1})| \leq |S_{yp}(z^{-1})| \leq 1 + |S_{yb}(z^{-1})|. \tag{3.42}$$

By substituting Eq. (3.40) into (3.42) we obtain Eq. (3.43).

$$1 - |1 - z^{-1}|^{-1} \leq |S_{yp}(z^{-1})| \leq 1 + |1 - z^{-1}|^{-1}. \tag{3.43}$$

Again, in order to find a closed loop function that fits inside of the upper and lower bounds of the template defined by Eqs. (3.41) and (3.42), we can pick specified parts for H_S. For example making $H_S = 1 - z^{-1}$ we introduce a zero at the zero frequency allowing perfect rejection of constant disturbances.

3.2.5 Controller design

Section 3.2.4 presented a robust pole placement design in Figure 3.7 for the control of the vehicle position with a model described by $\frac{B}{A}$. Therefore, the *Controller Design* will use the model of the yaw rate delivered by the *Estimator* and the forward speed delivered by the *Forward Dynamics* to first calculate the lateral position model $\frac{z^{-d}B_y}{A_y}$ and then calculate the vehicle model as follows:

$$\frac{A}{B} = \frac{z^{-d}B_{\dot{\beta}}}{A_{\dot{\beta}}} \cdot \frac{z^{-d}B_y}{A_y}. \tag{3.44}$$

Once having the model of the vehicle, it will use the regulation dynamics ω_r, ζ_r, α_2 and α_2 for the real-time calculation of the control parameters R, S and T as described in the previous section. Finally, the tracking dynamics will be also calculated in real-time with the parameters ω_t and ζ_t delivered by the *Gain Schedule* as well.

3.3 Optimal digital control

This section presents and describes the third method proposed in this research as a reference, with the main objective of comparing the previous two methods. This approach is picked as a benchmark because, apart from the intrinsic benefits of the Linear Quadratic Regulator (LQR), there has been some results that demonstrate attractive stability properties and robustness against uncertainties by using optimal controllers (Chen, 2014; Chen and Holohan, 2014a,b; Snider, 2009). Therefore, one option to deal with the mentioned changes in the dynamics of off-road vehicles is the implementation of an observer-based optimal digital controller.

An extended version of the optimal controller of Figure 2.10 is illustrated in Figure 3.8. It contains tracking dynamics (K) and it is adapted for the problem of controlling the lateral position (y) of a terrain vehicle. In this case, the *Forward Dynamics* are measuring the vehicle speed V_x to be used by the *Controller Design* which is in charge of the real-time calculation of the *Observer*, the state feedback F and the tracking dynamics K. The output of the *Observer* contains the estimated or "observed" state variables of the *Vehicle* ($\hat{x}$) which are used for the pole placement by multiplying them with the state feedback. The control signal u is the desired curvature which is the inverse of the turning radius (1/km) and the set point r is the desired lateral position in meters.

3.3.1 Observer-based control

Figure 3.9 depicts the detailed part of the observer-based optimal controller of the system presented in Figure 3.8. As it can be seen, the observer is constructed with a copy of the plant plus the estimator gain L for correcting the error between the measured output and the output produced by the observer. As already explained, the signal to be controlled is the lateral position

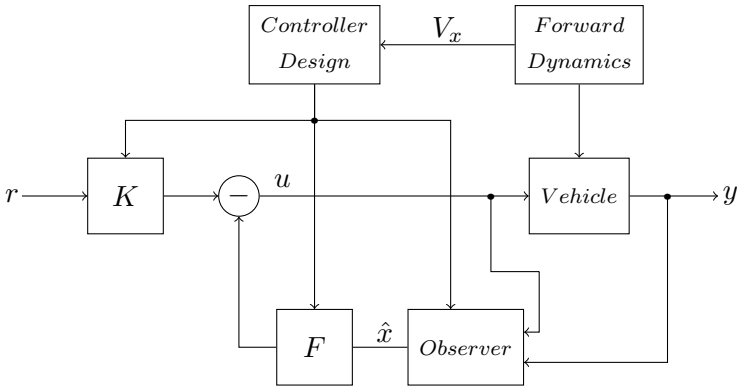


Figure 3.8 Block diagram of an observer-based controller for terrain vehicles.

y, which is not stable in open loop since a step in the steering input u will keep the vehicle turning in circles. The system is closed with an *Observer* and a state feedback F for the regulation dynamics. For the tracking dynamics, the gain K is added to be able to follow a reference lateral position r, i.e. zero when the vehicle drives along the desired path and one for driving 1 m to the left, parallel to the desired path (see Fig. 4.2). Negative values used for the reference have the meaning of driving on the right side of the desired path.

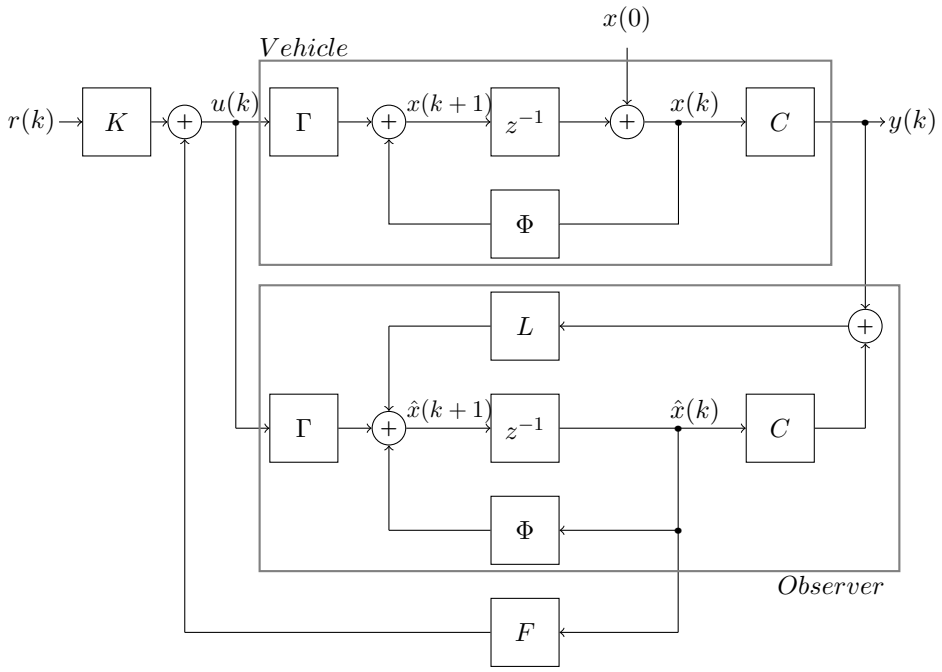


Figure 3.9 Block diagram of an observer-based state space controller.

3.3.1.1 Regulation

Generally speaking, the conditions for the design of the observer-based controller are that the system (the *Vehicle*) is observable and controllable. This means that the observer should be able to estimate all the state variables of the system and that the feedback should be able to bring all the poles to a desired place. To ensure this, the conditions presented here without proof are that the observability and the controllability matrices have full rank (Eqs. (3.45) and (3.46) respectively).

$$rank \begin{pmatrix} C \\ C\Phi \\ C\Phi^2 \\ \vdots \\ C\Phi^{(n-1)} \end{pmatrix} = n. \tag{3.45}$$

$$rank(\Phi \quad \Phi\Gamma \quad \Phi^2\Gamma \quad ... \quad \Phi^n\Gamma) \quad = \quad n. \tag{3.46}$$

Where n is the number of state variables. Then, using the linear regulation problem we can find the solution to the system in Figure 3.9 by first finding the state feedback $u = F \cdot x$ that minimizes the performance index

$$J_{LQG} = \frac{1}{2}\sum_{k=0}^{\infty}[x^T(k)Qx(k) + u^T(k)Ru(k)], \tag{3.47}$$

where the optimal solution is

$$F = -(R + \Gamma^T \cdot P_f \cdot \Gamma)^{-1} \cdot \Gamma^T \cdot P_f \cdot \Phi \tag{3.48}$$

and P_f is the solution to the Riccati equation

$$\begin{aligned} P_f &= Q + \Phi^T \cdot P_f \cdot \Phi \\ &- \Phi^T \cdot P_f \cdot \Gamma \cdot (R + \Gamma^T \cdot P_f \cdot \Gamma)^{-1} \cdot \Gamma^T \cdot P_f \cdot \Phi. \end{aligned} \tag{3.49}$$

For the *Observer*, the state estimate L is equivalent to the state feedback F by making the following replacements:

$$\begin{aligned} Q \to Q_e; \quad R \to R_e; \quad \Phi \to \Phi^T; \\ \Gamma \to C^T; \quad F \to L; \quad P_f \to P_l. \end{aligned} \tag{3.50}$$

Therefore, by duality the optimal estimator gain is

$$L = -((R_e + C \cdot P_l \cdot C^T)^{-1} \cdot C \cdot P_l \cdot \Phi^T)^T, \tag{3.51}$$

where P_l is the solution to the Riccati equation

$$\begin{aligned} P_l &= Q_e + \Phi \cdot P_l \cdot \Phi^T \\ &- \Phi \cdot P_l \cdot C^T \cdot (R_e + C \cdot P_l \cdot C^T)^{-1} \cdot C \cdot P_l \cdot \Phi^T. \end{aligned} \tag{3.52}$$

Here Q, Q_e, R and R_e can be used as tuning parameters to deal with uncertainties in the system. The details about how to find the solution to this problem are described in Åström and Wittenmark (1997) and Ogata (1995).

3.3.1.2 Tracking

To achieve zero steady-state error, i.e. the system output y equals the reference input r in steady state, the static gain should be 1. This can be done by finding the static gain of the system and multiplying it by its inverse. For instance, the static gain from r to y for the system in Figure 3.9 is $K \cdot C(\Phi + \Gamma F - I)^{-1}\Gamma$. Therefore, we can choose

$$K = \frac{1}{C(\Phi + \Gamma F - I)^{-1}\Gamma}. \tag{3.53}$$

This requires calculating the inverse of the matrix $(\Phi + \Gamma F - I)$ which is assumed to be non-singular since we expect a constant static gain for a stable system. Nevertheless, calculating the inverse of a matrix could consume a lot of computing resources. An alternative to solve this problem is by assuming that in steady state, the state vector x as well as the control input u will take the constant values x_{ss} and u_{ss} respectively and that $y_{ss} = r_{ss}$.

Therefore, in steady state the system will take the form

$$\begin{aligned} x_{ss} &= \Phi x_{ss} + \Gamma u_{ss} \\ r_{ss} &= C x_{ss} \end{aligned} \tag{3.54}$$

and for the system of Figure 3.9 the control law will take the form

$$u_{ss} = F x_{ss} + K r_{ss}, \tag{3.55}$$

since for an observable system we can assume that $\hat{x}_{ss} \approx x_{ss}$. We can relate the steady-state vector to the constant reference input as $x_{ss} = K_x r_{ss}$ and the steady-state control input that keeps x at x_{ss} as $u_{ss} = K_u r_{ss}$. Rewriting then the control law of Eq. (3.55) we have that $K_u r_{ss} = F K_x r_{ss} + K r_{ss}$ and solving for K we find that

$$K = F \cdot K_x + K_u. \tag{3.56}$$

To find K_x and K_u we use the same relation as for the control law and rewriting Eq. (3.54) we have that

$$\begin{aligned} K_x r_{ss} &= \Phi K_x r_{ss} + \Gamma K_u r_{ss} \\ r_{ss} &= C K_x r_{ss}. \end{aligned} \tag{3.57}$$

Solving for K_x and K_u we then find the solution to the following equation without using the inverse by applying the Gauss elimination or the Gauss-Jordan method:

$$\begin{bmatrix} 0 \\ 1 \end{bmatrix} = \begin{bmatrix} \Phi - I & \Gamma \\ C & 0 \end{bmatrix} \begin{bmatrix} K_x \\ K_u \end{bmatrix}. \tag{3.58}$$

3.3.2 Controller design

Since the model of the vehicle depends on the forward speed V_x, the *Controller Design* is in charge of calculating the vehicle model in real-time using the measured forward speed V_x. This model will be different depending on the type of vehicle we are working with. Once a vehicle model is available, it can be used in the *Observer* and for the calculation of the state feedback F,

the state estimate L and tracking gain K with the help of Eqs. (3.48), (3.51) and (3.56) respectively.

3.4 Summary of methods

In this chapter, three different lateral position digital controllers are presented to address the problem of time varying dynamics of off-road vehicles. The first one is based on MDPP and uses a second order transfer function as a reference. A recursive lest-squares estimator is used to identify the changes in the yaw rate dynamics in real time. For the speed adaptation, a gain scheduler is in charge of changing the PID parameters of the lateral position controller. The second controller is an improvement of the first one. Although it is also based on pole placement, it introduces a robust design. Additionally, the pole placement is based on regulation dynamics. For the lateral position, a tracking model is also introduced. In contrast to the first controller, it is directly applied to the lateral position and no PID parameters have to be tuned. Instead, a second order model is used as a reference for the desired closed loop response, leaving only two variables to be tuned: ω_t for the tracking response and ω_r for regulation dynamics. For the gain scheduler, a linear interpolation function can be used for those variables ($\omega_{t,r} = f(V_x)$). The rest of the control parameters can be set fixed in the implementation process (ζ_r, ζ_t, α_1 and α_2). The third controller is an optimal one, with an observer to filter measurement noise and disturbances. It is picked to be used as a benchmark due to its good characteristics and its previous results used for path following (Chen, 2014; Chen and Holohan, 2014a,b; Snider, 2009). Each controller has its pros and cons and they are discussed in Table 3.1.

It can be said in general, that even though the robust controller works with more complex theory, it might produce better results in terms of all the different scenarios that could be

covered. Also in terms of implementation, the tuning should be straightforward since in theory, only the natural frequencies (ω_t for the tracking model and ω_r for the regulation model) will be tuned by the end user.

Table 3.1 Summary of controllers.

Controller	Pros, cons and comments
Cascade STR	**Pros:** *The model is reduced to a 2nd order system using MDPP. *The user can change the desired behavior of the yaw rate. **Cons:** *Tuning the PID is time consuming. *Using recursive lest-squares might not always find a stable system. *Online calculations need few resources. **Comments:** Using a reduced model can generate overshoot in some vehicles e.g. where the steering increases the order of the system, reducing the accuracy of the controller.
Robust STR	**Pros:** *No need for tuning a PID. *Tracking and regulation behaviours can be tuned by the user. *Frequency response can be tuned as well. **Cons:** *Technically more complex. *Requires more know-how. *Not limited to 2nd order system. *Online gradient estimation has a higher level of complexity. *Online estimation might need more resources for faster identification. **Comments:** Requires more knowledge and there are more factors to take into account since the fixed parts can take many forms.
Optimal LQR	**Pros:** *The energy used for the actuators is less. *No high frequency parts in the control signal. *Low level of complexity. *Straightforward approach. **Cons:** *Less flexibility in tuning the behaviour of the vehicle. *Online calculations might need a lot of resources. *Slower response against disturbances. **Comments:** The system requires a model. Nevertheless, it could be a kinematic approximation making the design easier but less accurate.

Chapter 4

Experimental setup and results

The first two sections of this chapter introduce the models of the two different vehicles used for the experiments: a tractor-implement system and a skid-steering robot. Then, the results of the experiments of the different methods explained in Chapter 3 are presented respectively. The first test for the cascade self-tuning regulator was done using the model of the tractor-implement system and, since performing the tests using a real tractor with an implement that changes its cornering stiffness on a field is very complicated due to the resources needed, a 3D real-time simulations was used for validation instead. From the lessons learned with the cascade self-tuning regulator, the robust self-tuning regulator was simulated and implemented in a skid-steered robot which was available in real form for the validation of the method. Finally, the optimal controller was used as a benchmark for comparison with the other two methods presented.

4.1 Vehicle models

The two vehicle models used to verify that the controllers proposed in this research can be universally applied to different vehicles were the tractor-implement model and a kinematic-based skid-steering model.

4.1.1 Tractor-implement model

To be able to understand the system, an analysis of the vehicle dynamics, kinematics and its root-locus has to be performed. The dynamics and kinematics lead to the equation of motion of the system and its root-locus tells the response of the system according to the changes in the velocity, soil and hitch forces. This is essential for finding the model to be used as a reference for the controller design. If the dynamics of the reference model are too fast, the closed-loop system will experience steady state oscillations and even instability and if the dynamics are too slow, the controller will have a lot of room for improvement.

Figure 4.1 illustrates a 3-wheeled tractor-implement bicycle model. Here, $\dot{\beta}$ is the yaw rate around the center of gravity, δ is the steering angle and α_f, α_r and α_h are the front, rear and hitch slip angles, respectively. The distances from the front and rear axis to the center of gravity are a and b, respectively, and c is the distance from the rear axis to the hitch. Lateral forces at front, rear and hitch tires are represented by F_f, F_r and F_h, respectively, and by assuming constant longitudinal velocity (V_x), the longitudinal acceleration is null and the longitudinal forces are neglected. Therefore, the yaw rate dynamics of the model represented by Figure 4.1, can be expressed by analysing the simplified lateral dynamics with Eq. (4.1).

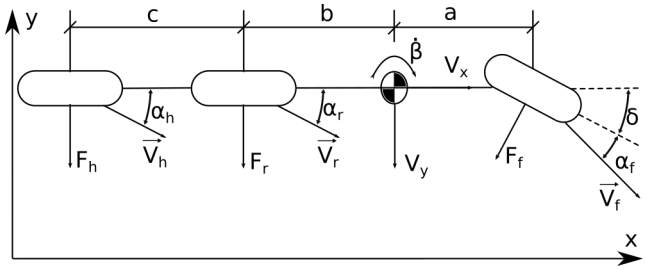


Figure 4.1 Bicycle model of a tractor with an implement.

$$\begin{aligned} \sum F_y &= m \cdot a_y \\ \sum M_{CG} &= I_z \cdot \ddot{\beta}. \end{aligned} \tag{4.1}$$

From a kinematics point of view, and since the system has null longitudinal acceleration, the lateral acceleration is expressed in Eq. (4.2).

$$a_y = \dot{V}_y + \dot{\beta} \cdot V_x. \tag{4.2}$$

Substituting into Eq. (4.1) and using the small angle approximation, we obtain the following simplified equation of motion:

$$\begin{aligned} m \cdot (\dot{V}_y + \dot{\beta} \cdot V_x) &= F_f + F_r + F_h \\ I_z \cdot \ddot{\beta} &= a \cdot F_f - b \cdot F_r - (c+b) \cdot F_h. \end{aligned} \tag{4.3}$$

Assuming constant lateral forces, their relationship to the slip angles are given in Eq. (4.4) (Derrick and Bevly, 2009; Gillespie, 1992).

$$\begin{aligned} F_f &= -C_{\alpha_f} \cdot \alpha_f \\ F_r &= -C_{\alpha_r} \cdot \alpha_r \\ F_h &= -C_{\alpha_h} \cdot \alpha_h, \end{aligned} \tag{4.4}$$

where C_{α_f}, C_{α_r} and C_{α_h} are the front, rear and hitch cornering stiffness and vary depending on the conditions and types of soil. By substituting (4.4) into (4.3), the equation of motion looks as follows:

$$\begin{aligned} m \cdot (\dot{V}_y + \dot{\beta} \cdot V_x) &= -C_{\alpha_f} \cdot \alpha_f - C_{\alpha_r} \cdot \alpha_r - C_{\alpha_h} \cdot \alpha_h \\ I_z \cdot \ddot{\beta} &= -a \cdot C_{\alpha_f} \cdot \alpha_f + b \cdot C_{\alpha_r} \cdot \alpha_r + (c+b) \cdot C_{\alpha_h} \cdot \alpha_h. \end{aligned} \tag{4.5}$$

Assuming a rigid body, we can say in general that the absolute lineal velocity at any of its points can be expressed as the lineal velocity of its center of gravity plus the velocity of the point with respect to its center of gravity and so we get Eq. (4.6), where each velocity is represented by its x and y

coordinates.

$$\begin{aligned} \frac{V_{f_y}}{V_{f_x}} &= \frac{V_y + \dot{\beta} \cdot a}{V_x} \\ \frac{V_{r_y}}{V_{r_x}} &= \frac{V_y - \dot{\beta} \cdot b}{V_x} \\ \frac{V_{h_y}}{V_{h_x}} &= \frac{V_y - \dot{\beta} \cdot (b+c)}{V_x}. \end{aligned} \tag{4.6}$$

We can also observe from Figure 4.1 that

$$\begin{aligned} \tan(\alpha_f + \delta) &= \frac{V_{f_y}}{V_{f_x}} \\ \tan(\alpha_r) &= \frac{V_{r_y}}{V_{r_x}} \\ \tan(\alpha_h) &= \frac{V_{h_y}}{V_{h_x}}, \end{aligned} \tag{4.7}$$

and by substituting Eq. (4.6) into Eq. (4.7), applying the small angle approximation and solving for α, the relationship between the slip angles and the longitudinal and lateral velocities of the center of gravity is found as follows:

$$\begin{aligned} \alpha_f &= \frac{V_y + \dot{\beta} \cdot a}{V_x} - \delta \\ \alpha_r &= \frac{V_y - \dot{\beta} \cdot b}{V_x} \\ \alpha_h &= \frac{V_y - \dot{\beta} \cdot (b+c)}{V_x}, \end{aligned} \tag{4.8}$$

and substituting Eq. (4.8) into Eq. (4.5) results into

$$\begin{aligned} m \cdot (\dot{V}_y + \dot{\beta} \cdot V_x) = &- C_{\alpha_f} \cdot (\frac{V_y + \dot{\beta} \cdot a}{V_x} - \delta) \\ &- C_{\alpha_r} \cdot \frac{V_y - \dot{\beta} \cdot b}{V_x} - C_{\alpha_h} \cdot \frac{V_y - \dot{\beta} \cdot (b+c)}{V_x} \\ I_z \cdot \ddot{\beta} = &- a \cdot C_{\alpha_f} \cdot (\frac{V_y + \dot{\beta} \cdot a}{V_x} - \delta) \\ &+ b \cdot C_{\alpha_r} \cdot \frac{V_y - \dot{\beta} \cdot b}{V_x} + (c+b) \cdot C_{\alpha_h} \cdot \frac{V_y - \dot{\beta} \cdot (b+c)}{V_x}. \end{aligned} \tag{4.9}$$

By introducing the following new variables C_1, C_2 and C_3 such as

$$\begin{aligned} C_1 &= -a \cdot C_{\alpha_f} + b \cdot C_{\alpha_r} + (b+c) \cdot C_{\alpha_h} \\ C_2 &= C_{\alpha_f} + C_{\alpha_r} + C_{\alpha_h} \\ C_3 &= a^2 \cdot C_{\alpha_f} + b^2 \cdot C_{\alpha_r} + (b+c)^2 \cdot C_{\alpha_h}, \end{aligned} \tag{4.10}$$

Eq. (4.9) can be rewritten into the following state space equation of motion with state variables V_y and $\dot{\beta}$:

$$\begin{bmatrix} \dot{V}_y \\ \ddot{\beta} \end{bmatrix} = \begin{bmatrix} -\frac{C_2}{m \cdot V_x} & \frac{C_1}{m \cdot V_x} - V_x \\ \frac{C_1}{I_z \cdot V_x} & -\frac{C_3}{I_z \cdot V_x} \end{bmatrix} \cdot \begin{bmatrix} V_y \\ \dot{\beta} \end{bmatrix} + \begin{bmatrix} \frac{C_{\alpha_f}}{m} \\ \frac{a \cdot C_{\alpha_f}}{I_z} \end{bmatrix} \cdot \delta. \tag{4.11}$$

Using Laplace transform, the transfer function between the steering angle and the yaw rate can be found by solving the following equation:

$$\frac{\dot{\beta(s)}}{\delta(s)} = \frac{C \cdot adj(s \cdot I - A) \cdot B}{det(s \cdot I - A)}, \tag{4.12}$$

where $C = [0 \quad 1]$ since it is solved only for $\dot{\beta}$, and so one gets that

$$\frac{\dot{\beta(s)}}{\delta(s)} = \frac{b_2 \cdot s + (b_1 \cdot a_{21} - b_2 \cdot a_{11})}{(s - a_{11})(s - a_{22}) - a_{12} \cdot a_{21}}, \tag{4.13}$$

with

$$\begin{aligned} a_{11} &= -\frac{C_2}{m \cdot V_x} \\ a_{12} &= \frac{C_1}{m \cdot V_x} - V_x \\ a_{21} &= \frac{C_1}{I_z \cdot V_x} \\ a_{22} &= -\frac{C_3}{I_z \cdot V_x} \\ b_1 &= \frac{C_{\alpha_f}}{m} \\ b_2 &= \frac{a \cdot C_{\alpha_f}}{I_z}. \end{aligned} \tag{4.14}$$

Finally, the simplified transfer function of the yaw rate with respect to the steering angle is presented in Eqs. (4.15) and (4.16).

$$G_{\dot{\beta}} = \frac{n_1 s + n_0}{s^2 + d_1 s + d_0}. \tag{4.15}$$

$$
\begin{aligned}
n_0 &= \frac{C_{af} \cdot C_1 + a \cdot C_{af} \cdot C_2}{I_z \cdot m \cdot V_x} \\
n_1 &= \frac{a \cdot C_{\alpha_f}}{I_z} \\
d_0 &= \frac{C_2 \cdot C_3 - C_1^2}{I_z \cdot m \cdot V_x^2} + \frac{C_1}{I_z} \\
d_1 &= \frac{C_2}{m \cdot V_x} + \frac{C_3}{I_z \cdot V_x}.
\end{aligned}
\tag{4.16}
$$

4.1.2 Skid-steering simplified model

Besides the aforementioned changes in the vehicle dynamics due to changes in the speed and the interaction with the terrain, a skid-steered vehicle have some other challenges to be covered. There are intrinsic non-linearities related to the steering system of such vehicles that make the design of a controller a very complex task. Therefore, a simplified model of a skid-steered vehicle presented here will facilitate the use of some linear controllers.

Figure 4.2 shows a representation of a skid-steered robot where y is the lateral position to a desired path and it is also the variable to be controlled. The yaw rate of the robot (ω) and its linear velocity (V) are produced by the difference between the velocities of the right and left axis (V_R and V_L respectively), where β represents the angle of the velocity vector to the center of gravity. The angular velocities of the right and left wheels are represented by ω_R and ω_L, respectively, and the distance between axes is $2c$. The angle with respect to the desired path is represented by ν and r is the wheel radius.

From the kinematic point of view, we can define the yaw rate at the center of gravity and the linear velocity by the following equations:

$$V = r * \frac{\omega_L + \omega_R}{2}, \tag{4.17}$$

$$\omega = r * \frac{-\omega_L + \omega_R}{2c}. \tag{4.18}$$

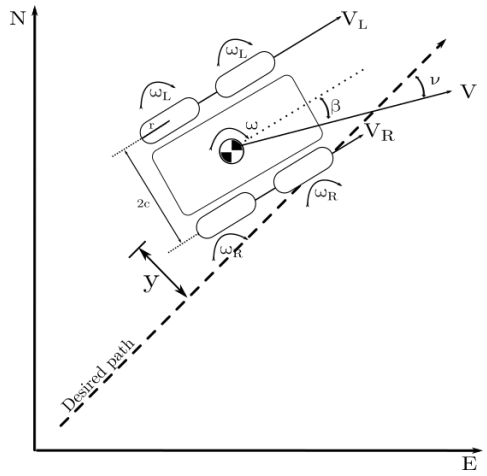


Figure 4.2 Skid-steering model with lateral position.

By rewriting Eq. (4.18) into right and left velocities we obtain Eq. (4.19). And by defining a time delay τ we find a first order yaw-rate transfer function presented in Eq. (4.20).

$$\omega = \frac{V_R - V_L}{2c} = \frac{\Delta V}{2c}. \tag{4.19}$$

$$G_\omega(s) = \frac{\omega(s)}{\Delta V(s)} = \frac{1/2c}{\tau \cdot s + 1}. \tag{4.20}$$

Finally using the z-transform we obtain the following digital transfer function:

$$G_\omega(z^{-1}) = \frac{b_{r_1} z^{-1}}{1 + a_{r_1} z^{-1}} \tag{4.21}$$

where

$$\begin{aligned} b_{r_1} &= (1/2c) * (1 - e^{-T_s/\tau}) \\ a_{r_1} &= -e^{-T_s/\tau}. \end{aligned} \tag{4.22}$$

Generally speaking, the lateral position y with respect to a *desired path* is related to the yaw rate of the vehicle in the form of Eq. (4.23), where β is the side slip angle and ν the course angle with respect to the *desired path* (Fig. 4.2) (Gartley and Bevly, 2008).

$$\begin{aligned} \dot{y} &= V \ sin(\nu) \\ \dot{\nu} &= \omega + \dot{\beta}. \end{aligned} \tag{4.23}$$

Neglecting side slip, linearising and using the small angle approximation we find the transfer function of the lateral position with respect to the yaw rate in Eq. (4.24).

$$G_y(s) = \frac{y(s)}{\omega(s)} = \frac{V}{s^2}. \tag{4.24}$$

Again, using the z-transform we obtain the following digital transfer function of the lateral position with respect to the yaw rate:

$$G_y(z^{-1}) = \frac{b_{l_0} + b_{l_1} z^{-1}}{1 + a_{l_1} z^{-1} + a_{l_2} z^{-2}}, \tag{4.25}$$

where

$$\begin{aligned} b_{l_0} &= V * T_s^2/2 \\ b_{l_1} &= V * T_s^2/2 \\ a_{l_1} &= -2 \\ a_{l_2} &= 1. \end{aligned} \tag{4.26}$$

Multiplying Eqs. (4.21) and (4.25) we obtain the digital model of the lateral position with respect to a *desired path* of a skid-steered robot to be used for our RST controller design (Eq. (4.27)).

$$G_{\frac{B}{A}} = \frac{b_2 z^{-2} + b_3 z^{-3}}{1 + a_1 z^{-1} + a_2 z^{-2} + a_3 z^{-3}}, \tag{4.27}$$

where

$$\begin{aligned} b_2 &= b_{r_1} * b_{l_0} \\ b_3 &= b_{r_1} * b_{l_1} \\ a_1 &= a_{l_1} + a_{r_1} \\ a_2 &= 1 + a_{l_1} * a_{r_1} \\ a_3 &= a_{r_1}. \end{aligned} \tag{4.28}$$

Finally, the following digital state space of third order can be used to represent our plant model:

$$\begin{aligned} x(k+1) &= \Phi x(k) + \Gamma u(k) \\ y(k) &= C x(k) \end{aligned} \tag{4.29}$$

where

$$\begin{aligned} \Phi &= \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ -a_3 & -a_2 & -a_1 \end{bmatrix} \\ \Gamma &= \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \\ C &= \begin{bmatrix} b_3 & b_2 & 0 \end{bmatrix}. \end{aligned} \tag{4.30}$$

This simplified first order model presented above can be used for the representation of a real agricultural skid-steered robot (Fig. 4.3) electrically powered with 4 wheels able to reach a maximal speed of 1.0 m/s. It has a weight of 40 kg and a state-of-the-art GNSS receiver with RTK-correction which provides global geographic position. The ECU with a 400 MHz processor, contains a gyroscope, WLAN module and a CAN interfaces for the different internal communications. The angular velocity of the motors is measured with hall sensors.

Figure 4.3 Skid-steered robot for corn seeding provided by AGCO GmbH | Fendt.

4.2 Cascade self-tuning regulator

This section presents the results of applying the first controller proposed in this research, to a tractor system with varying hitch forces. Since this method is based on a pole placement

design, an analysis of the pole-zero plots will illustrate the changes in the system dynamics and help to pick the desired poles for the characteristic polynomial of the closed-loop reference model. This analysis will also help to understand the behaviour of the vehicle according the changes in the speed or in the hitch forces or a combination of both.

From the system described in Eqs. (4.15) and (4.16), the changes in the system dynamics can be analysed by changing the velocity and cornering stiffness. Table 4.1 presents the data of a Fendt Tractor Vario 939 to be used as an example and for the sake of simplicity, the values used at the front and rear are approximated, whereas the values for the hitch are moving from 0 to 5000 to be able to test the adaptation of the method.

Table 4.1 Technical data of a Fendt tractor of the model Vario 939

Tractor data		
a	1.696	m
b	3.958	m
c	2.190	m
m	18000	kg
Izz	59312	kg-m
C_{af}	3600	N/deg
C_{ar}	6250	N/deg
C_{ah}	0-5000	N/deg
V_x	2-15	m/s

It can be seen from table 4.2 that at relative medium speed and cornering stiffness, there will be two poles and one zero. On pole with a higher response and the other pole, together with the zero, are placed in a lower response nearer to the origin. It can also be observed in the following Figures 4.4 and 4.5 that, on the one hand, as the cornering stiffness increases, so does the response in the negative half-plane for all poles and zeros (see tables 4.3 and 4.4). On the other hand, as the velocity increases, the response moves in the negative plane nearer to the center up to a place where the imaginary part of the poles increases along with the increments of the cornering stiffness

(see Fig. 4.5 and Tab. 4.4). This means that e.g. one will find the higher oscillations at a lineal velocity of 15 m/s with a cornering stiffness of around 3000 N/deg. In real conditions the maximum working speed will rarely reach the 25 km/h (Approx. 7 m/s) however, 15 m/s is used for this analysis.

Table 4.2 Poles-Zeros mapping (V_x = 5 m/s and C_{α_h} =1000 N/deg)

	Poles-Zeros		
C_{α_h}	P 02	P 01	Z 01
1000	-7.034	-0.475	-0.132

Contrary to Figure 4.5, a mapping of a varying velocity using the same cornering stiffness can be seen in Figure 4.6, which can be interpreted as increasing the velocity while the weight and soil contact at the implement stays constant. It can be seen that the steering system will start oscillating at a velocity of 12.5 m/s and the oscillations will increase as the velocity increases (Tab. 4.5).

Table 4.3 Poles-Zeros mapping C_{α_h} (100-5000 N/deg) @ V_x =2.5 m/s

	Poles-Zeros		
C_{α_h}	P 02	P 01	Z 01
100	-8.0627	-0.133	-0.126
814	-13.305	-0.304	-0.237
1528	-18.647	-0.377	-0.347
2242	-24.021	-0.416	-0.457
2956	-29.410	-0.441	-0.568
3670	-34.807	-0.458	-0.678
4384	-40.208	-0.471	-0.788

Table 4.4 Poles-Zeros mapping C_{α_h} (100-5000 N/deg) @ V_x =15 m/s

	Poles-Zeros		
C_{α_h}	P 02	P 01	Z 01
100	-1.281	-0.084	-0.021
814	-1.134-1.019i	-1.134+1.019i	-0.039
1528	-1.585-1.424i	-1.585+1.424i	-0.057
2242	-2.036-1.616i	-2.036+1.616i	-0.076
2956	-2.487-1.669i	-2.487+1.669i	-0.094
3670	-2.938-1.598i	-2.938+1.598i	-0.113
4384	-3.389-1.384i	-3.389+1.384i	-0.131

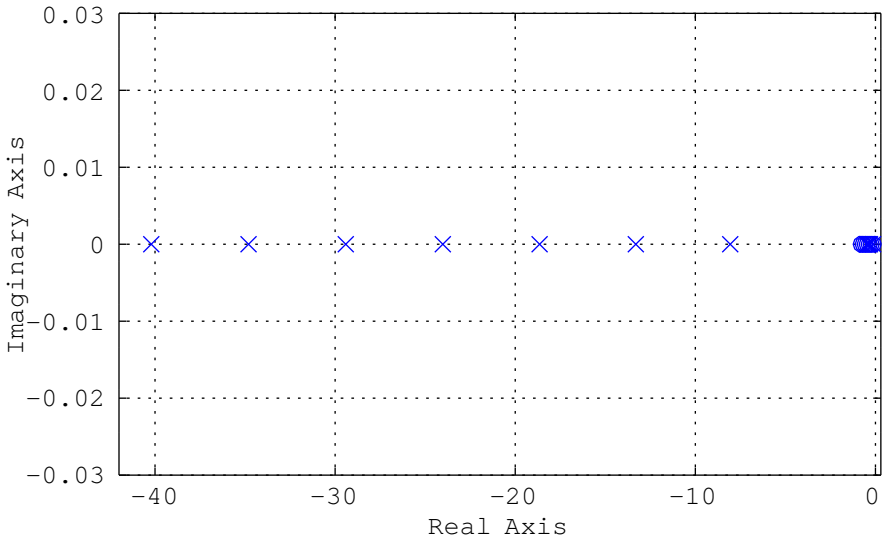


Figure 4.4 Poles-Zeros mapping of a tractor with an implement at 2.5 m/s with a varying hitch cornering stiffness (C_{α_h}) from 100 to 5000 N/deg.

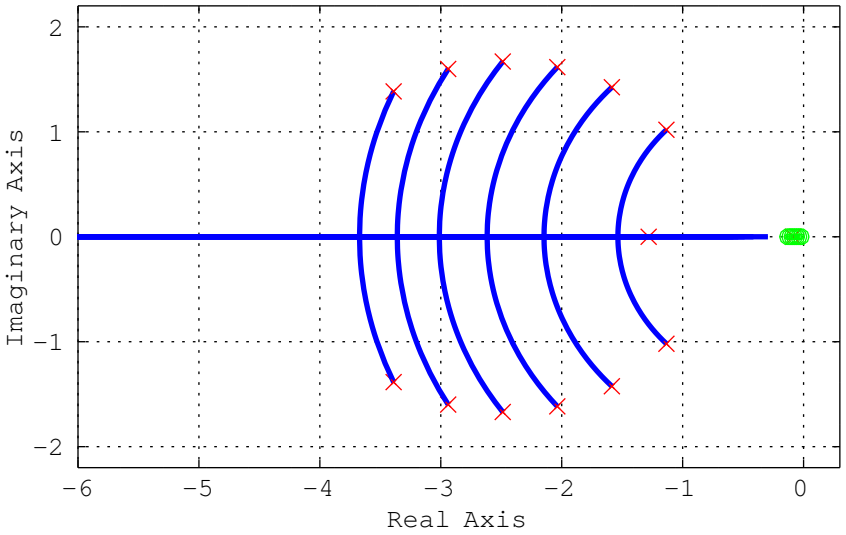


Figure 4.5 Poles-Zeros mapping of a tractor with an implement at 15 m/s with a varying hitch cornering stiffness (C_{α_h}) from 100 to 5000 N/deg.

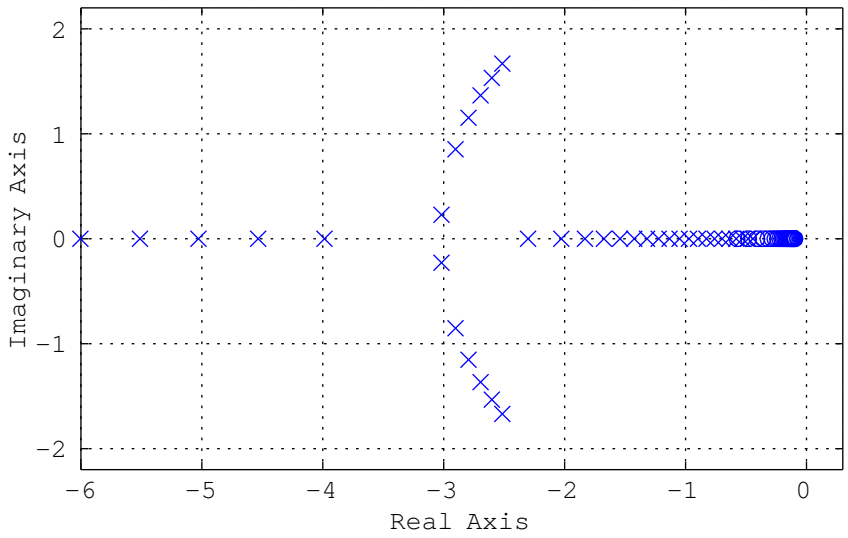


Figure 4.6 Poles-Zeros mapping of a tractor with an implement at C_{α_h} =3000 N/deg with a varying speed (V_x) from 2.5 to 15 m/s.

We can say in general, that an increasing value of cornering stiffness means that the implement gets more load. This means that achieving a certain yaw rate will be easier when driving with empty load. Also, the maximum steering angle that the machine is able to apply will limit the response of the system. All these factors, along with measurement disturbances and terrain changes, have to be taken into account in the design.

To validate the method presented in Section 3.1, two sets of tests were performed: one using a Matlab simulation and one using an Openframeworks-OpenGL environment. For the Matlab simulation, Eqs. (4.15) and (4.16) together with the data from table 4.1 were combined to generate the response for different values of V_x and $C_{\alpha h}$. This data was used to identify the system and apply the control algorithm. For the Openframeworks environment, and since measuring real online changes of the cornering stiffness is very complicated, an online simulation was implemented using a combination of an identified system of a Fendt Tractor Vario 939 with the model of Eqs. (4.15) and (4.16) (Fernandez et al., 2016). This was also programmed using threads for each component (steering, yaw rate, lateral position, etc.) in a similar way

Table 4.5 Poles-Zeros mapping V_x(2.5-15 m/s) @ C_{α_h} =3000 N/deg

	Poles-Zeros		
V_x	P 02	P 01	Z 01
2.5	-29.742	-0.442	-0.574
3.0	-24.672	-0.482	-0.479
3.5	-21.032	-0.529	-0.410
4.0	-18.284	-0.581	-0.359
4.5	-16.132	-0.637	-0.319
5.0	-14.395	-0.697	-0.287
5.5	-12.960	-0.760	-0.261
6.0	-11.750	-0.827	-0.239
6.5	-10.712	-0.897	-0.221
7.0	-9.808	-0.971	-0.205
7.5	-9.011	-1.049	-0.191
8.0	-8.299	-1.133	-0.179
8.5	-7.655	-1.222	-0.169
9.0	-7.065	-1.318	-0.159
9.5	-6.519	-1.424	-0.151
10.0	-6.004	-1.541	-0.143
10.5	-5.511	-1.675	-0.136
11.0	-5.028	-1.832	-0.130
11.5	-4.534	-2.027	-0.125
12.0	-3.986	-2.302	-0.119
12.5	-3.018-0.228i	-3.018+0.228i	-0.115
13.5	-2.902-0.852i	-2.902+0.852i	-0.110
13.5	-2.794-1.152i	-2.794+1.152i	-0.106
14.5	-2.695-1.366i	-2.695+1.366i	-0.102
14.5	-2.602-1.532i	-2.602+1.532i	-0.099
15.5	-2.515-1.668i	-2.515+1.668i	-0.095

as it is done inside a vehicle taking into account cycle times and separation of actuation and output signals. The analysis of both test-results are described in more detail in the following subsections.

4.2.1 Simulation

Using the data from table 4.1, the following closed-loop reference model system is to be used (Eq. (4.31)), which corresponds to a second order system with a natural frequency $\omega_n = 4$, a damping ratio $\zeta = 0.8$ and a sampling time of 40 ms.

$$\frac{y(q)}{u(q)} = \frac{0.022q}{q^2 - 1.751q + 0.773} \tag{4.31}$$

The goal is to make the closed-loop yaw rate dynamics, to follow the reference model of Eq. (4.31) using Algorithm 1. For that, Eq. (3.8) can be rewritten in the form of $R \cdot \delta = T \cdot \dot{\beta}_{des} - S \cdot \dot{\beta}_{mea}$, where R, S and T are the control parameters to be updated online.

The lower part of Figure 4.7 shows in blue a row of desired set-points ($\dot{\beta}_{des}$) to be followed by a tractor with an implement driving at 3.6 kmph. For this scenario, a cornering stiffness of 300 N/deg at the hitch represents an implement of relatively low load. Given a low initial value of the parameters to be identified (e.g. 0.1), the yaw rate of the vehicle is measured ($\dot{\beta}_{mea}$) and used to identify the tractor-implement system with the help of Eq. (3.6). Then, the identified parameters are used iteratively to calculate the new control parameters R, T and S (Eq. (3.11)). This can be noticed in the first seconds where the measured and the identified lines present some abrupt changes until the identified parameters converge and the correct control parameters are calculated stabilizing the system.

The upper part of the same figure presents the form of the control signal δ applied to the steering system of the vehicle. It can be observed that small steering angles (2 to 8 degrees) are enough to bring the vehicle to the desired yaw rates. Figure 4.8 presents the same vehicle driving at higher speed and having bigger load at the hitch. Here, the steering input to be applied has to be bigger to compensate for those changes.

Normally one could expect that at higher velocities, the vehicle is more sensitive to small changes in the steering input. Nevertheless, we can see in Figures 4.8 and 4.9 that as the cornering stiffness increases, so does the amplitude of the steering angle. Also, with higher values of cornering stiffness, the adaptive steering input increases gradually as a ramp to maintain the same yaw rate, whereas at lower values the input takes more the form of a step. As a result, this demonstrates that the

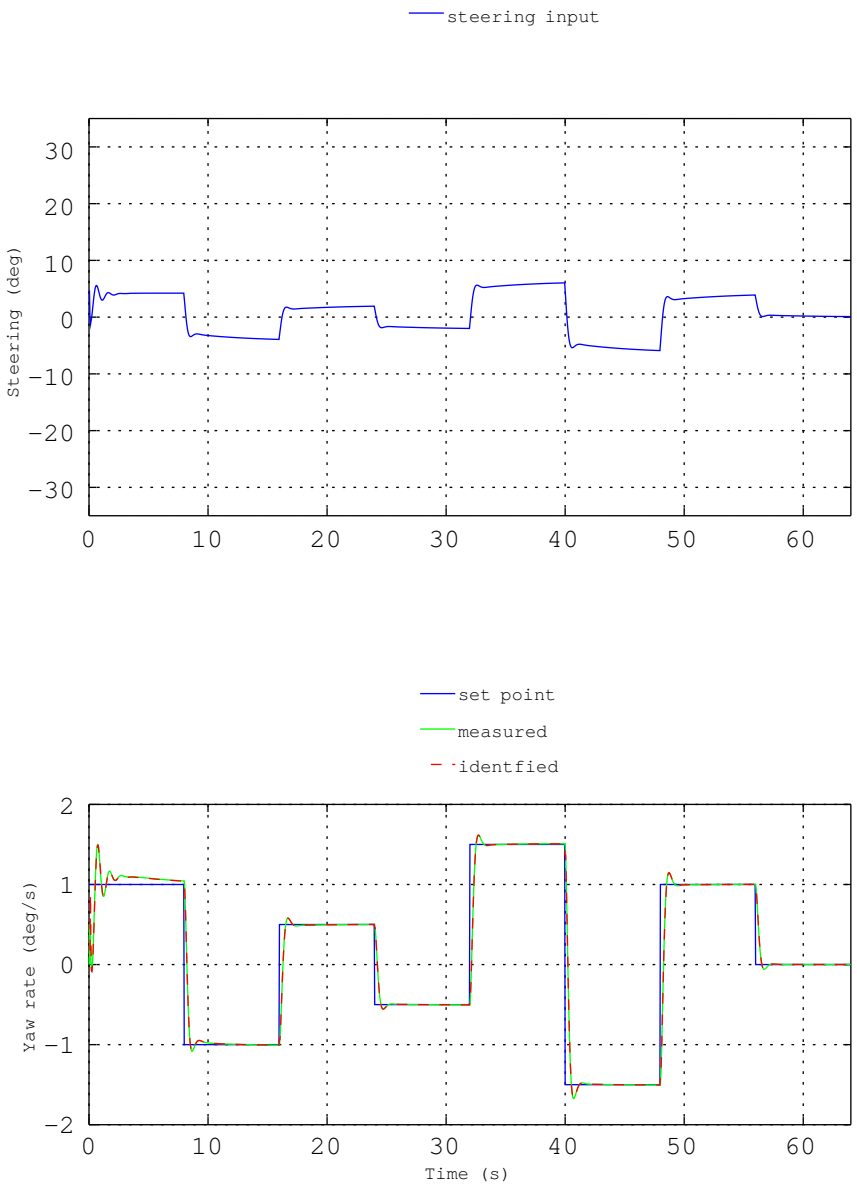


Figure 4.7 Yaw rate response for $V_x = 2$ m/s and $C_{\alpha h} = 300$ N/deg. (a) Steering input (deg). (b) Yaw rate (deg/s).

algorithm adapts the control signal quite well to the different velocities and hitch cornering stiffness. It is very important to mention, that the adaptation depends on the identified system. For instance, Figure 4.9 depicts that the measured signal follows quite well the identified one. Nevertheless, the identified signal deviates from the original one producing a steady state error.

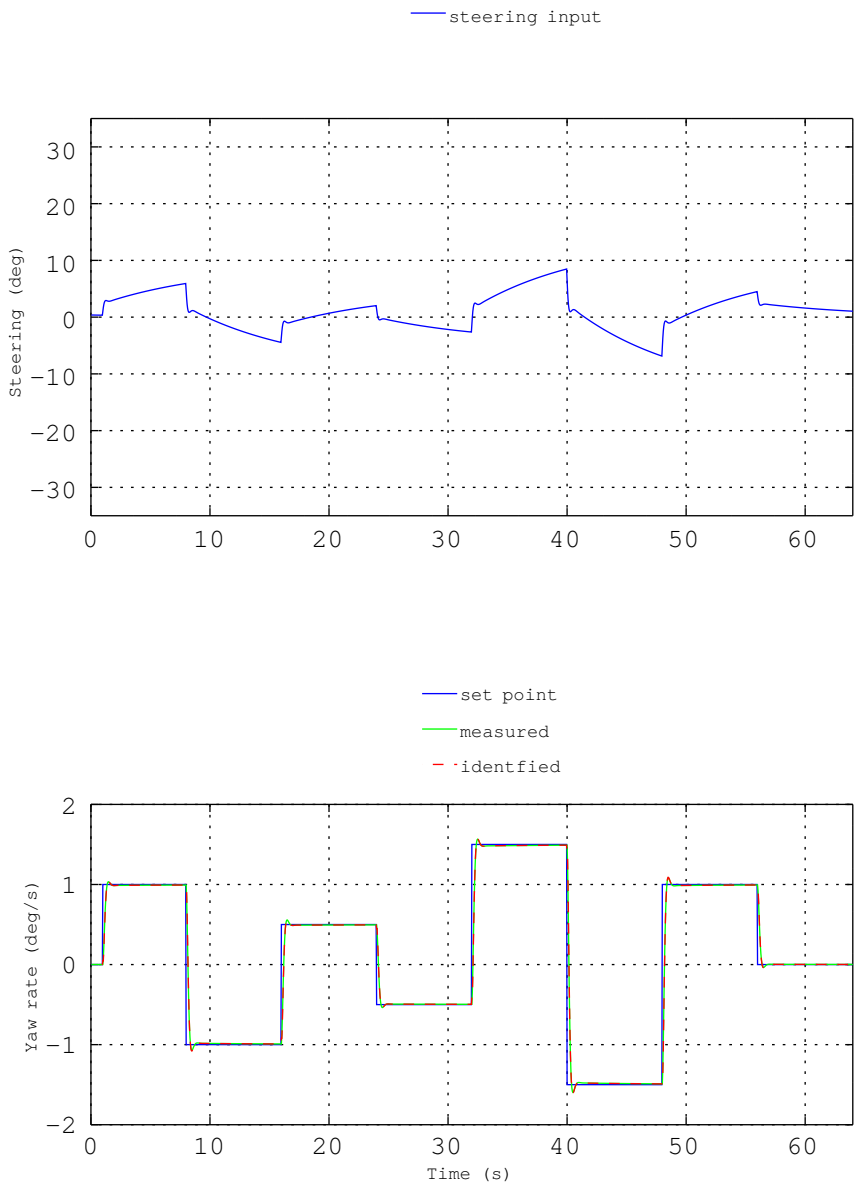


Figure 4.8 Yaw rate response for V_x =5 m/s and $C_{\alpha h}$ =1000 N/deg. (a) Steering input (deg). (b) Yaw rate (deg/s).

4.2.2 Validation

In contrast to the Matlab simulation, the 3D environment with Openframeworks presented here, also includes the hydraulic-steering dynamics. This would increase the order of the plant from 2nd to 5th, since the hydraulic steering system is represented by a third order system. For the steering input, curvature was used since a real tractor uses it as the control signal. Figures 4.10 to 4.12 present some results of the Openframeworks validation.

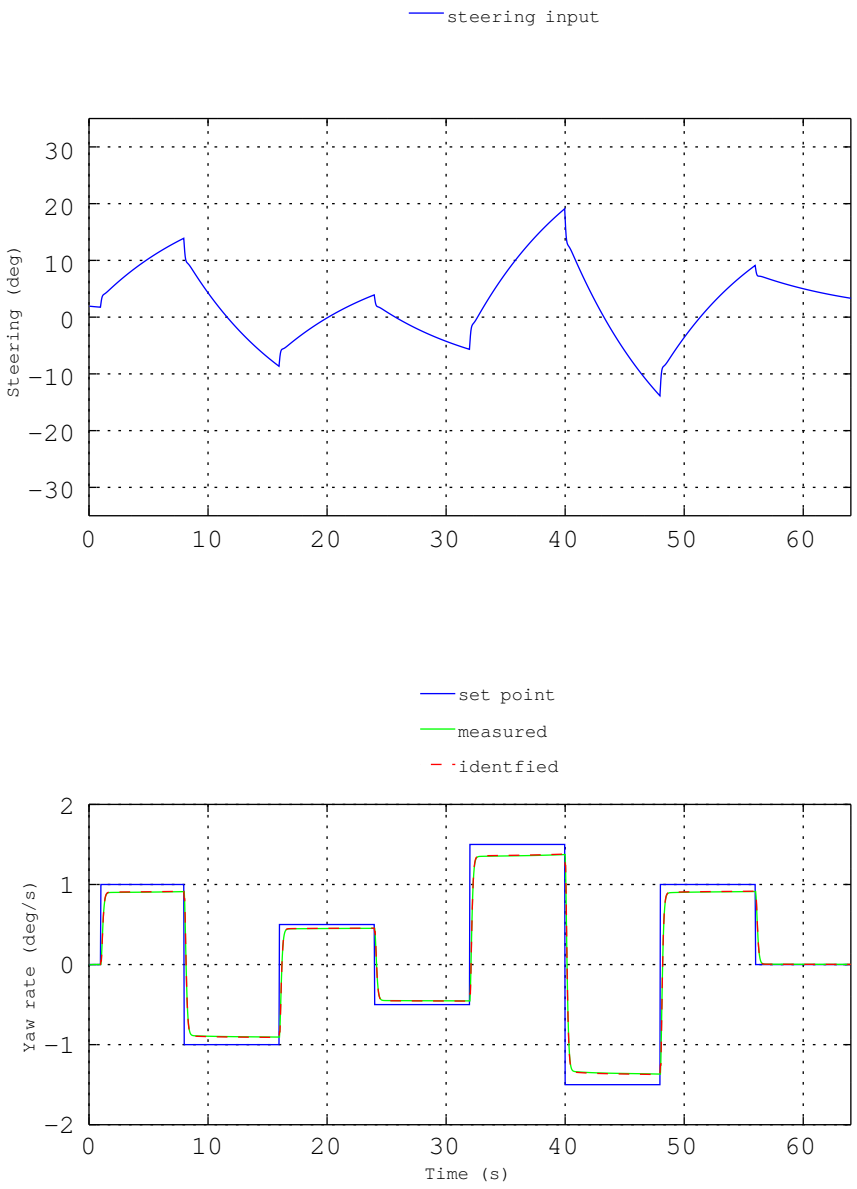


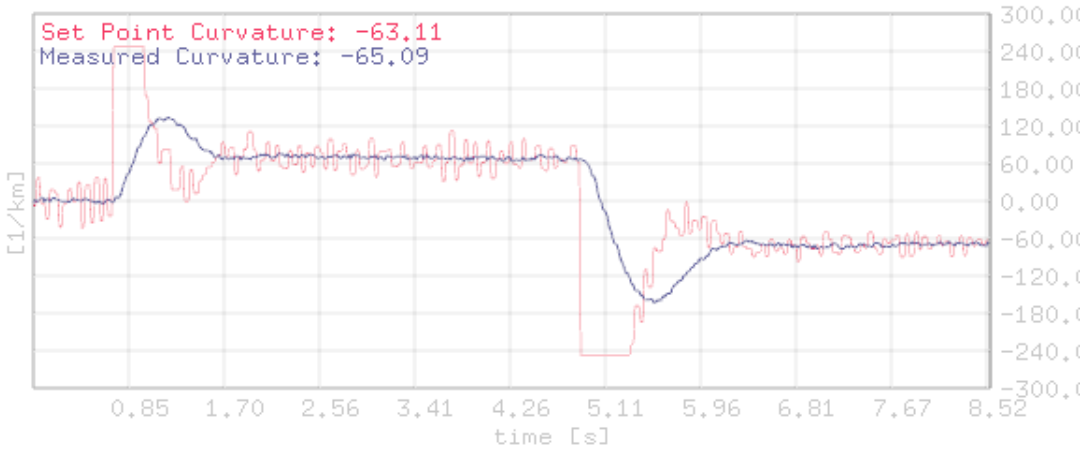
Figure 4.9 Yaw rate response for V_x =10 m/s and $C_{\alpha h}$ =3000 N/deg. (a) Steering input (deg). (b) Yaw rate (deg/s).

It also contains measurement noise and control disturbances. In this case, the control signal (curvature in km^{-1}) of the self-tuning regulator is applied to the steering system and the measured curvature as well as the measured yaw rate are the outputs. We can appreciate that the control signal saturates at around 120 km^{-1} since the real steering system is not able to turn more than 28^o. From the aspect of the pole-placement analysis presented in Section 4.2.1, we expect to have more oscillations as the veloc-

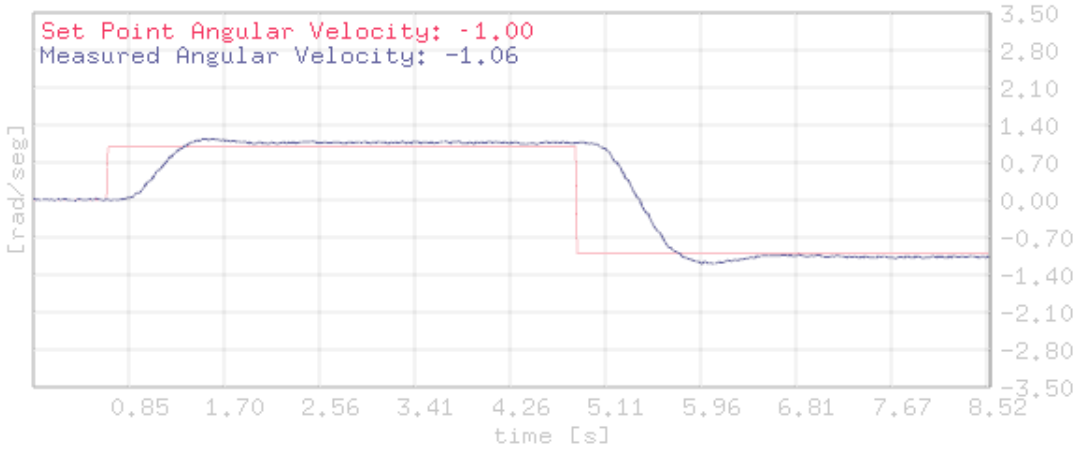
ity and cornering stiffness increase (Figs. 4.5 and 4.6). This effect is amplified due to the increment of order by the steering system and since the designed self-tuning regulator was based on a 2nd order reference model (Eq. (4.31)), the order of the identified and reference models has to be increased as well to mitigate the oscillations seen in Figures 4.11 and 4.12. Nevertheless, the identified 2nd order system seems to be enough for the adaptation to be able to control the vehicle. Compared to Figures 4.8 and 4.9, the tendency of increasing the control signal as the velocity and hitch cornering stiffness increases, is also to be found here. Finally, we can say that in general the results are satisfactory and the algorithm is able to adapt the control signal to the changes of velocity and hitch cornering stiffness.

4.3 Robust self-tuning regulator

This section presets the experiment and results obtained by applying the second controller proposed in this research. The first improvement expected here is the use of only one controller for the lateral position. Also, the introduction of tracking and regulation dynamics, including a robust design. In contrast to the last experiment in the previous section using a tractor model, this method is implemented on an agricultural robot. This fact indirectly help us to assess the capabilities of this method to be used for different vehicles. The calculations done for the controller design presented in Section 3.2 are performed online inside an ECU of the real robot presented in Section 4.1.2, but for demonstration purposes the lateral speed is neglected and a fixed forward speed can be used ($V_x = V = 0.5m/s$). The sampling time of the digital system is 100 ms. The track-width of 0.455 m corresponds also to the real robot described in Section 4.1.2, with a time constant of 100 ms (see Tab. 4.6).

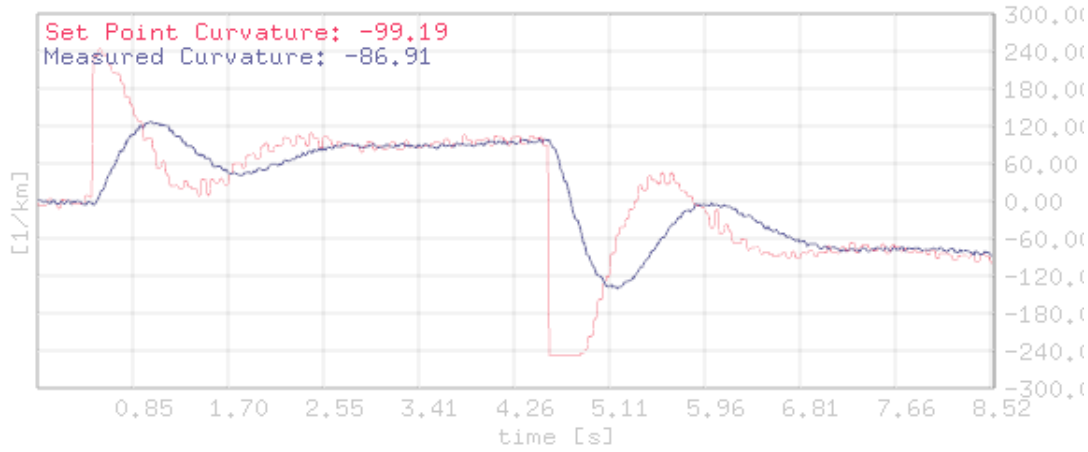


(a)

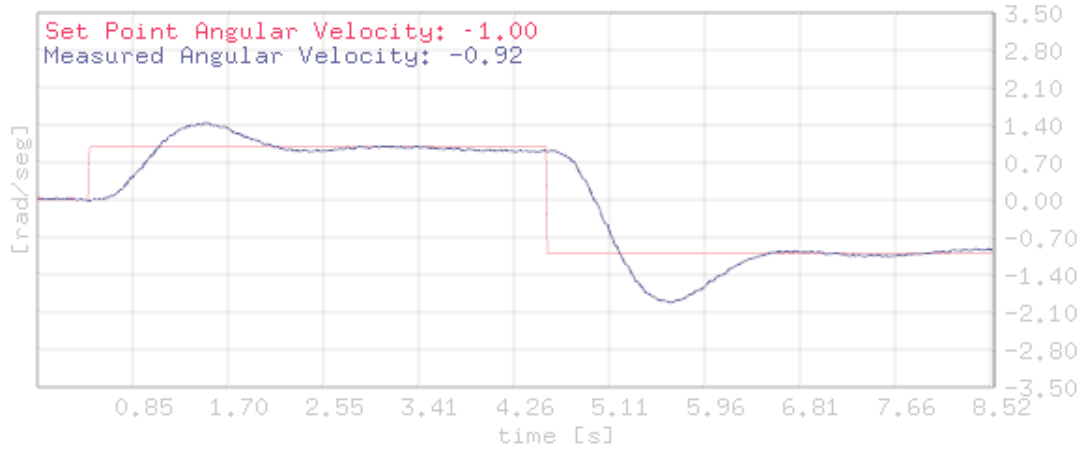


(b)

Figure 4.10 Yaw rate control. Validation with Openframeworks for V_x =2 m/s and $C_{\alpha h}$ =300 N/deg. (a) Steering input (1/km). (b) Yaw rate (rad/s).

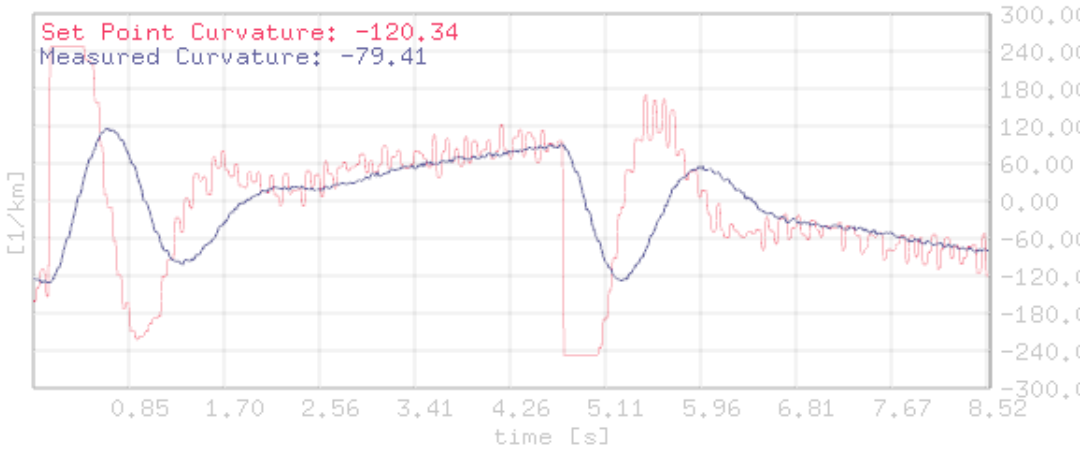


(a)

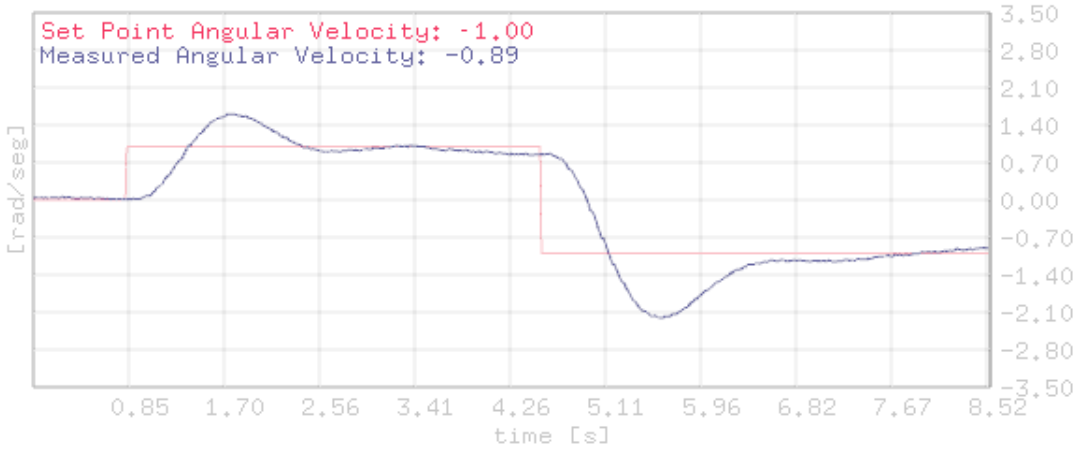


(b)

Figure 4.11 Yaw rate control. Validation with Openframeworks for V_x =5 m/s and $C_{\alpha h}$ =1000 N/deg. (a) Steering input (1/km). (b) Yaw rate (rad/s).



(a)



(b)

Figure 4.12 Yaw rate control. Validation with Openframeworks for V_x =10 m/s and $C_{\alpha h}$ =3000 N/deg. (a) Steering input (1/km). (b) Yaw rate (rad/s).

Table 4.6 Technical data of a skid-steered robot.

Robot data		
T_s	0.100	ms
τ	0.100	
$2c$	0.455	m
V_x	0.500	m/s

Substituting the data from table 4.6 into Eqs. (4.22), (4.26) and (4.28) we get the following plant model parameters:

$$\begin{aligned} d &= 0 \\ B(z^{-1}) &= 0.003z^{-2} + 0.003z^{-3} \\ A(z^{-1}) &= 1 - 2.368z^{-1} + 1.736z^{-2} - 0.367z^{-3}. \end{aligned} \tag{4.32}$$

To shape the sensitivity functions we choose the fixed parts represented in Eq. (4.33) and the new poles and zeros to be used in the right part of the Bezout Eq. (3.37) are expressed in Eq. (4.34).

$$\begin{aligned} H_R(z^{-1}) &= 1 + 0.5z^{-1} \\ H_S(z^{-1}) &= 2 - 1.0z^{-1}. \end{aligned} \tag{4.33}$$

$$\begin{aligned} B'(z^{-1}) &= 0.003473z^{-2} + 0.0069461z^{-3} + 0.003473z^{-4} \\ A'(z^{-1}) &= 1 - 2.868z^{-1} + 2.92z^{-2} - 1.236z^{-3} + 0.1839z^{-4}. \end{aligned} \tag{4.34}$$

For the regulation dynamics we use $T_s = 0.1$, $\omega_r = 1.5 + 0.6 \cdot V_x$, $\zeta_r = 1.0$ and $\alpha_1 = \alpha_2 = -0.5$ and so with a forward speed of 0.5 m/s, the left part of the Bezout Eq. (3.37) appears as follows:

$$P(z^{-1}) = P_D \cdot P_F = (1 - 1.846z^{-1} + 0.8521z^{-2}) \cdot (1 - 0.5z^{-1})^2. \tag{4.35}$$

For the tracking dynamics we use $T_s = 0.1$; $\omega_t = 2.5 + 0.5 \cdot V_x$; and $\zeta_t = 1.0$ to find the following parameters of the reference model H_m:

$$\begin{aligned} Bm(z^{-1}) &= 0.01752z^{-1} + 0.01534z^{-2} \\ Am(z^{-1}) &= 1 - 1.637z^{-1} + 0.6703z^{-2}. \end{aligned} \tag{4.36}$$

Finally, using these results to solve Eqs. (3.29) and (3.33), we find the following control parameters R, S and T:

$$\begin{aligned} R(z^{-1}) &= 1 - 0.4784z^{-1} + 0.04941z^{-2} - 0.005427z^{-3} - 0.01235z^{-4} \\ S(z^{-1}) &= 8.788 - 6.796z^{-1} - 7.374z^{-2} + 6.903z^{-3} - 1.308z^{-4} \\ T(z^{-1}) &= 144 - 409.7z^{-1} + 424.4z^{-2} + 189.1z^{-3} - 30.67z^{-4}. \end{aligned} \tag{4.37}$$

4.3.1 Simulation

With these results we can prove the robustness of the design for the different sensitivity functions. Figure 4.13 depicts the Nyquist plot of the open loop function $H_{ol} = S_{yp}^{-1} - 1$ in green. Since the critical point $[-1, j0]$ is at the left of the plot, we conclude that the open loop system has no unstable poles. Here, the inverse of the gain margin $1/\Delta G$ and the phase margin $\Delta\Phi$ are stability margins with respect to the critical point. It can be seen that the modulus margin ΔM is the radius of a circle centered at the critical point and tangent to the Nyquist plot. It tells us the distance between H_{ol} and the critical point. Therefore, a good modulus margin (e.g. $\Delta M \geq 0.5$) implies stable gain and phase margins.

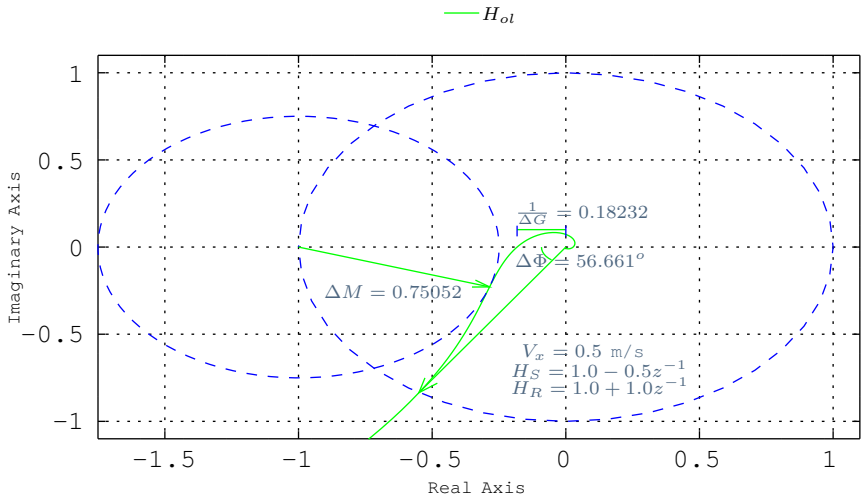


Figure 4.13 *Nyquist plot.*

Figure 4.14 shows in blue the applied modulus and delay margins to the *output sensitivity function* S_{yp}. The upper-left bound is built by a modulus margin $\Delta M = 0.5(-6dB)$ (Eq. (3.41)) and the upper-right and lower bounds represent the limits of the delay

margin (Eq. (3.43)). The *output sensitivity function* in green is the result of imposing the fixed part $H_S(z^{-1})$ to the pole placement design in an iterative way until a satisfactory result was found (Eq. (4.33)) (Landau and Zito, 2006). It can be seen that using different values for the fixed part H_S will shape the *output sensitivity function* differently. It can also be seen that, although using $H_S = 1 - 1z^{-1}$ (dashed green line) will reject constant disturbances in steady state, the shaped S_{yp} is closer to the upper margin. This will make the system less tolerant to changes on the system dynamics at lower frequencies and to fix that we should impose slower regulation dynamics (e.g. $\omega_0 < 0.8$). On the other hand, using $H_S = 1 - 0.5z^{-1}$ (solid green line) will decrease the gain, making the system more tolerant to uncertainties with faster regulation dynamics.

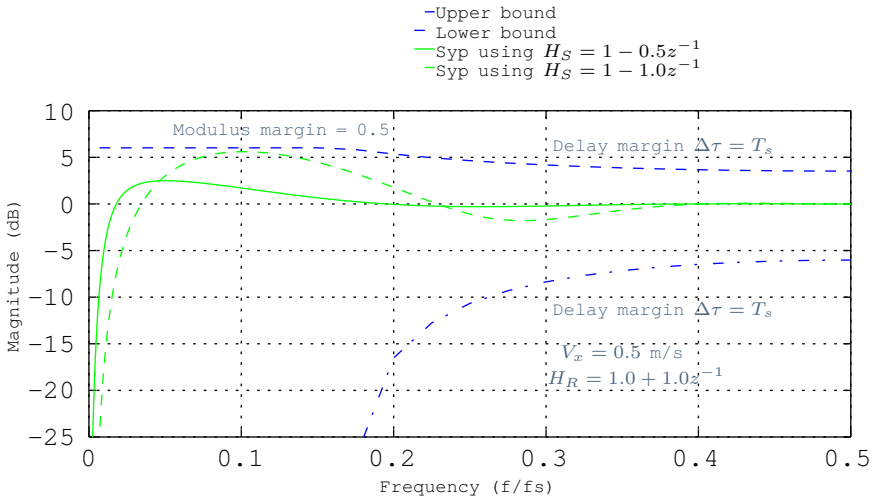


Figure 4.14 *Output sensitivity function.*

Figure 4.15 shows in green the *noise sensitivity function* S_{yb} and in blue the bound related to the delay margin $\Delta\tau = T_s$ (Eq. (3.40)). The delay margin was already set for the margins of the *output sensitivity function* (S_{yp}) with good results (Fig. 4.14). Nevertheless, the delay margin is related to the *noise sensitivity function* and the fact that S_{yb} is under the bound confirms the good choice of the fixed parts (Eq. (4.33)).

Figure 4.16 shows in green the *input sensitivity function* (S_{up}) for a pole placement with the imposed fixed part $H_R = 1 + 1z^{-1}$

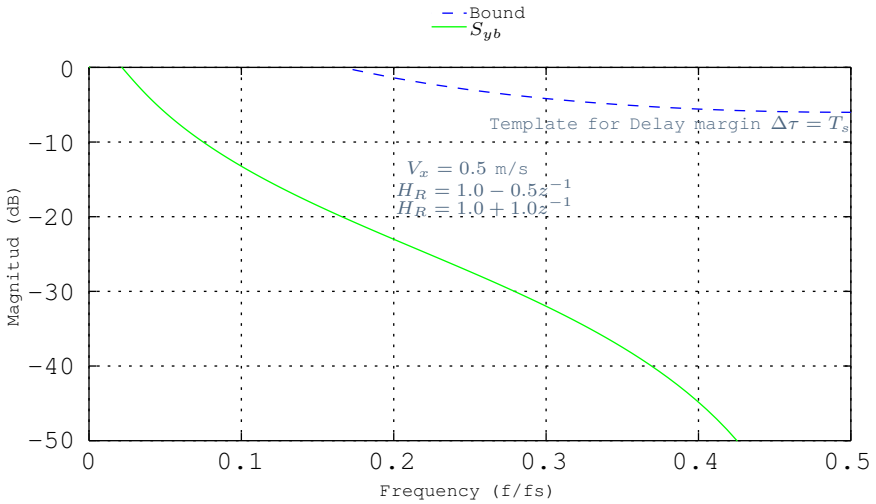


Figure 4.15 *Noise sensitivity function.*

(Eq. (4.33)) which will impose a zero at $0.5f_s$ reducing the actuator stress at higher frequencies. This has the effect of limiting not only the actuator stress but also the reaction to measurement noise and disturbances making the robot drive more smoothly and not react right away after every stone or irregularity in the soil. It will also reduce the energy consumed by the motors and increase their lifetime.

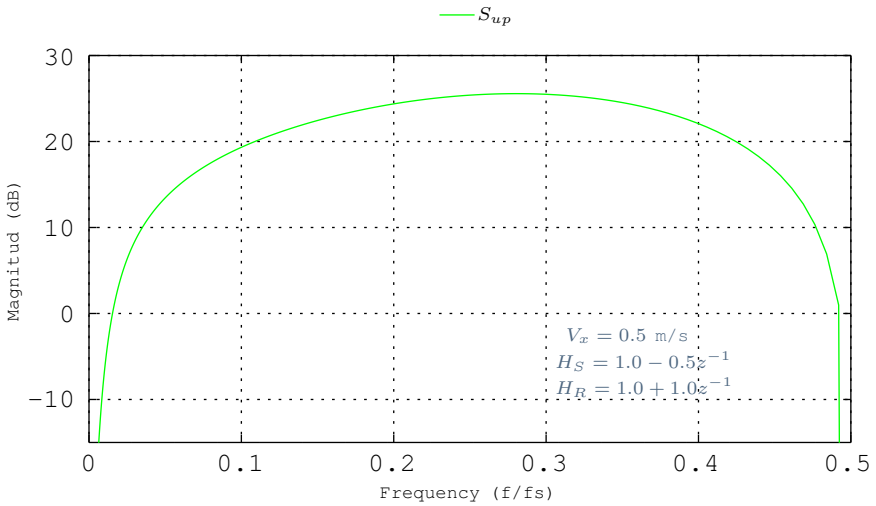


Figure 4.16 *Input sensitivity function.*

The resulting step response of the different shaping of the *output sensitivity functions* is illustrated in Figure 4.17. The dashed green line corresponds to the response using the fixed part $H_S = 1 - 1.0z^{-1}$, whereas the solid green one corresponds to the use of $H_S = 1 - 0.5z^{-1}$. It can be seen that the solid green

line arrives faster at the steady state (around 4 s) whereas the dashed one needs more than 8 s and contains oscillations. As mentioned before, the oscillations can be fixed by picking slower regulation dynamics ($\omega_0 < 0.8$) or by driving faster. Nevertheless, we could reduce the error at the turning points by reducing the speed and to this end, it is desirable to have the system react faster to changes.

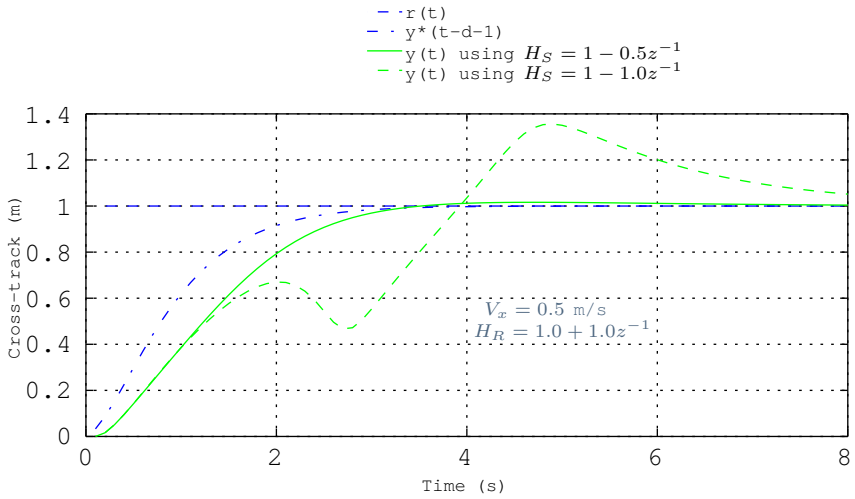


Figure 4.17 Step response over 1 m cross-track.

A 3D real-time simulation using the results of Eqs. (4.32), (4.36) and (4.37) is shown in Figure 4.18. It depicts representative results which are also useful to prove the robustness of the behaviour in an interactive environment (Fernandez et al., 2016). For the simulation, a real path with geographic coordinates was used so we could compare these results with the results of a real robot using the same path. The simulation presents very satisfactory results where the largest error (at the fifth turning point) is around 20 cm which is comparable to other more complex solutions (Arslan and Temeltas, 2011; Caracciolo et al., 1999; Hwang et al., 2013; Inoue et al., 2013; Jeon and Jeong, 2015; Tchoń et al., 2015). These errors at the turning points are expected since in order to be able to turn with zero curvature radius, the skid-steered robot should stop completely. In contrast, in the simulation the robot was reducing its velocity to 0.1 m/s to test its robustness and adaptation.

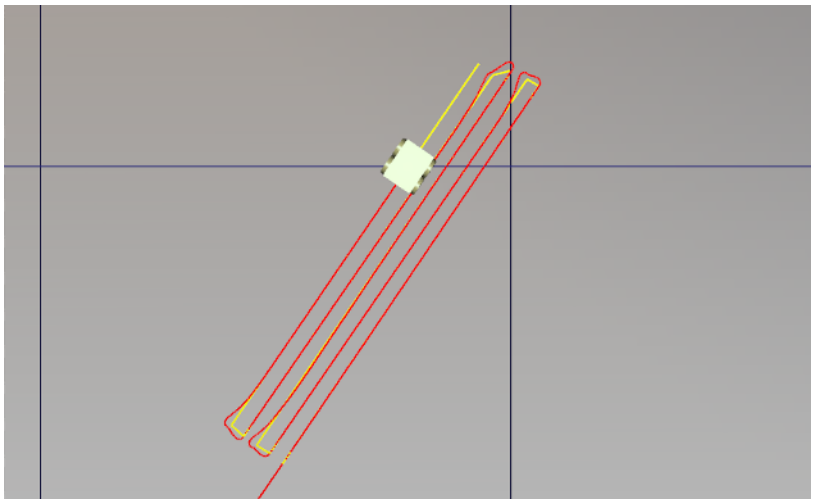


Figure 4.18 3D real-time simulation with Openframeworks and OpenGL. The yellow line represents the desired navigation path and the red line is the actual driven path.

4.3.2 Validation

The same coordinates used for the simulation (Fig. 4.18) were sent to the real robot and the results are shown in Figure 4.19. It can be seen that the robot was successfully following the path with a similar type of error in the turning points as presented in the simulation with a maximum of 18 cm also at the fifth turning point (see Fig. 4.20). In contrast to the simulation, disturbances due to terrain irregularities can be appreciated in the real approach. Nevertheless, the controller was able to react very smooth to disturbances without producing an error of more than 2 cm once on the lane. The same path was tested in different terrains and the skid-steered robot produced similar results.

The Figure 4.20 shows at the top the error of the lateral position or cross-track of the vehicle and at the bottom the forward speed. The vehicle was driving with a minimum speed of 0.1 m/s at the turning points and a maximum speed of 0.5 m/s once on the lane. It can be seen how the vehicle slows down near to each turning point and speeds up right after the turning point. It can also be seen that at the headland, the vehicle does not have enough lane to speed up to the maximum speed when the next turning point approaches making the vehicle slow down again. The test was performed on a gravel terrain where the wheels were able

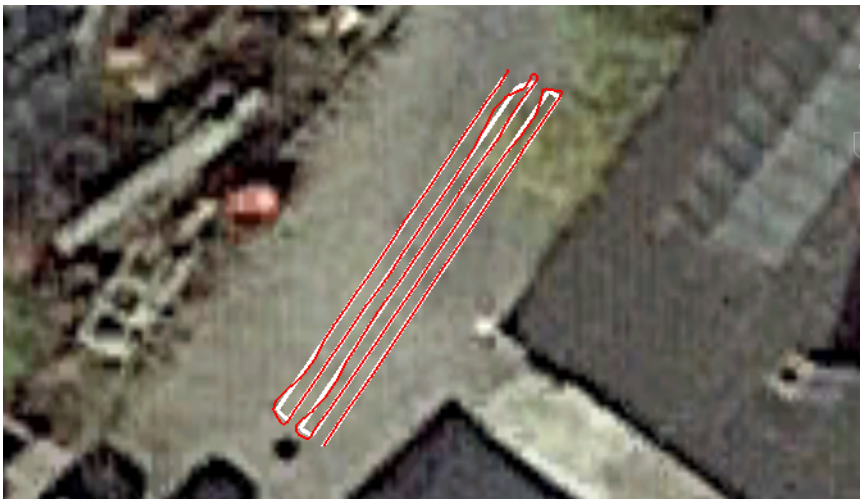
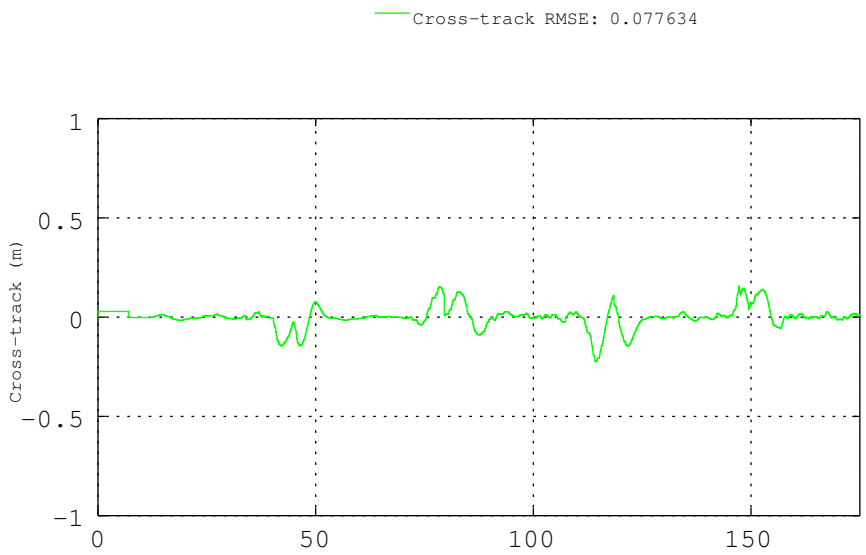


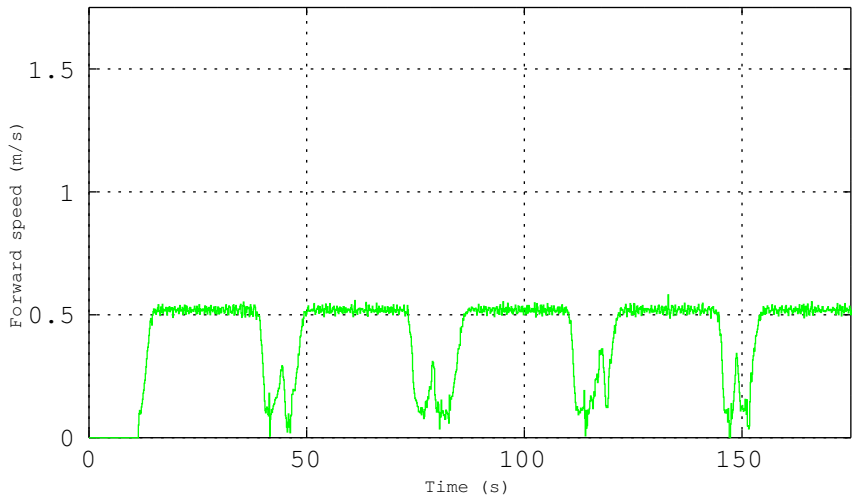
Figure 4.19 Real skid-steered robot. Lane-tracking control on the field. Visualisation with Google-Earth. The white line represents the desired path sent to the robot and the red line is the actual path driven by the robot.

to easily slip when turning. The slow speed allows for a high driving accuracy with a low cross-track root-mean-square error (RMSE) of 0.07. Furthermore, not estimation was used, instead, a reduce model (Eq. (4.32)) was used for the online calculation of the controller parameters R, S and T.

In Figures 4.21 and 4.22, the test was performed in a more difficult terrain, ploughed, with a lot of stones and with some parts dry and others very wet. Furthermore, the vehicle was driving faster with a minimum speed of 0.2 m/s at the turning points and a maximum speed of 1.5 m/s on the lane. It can be seen in Figure 4.21 how the vehicle has more difficulties at staying on the lane compared to the previous terrain (Fig. 4.19). The increase in the error at the turning points is due to the fact that the minimum speed is 0.2 m/s instead of 0.1 m/s. This was desired since the goal is to drive as fast as possible with a minimum acceptable error. The irregularities at the curves are due to the high irregularities in the terrain. Nevertheless, the results are satisfactory since the vehicle managed to follow the lane with a maximum error of 10 cm. Including the headland-error, the rsme equals 0.17, which is acceptable since the error at the curves increased due to the speed increment.



(a)



(b)

Figure 4.20 (a) Lateral position error throughout the driven path. (b) Change of forward speed throughout the driven path. Minimum V_x =0.1 m/s and maximum V_x =0.5 m/s.

In comparison with Figure 4.20, the lower part of Figure 4.22 presents shorter times when driving on the lane due to the speed increment of 1 m/s on the lane. The times when taking the curve at the headland are almost three times bigger that the time on the lane. Nevertheless, with the new speeds the vehicle was able to achieve one curve and one lane more within the same time.

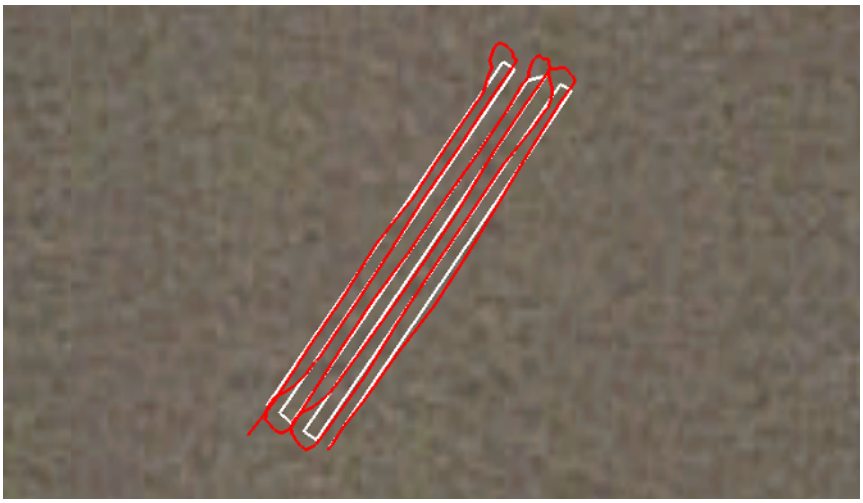
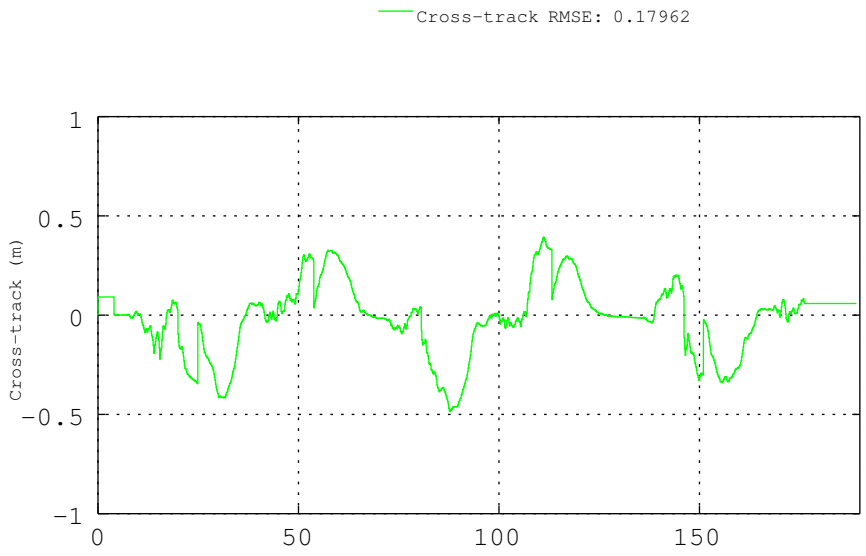


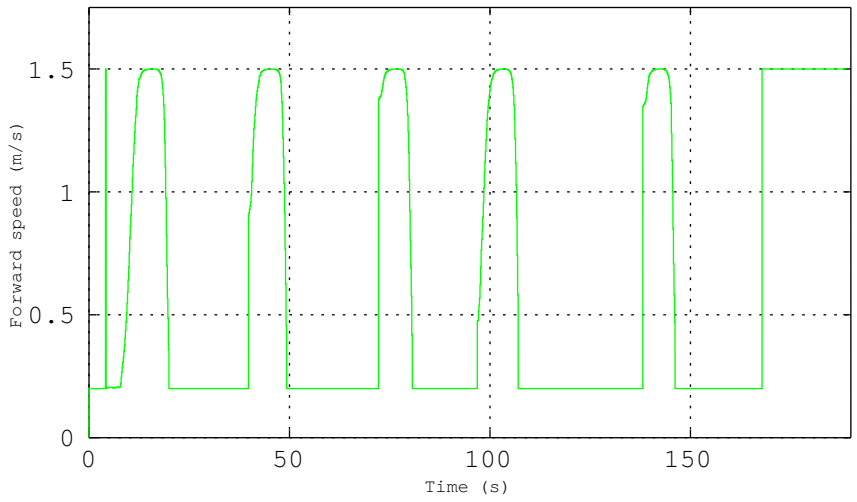
Figure 4.21 Real skid-steered robot. Lane-tracking control on a very difficult field. Visualisation with Google-Earth.

The Figure 4.23 shows the control signal in km^{-1}, which is the inverse of the curvature radius. High frequency elements can be appreciated since the closed loop system is of 4th order. The measurements of the yaw rate were done using the raw data from the IMU and the control signal could be smoothed if a filter is applied to the IMU data. Nevertheless, some accuracy could be lost and we opted to use the raw data also to test the reaction of controller against measurement noise.

As aforementioned, the results and figures shown above, represent the system without using the online estimator. In contrast, some results using the online gradient estimator are illustrated in Figure 4.24. Here, the vehicle was driving with a minimum speed of 0.2 m/s at the curves and a maximum speed of 1.5 m/s once on the lane. The lanes distribution are the same used above with a length of 10 m per lane and a separation of around 0.33 m. The results appear to be very positive, having errors similar to those when driving slower. The following Figure 4.25 shows at the top a cross-track comparison of both methods (with online gradient estimator in blue and with the static model in green). In the lower plot of the figure, the distance to the turning point is depicted. The results using the estimator appear to fit better the system dynamics. Also, the RMSE is 1/3 smaller than the error



(a)



(b)

Figure 4.22 (a) Lateral position error throughout the driven path. (b) Change of forward speed throughout the driven path. Minimum V_x =0.2 m/s and maximum V_x =1.5 m/s.

using the static model of Eq. (4.27). Since the static model was an approximation, the identified estimator finds a model with shorter time delay and by doing so, the *Controller Design* calculates the parameters for a slightly faster system. Also, since this identified system is faster and with less error, the 5 lanes are completed almost 30 s faster.

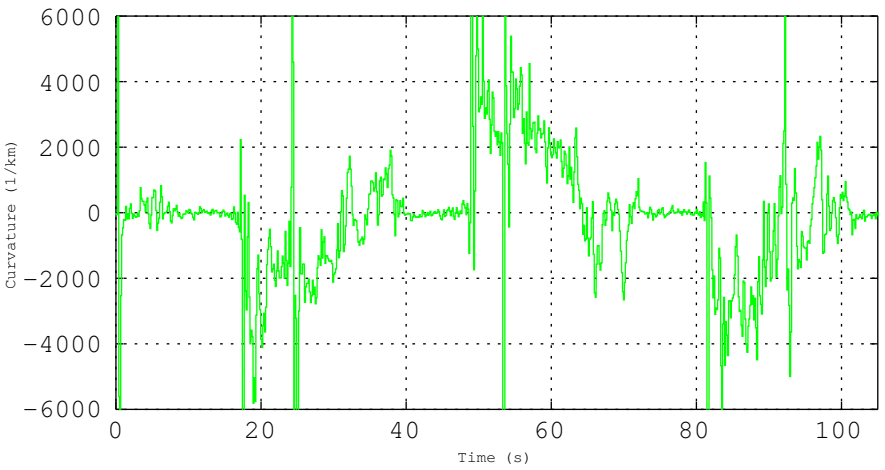


Figure 4.23 Curvature applied to the motors in km^{-1} which is the inverse of the turning radius.

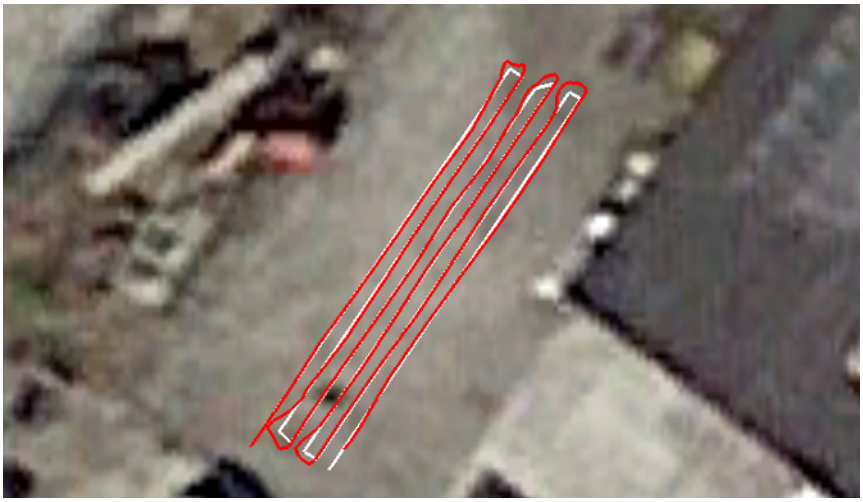
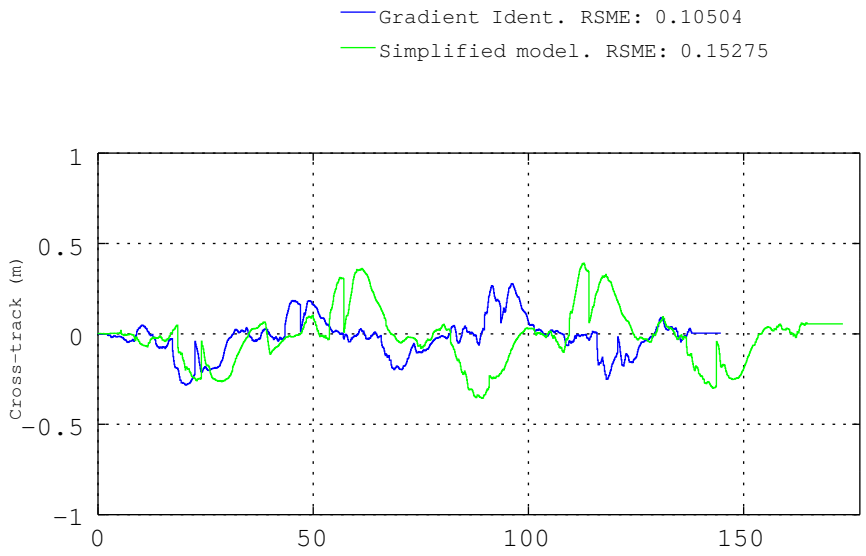


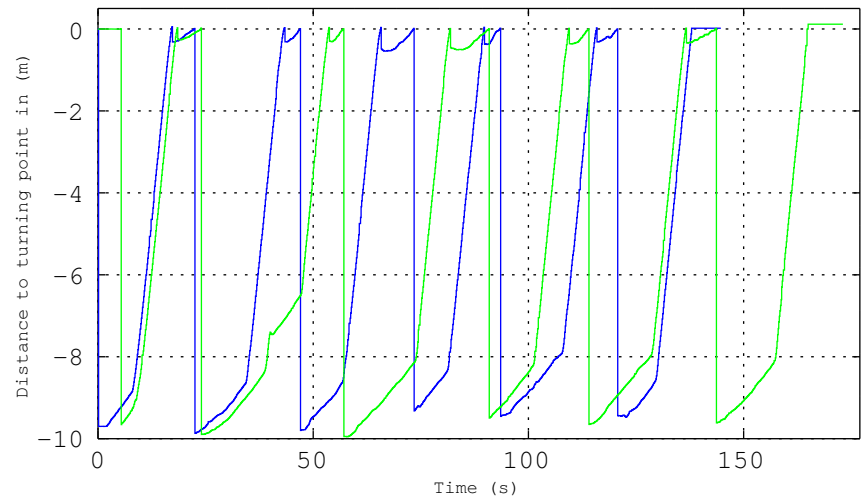
Figure 4.24 Real skid-steered robot. Lane-tracking control using an online gradient algorithm for the model estimation. Visualisation with Google-Earth.

4.4 Optimal digital control

This section presents the results of applying an optimal controller to the robot described in Section 4.1.2. These results can be used as a benchmark to be compared with those of the robust self-tuning design obtained from the same vehicle in the previous Section 4.3. Data from Table 4.7 provide sufficient information to design and implement the controller introduced in Section 3.3, into the skid-steered robot (Sec. 4.1.2). The parameters Q and R can be used as tuning parameters to cover for other factors not



(a)



(b)

Figure 4.25 Comparison using a defined static model vs. using a gradient model estimation. In both approaches the minimum V_x =0.2 m/s and the maximum V_x =1.5 m/s. (a) Lateral position error throughout the driven path. (b) Distance to next turning point.

included such as weight and other uncertainties due to the use of a simplified first order system.

Table 4.7 Technical data of a skid-steered robot.

Robot data		
T_s	0.100	ms
τ	0.100	
$2c$	0.455	m
V_x	0.100-1.500	m/s
Q	0.100	
R	0.100	

Using $V_x = V = 0.5$ m/s and substituting the rest of the data from Table 4.7 into Eqs. (4.25), (4.26), (4.27) and (4.28) we obtain the representation of the vehicle (Eq. 4.30), expressed in the equation below (4.38) which is used in the observer as well (see Fig. 3.9). For the range of working speeds, these calculations take place in real-time in the ECU using the measured forward speed.

$$\begin{aligned} \Phi &= \begin{bmatrix} 0.000 & 1.000 & 0.000 \\ 0.000 & 0.000 & 1.000 \\ 0.367 & -1.735 & 2.367 \end{bmatrix} \\ \Gamma &= \begin{bmatrix} 0.000 \\ 0.000 \\ 1.000 \end{bmatrix} \\ C &= \begin{bmatrix} 0.003 & 0.003 & 0.000 \end{bmatrix}. \end{aligned} \tag{4.38}$$

The controllability and the observability matrices have full rank (Eqs. (4.39) and (4.40) respectively). Therefore, our system is controllable and observable and we can proceed to find the optimal solution for the state feedback and state estimate.

$$rank \begin{pmatrix} 0.000 & 0.000 & 1.000 \\ 0.000 & 1.000 & 2.367 \\ 1.000 & 2.367 & 3.871 \end{pmatrix} = 3, \tag{4.39}$$

$$rank \begin{pmatrix} 0.003 & 0.003 & 0.000 \\ 0.000 & 0.003 & 0.003 \\ 0.001 & -0.006 & 0.011 \end{pmatrix} = 3. \tag{4.40}$$

It was shown in Section 3.3.1 that the state feedback F and the state estimate L can be found with the Riccati Eqs. (3.49) and (3.52)as follows:

$$\begin{aligned} P_f &= Q + \Phi^T \cdot P_f \cdot \Phi \\ &- \Phi^T \cdot P_f \cdot \Gamma \cdot (R + \Gamma^T \cdot P_f \cdot \Gamma)^{-1} \cdot \Gamma^T \cdot P_f \cdot \Phi; \end{aligned} \tag{3.49}$$

$$\begin{aligned} P_l &= Q_e + \Phi \cdot P_l \cdot \Phi^T \\ &- \Phi \cdot P_l \cdot C^T \cdot (R_e + C \cdot P_l \cdot C^T)^{-1} \cdot C \cdot P_l \cdot \Phi^T. \end{aligned} \tag{3.52}$$

Nevertheless, as it can be seen, the solution to the differential nonlinear Riccati equation depends on itself. There are different methods to solve the equation and one would be in an iterative manner. Therefore, for a digital system Eqs. (3.49) and (3.52) can be rewritten as following:

$$\begin{aligned} P_f(k) &= Q + \Phi' \cdot P_f(k-1) \cdot \Phi \\ &\quad -\Phi' \cdot P_f(k-1) \cdot \Gamma \\ &\quad \cdot (R + \Gamma' \cdot P_f(k-1) \cdot \Gamma)^{-1} \cdot \Gamma' \cdot P_f(k-1) \cdot \Phi, \end{aligned} \tag{4.41}$$

$$\begin{aligned} P_l(k) &= Q_e + \Phi \cdot P_l(k-1) \cdot \Phi' \\ &\quad -\Phi \cdot P_l(k-1) \cdot C' \\ &\quad \cdot (R_e + C \cdot P_l(k-1) \cdot C')^{-1} \cdot C \cdot P_l(k-1) \cdot \Phi', \end{aligned} \tag{4.42}$$

where the solution can be found by iterating at each cycle time until P_f and P_l converge (Ogata, 1995). For our problem, the solution converges after around 100 cycles with the initial values of $P_f = P_l = 0$ (Eqs. (4.43) and (4.44)). With a cycle time of 100 ms, these solutions will be found after 10 s which might be too slow for our vehicle. For example driving at 1.5 m/s, the vehicle will not be able to follow the path at least for the first 10 m. To solve that, the solution can be calculated first in a simulation. Then, this solution can be used as the initial conditions inside

the vehicle and since there might be discrepancies between the simulation and the real vehicle, some calculations still have to be done. Nevertheless, it will converge only after a couple of cycle times. This will be the case as well for the different range of working speeds, since for each speed a different solution has to be calculated in real-time. Also in this case, the discrepancies of the solutions between different speeds are small enough to be calculated in real-time.

$$P_f = \begin{bmatrix} 0.003 & -0.011 & 0.009 \\ -0.011 & 0.046 & -0.037 \\ 0.009 & -0.037 & 0.029 \end{bmatrix}. \tag{4.43}$$

$$P_l = \begin{bmatrix} 544.010 & 615.560 & 687.410 \\ 615.560 & 706.210 & 797.820 \\ 687.410 & 797.820 & 911.010 \end{bmatrix}. \tag{4.44}$$

From the solution to the Riccati equation, the optimal state feedback and state estimate as well as the tracking gain are expressed in Eq. (4.45) and the controller is ready to be implemented. Figure 4.26 illustrates a step response of 1 m using these results (Eqs. (4.38) and (4.45)) to validate the controller.

$$\begin{aligned} F &= \begin{bmatrix} -0.084 & 0.326 & -0.260 \end{bmatrix} \\ L &= \begin{bmatrix} -35.332 \\ -39.701 \\ -44.083 \end{bmatrix} \\ K &= 2.774. \end{aligned} \tag{4.45}$$

4.4.1 Simulation

A 3D real-time simulation was programmed with C++ using Openframeworks and some representative results can be seen in Figures 4.27 and 4.28 (Fernandez et al., 2016). The yellow line represents the way-path to be followed by the robot and contains real geographic coordinates. The red line is the actual driven path. It can be

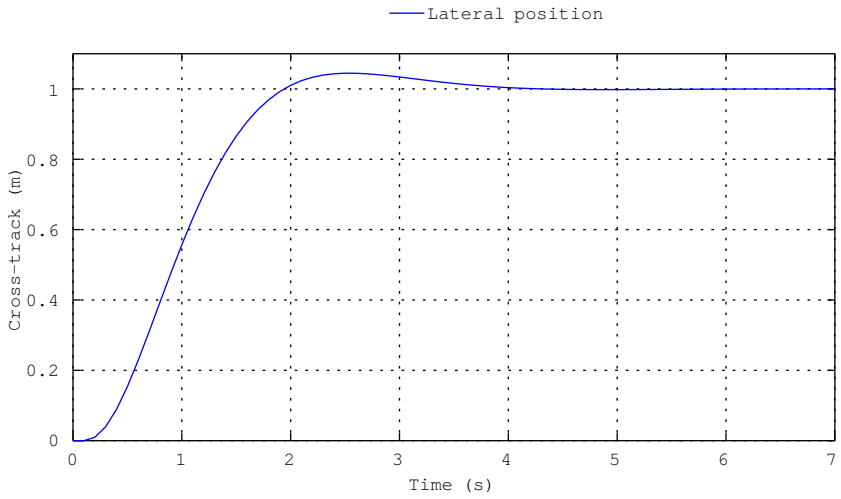


Figure 4.26 Step input of 1 m to the left of the desired path.

seen in Figure 4.28 that the error on the lateral position increases in the turning points since those correspond to an input step applied by changing the course angle to the desired path (ν in Figure 4.2).

For the simulation, the forward speed of the robot was changing from 1.5 m/s on the lane to 0.1 m/s at the headland. After each turn, we can appreciate some small oscillations before the robot is completely on the lane and this is due to two factors. The first one is because Q and R were tuned to provide the quickest response. Bigger values will avoid the oscillations, but the robot will take longer to arrive at the lane and the turning curve will be larger. The second one is because of the change in the forward speed, which requires new real-time calculations of the control parameters. When slowing down from 1.5 to 0.5 m/s before the turning points, the adaptation cannot be appreciated, which confirms that the solution converges fast enough. When speeding up again to 1.5 m/s after the headland, some small effect is appreciated since at higher speeds the vehicle needs higher response dynamics that have to be calculated, and at those speeds, if the vehicle needs 5 cycles (500 ms) to calculate the control parameters it will take around 75 cm.

From Figure 4.27 one could expect that the smoothness of the robot decreases when following smooth curves, since it will have

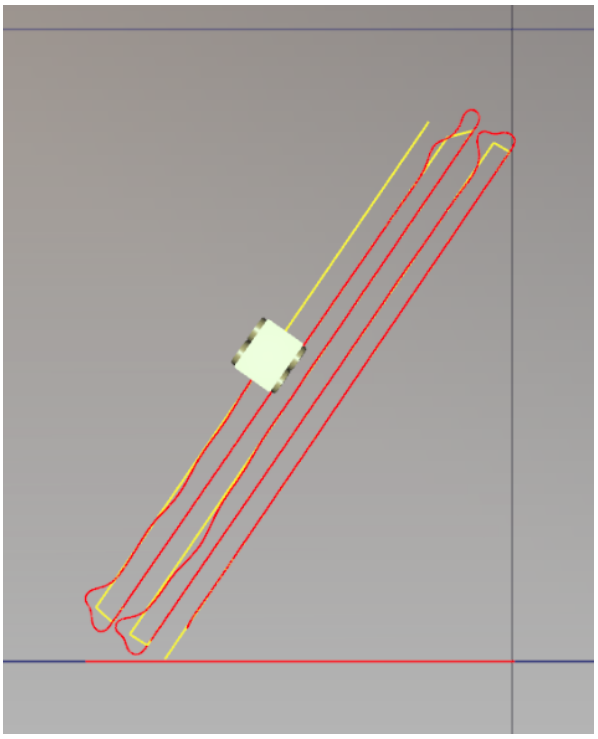


Figure 4.27 3D real-time simulation with Openframeworks and OpenGL (path following).

a continues input of steps by updating small straight "desired paths" with their corresponding course angles ν. Nevertheless, Figure 4.28 illustrates how the robot smoothly follows a lane change of 2 m. At the bottom of the way-line, the step input is very clear with a change in the course angle of 90^o. Then at the lane change, in every cycle time the course angle is updated to the corresponding change and the steps do not affect the driving smoothness, having an error of around 7 cm at the curve. Similar errors were found in the different simulations performed.

4.4.2 Validation

Figure 4.29 shows the results using the real robot. Here the white line is the same as the yellow one used in the simulation to be able to compare directly with the simulation results. Each lane is around 10 m long and the separation between lanes is 33 cm. As expected, it can be seen that the real robot presents a bigger error. The yellow line at the top presents a separation of around 90 cm at the time of changing the lane(see Fig. 4.30). On

Figure 4.28 3D real-time simulation with Openframeworks and OpenGL (lane change).

the other hand, the largest error when entering the lane is around 13 cm (yellow line at the third lane from right to left, almost at the bottom of Figure 4.29). The results are satisfactory if the cross-track at the headland is not considered very important by the user, since the error measured once on the lane is less than 5 cm and can be related to measurement noise and disturbances due to irregularities on the terrain. This error is also comparable to results obtained with more complex solutions (Pazderski and Kozłowski, 2008; Tchoń et al., 2015). The same test was performed in different terrains and the robot produced similar results.

In Figure 4.31, the signal produced by the controller is presented in the form of a curvature, which is the inverse of the turning radius. It can be recognized where the turning points took place through the peaks (e.g. at 20 and 40 s). One of the main advantages of using an optimal controller can be seen here, since the signal does not contain high frequency members and the changes are very smooth without stressing the motors. Since the

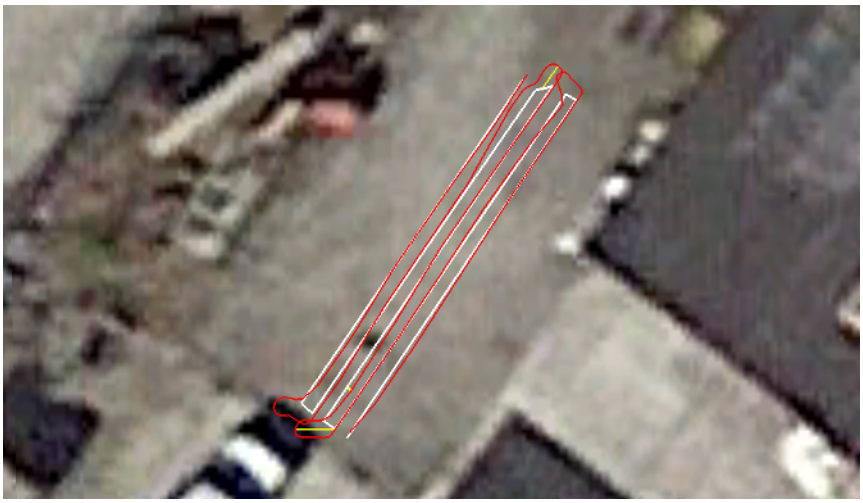


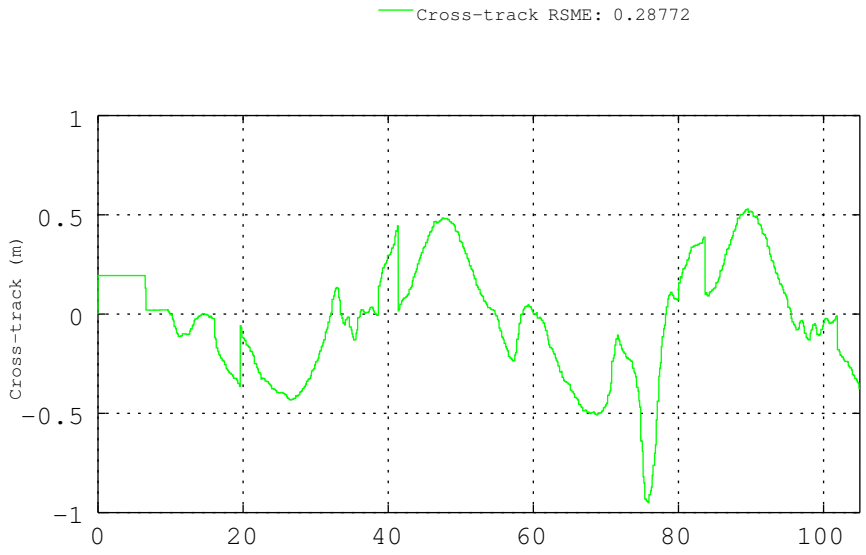
Figure 4.29 Real skid-steered robot. Lane-tracking control on the field. Visualisation with Google-Earth.

observer acts as a Kalman filter, the control signal does not react to unnecessary disturbances and measurement noise without loosing much responsiveness. On the other hand, it could take longer to arrive to the lane at the turning points. Nevertheless, the trade-off between actuator stress and system accuracy of an optimal controller can be appreciated here.

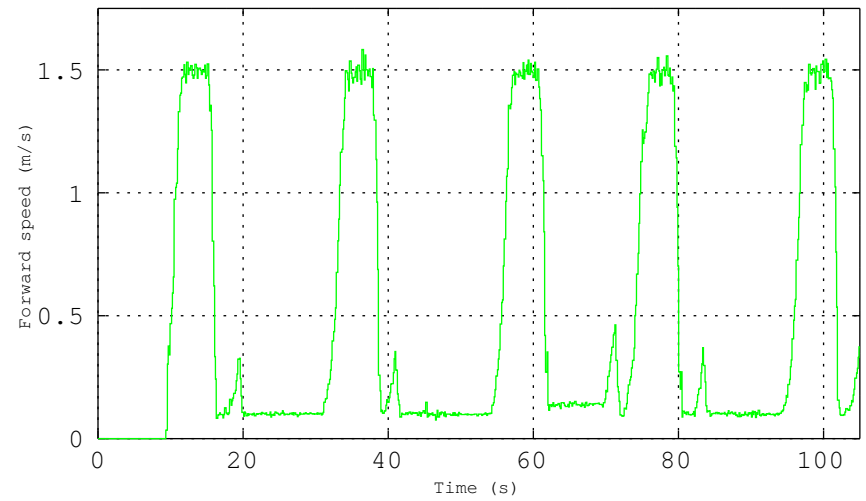
4.5 Summary of results

The first natural conclusion is that the robust self-tuning approach has more advantages than the cascade self-tuning one, beginning with the fact that it is not a nested set of controllers. There is no need for using PID and the only controller used, introduces a robust and more flexible design that can work with higher order models. This advantage can be used in situations like the one shown in Figures 4.10 to 4.12, where an overshoot is produced since the reference model used for the MDPP is of order two, and the real tractor-implement model would be of order 5 (including the hydraulic steering system).

For the sake of saving resources, the field tests where only performed with the skid-steered robot whit the robust self-tuning approach and with the observer-based optimal approach, where the



(a)



(b)

Figure 4.30 (a) Lateral position error throughout the driven path. (b) Change of forward speed throughout the driven path. Minimum V_x =0.2 m/s and maximum V_x =1.5 m/s.

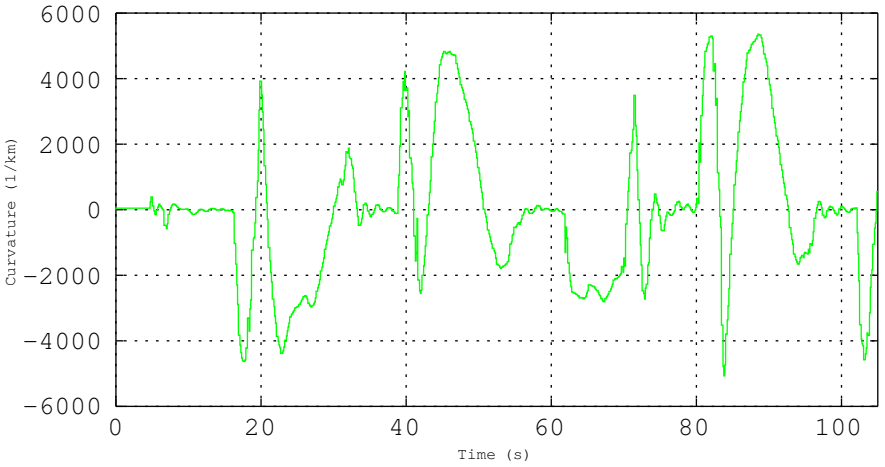
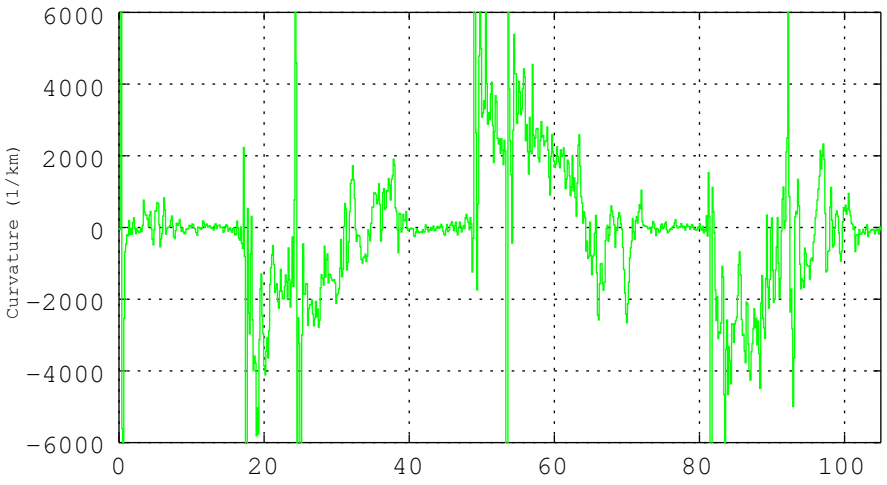


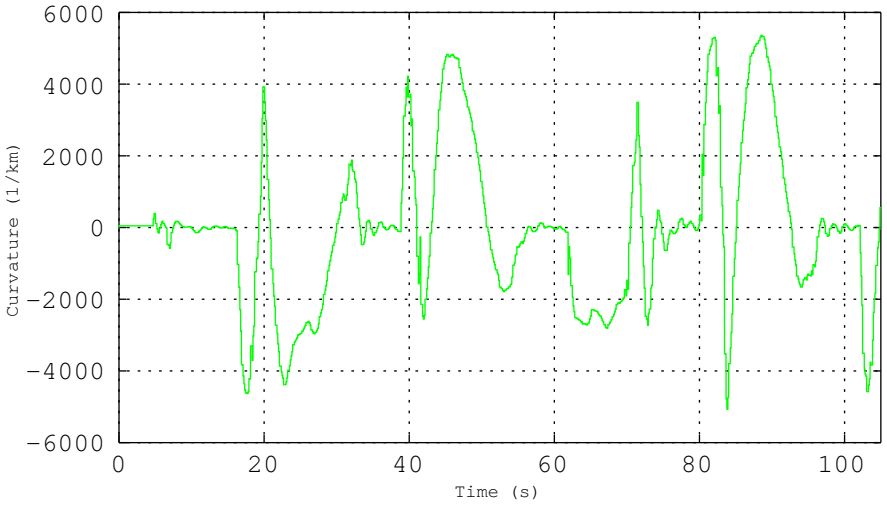
Figure 4.31 Curvature applied in km^{-1}.

last one is used as a benchmark. Figure 4.32 presents a comparison of the control signals of the two different controllers. At the top, the curvature produced by the robust self-tuning regulator (also illustrated in Fig. 4.23). At the bottom, the curvature produced by the optimal controller (also illustrated in Fig. 4.31).

Since the robust self-tuning regulator uses a controller of 5th order, the high frequency oscillations can be appreciated in contrast to the control signal produced by the optimal controller, which is of 3rd order. In this case, the optimal controller has the advantage of driving very smooth and treating the motors more efficiently. Also the reaction of the robot is more smooth and can be accounted as rejection against measurement noise and small disturbances, with the cost of reducing the accuracy of the path following. The difference in accuracy between both methods can be appreciated especially by comparing Figures 4.19 and 4.29 where it can be seen that the robust controller is more accurate and does not separate too much from the tracks at the headland. Also the RMSE of the robust controller is 0.17, against 0.28 of the optimal controller with the same path and same conditions. This is a big difference and concludes that the self-tuning regulator produces the best results regarding accuracy.



(a)



(b)

Figure 4.32 Curvature comparison in km^{-1}. (a) Curvature generate by the robust digital controller. (b) Curvature generated by the optimal controller.

Chapter 5

Conclusions

The dynamics of off-road vehicles are constantly changing due to different factors such as soil irregularities, tooling and changes in the speed, making the design and implementation of autonomous path following systems a very complex task.

To solve this issue, this research presented first a cascade system with a self-tuning regulator applied to the yaw rate dynamics in the inner loop, together with a gain scheduler used for the PID parameters in the outer loop controlling the lateral position. Nevertheless, only the implementation of an adaptive controller is not enough to account for uncertainties on the terrain. Therefore, a robust and adaptive digital method directly used for the control of the lateral position was implemented. This approach should not only cover for changes in the vehicle dynamics due to changes in the speed and in the interaction with the soil, it also covers different types of vehicles such as Ackermann-steering and skid-steering. The method is based on a digital RST pole placement and introduces fixed parts to the Bezout polynomial in order to shape the different sensitivity functions to guarantee robustness against uncertainties due to noise and disturbances as well as non-linearities not considered in the model. To tune the controller, two second-order systems are used for modeling desired tracking and regulation dynamics respectively. Since in this approach, the changes in the speed

affect both tracking and regulation dynamics, a gain scheduler is in charge of picking the desired parameters for the tracking and regulation models according to the measured speed. In addition to the robustness added to the design, a gradient estimator can be applied for the yaw rate dynamics to adapt the system to the soil irregularities, i.e., the parameters of the nominal model of the vehicle will be shifted by the estimator depending on the soil changes in order to increase the accuracy of the controller. Since the robustness of the system should already cover for uncertainties, this estimator can be switched off and use instead some preferred static model or previously identified model parameters. The results presented demonstrated the robustness of the controller and its ability to react nicely against terrain irregularities and disturbances. It was also possible to implement this linear robust digital solution into an ECU with limited resources and obtain similar results to more complex, even nonlinear approaches presented in the literature.

Finally, an observer-based optimal controller was used as a general benchmark and the results produced by this controller were less accurate than those from the adaptive-robust controller. Nevertheless, the control signal showed its ability to save energy and increase the life of the actuators due to the lack of high frequency elements. This advantage can be used in some cases where high accuracy is not necessarily required at the turning points.

5.1 Discussion of results

It should be first mentioned, that even though the objectives presented at the beginning of the document were achieved in a general way, those were changing throughout the natural evolution of the research. More precisely, at the beginning of the research only one general goal of finding a universal adaptive controller for all vehicles was set. As the research was mov-

ing forward, it was quite clear that more specific and realistic goals were needed. For instance, only an adaptive approach was not enough and a robust design was taken into account. Also, the decision of leaving nonlinear controllers out of the scope of this research was taken. Finally, the original objective was to perform all the tests with a tractor with varying hitch forces. Nevertheless, the circumstances led to the implementation using a skid-steering robot which, on the one hand, allowed to verify that the design can be used in different types of vehicles and, on the other hand, reduced the amount of resources needed in terms of machinery, equipment and field tests. This will also help to better understand the breakdown of the following discussion of results.

The first adaptive controller examined in this research is the cascade self-tuning regulator. It was tested for a tractor with hitch forces and uses recursive least-squares for the estimation of the yaw rate dynamics. The design used was an online MDPP of a RST controller. Even though the results were positive, there were different aspects that could be taken into account for improvement. The first aspect is the fact that the controller was applied only to the yaw rate dynamics. That brought us to the need of using a cascade controller where the inner part was the yaw rate controller and the outer one was a lateral position PID controller. The latter is not adaptable per se, and a tuning action has to be performed for different speeds. This increases the system complexity and the time and resources needed in the implementation phase. The second aspect is the usage of a MDPP, which limited the system to be of a second order. That perhaps is not ideal, since a bigger order system can take into account the steering and other dynamics not considered in a 2nd order one. For instance, a model of a big tractor with hitch forces will need to include tire relaxation if one wants to control the lateral position at higher speeds. This will naturally increase

the order of the system to be controlled (Kayacan et al., 2015; Owen and Bernard, 1982; Rolla, 1983).

This led to the need of improving the digital RST controller, by making it robust and adaptive. Instead of using a MDPP, we let the order of the system be as big as needed, and we also add some fixed parts to the characteristic equation in order to shape the sensitivity functions, introducing robustness to our design. Also, since the order of the system is no limited, the need of using a cascade is not necessary anymore, and the controller can cover the lateral position as well. This means that instead of looking for the ideal PID parameters for every different forward velocity, we simply look for the second order characteristic desired for each speed, i.e., a gain scheduler is introduced, and by picking de damping $\zeta_r = 1$, we only need to adapt the natural frequency as a function of the forward speed ($\omega_r = f(V_x)$). We also added a tracking model to shape the input step, which can be also adapted through the gain scheduler to the different speeds ($\omega_t = f(V_x)$).

The results shown by the robust and adaptive digital controller were very satisfactory. Nevertheless, in order to benchmark the results, we implemented an optimal digital controller which has shown very good path following performance and robustness against changes in the vehicle dynamics (Sharp, 2005). This controller seemed to be more stable against disturbances, since the actuators (motors) were not eager to jump to every change in the soil. Nevertheless it took longer to the vehicle to get into the track, once it was shifted by a big disturbance. In contrast, the robust adaptive controller showed more accuracy once on the track, and was faster at getting into the track with the drawback of being very responsive to every small disturbance (see Fig. 4.19 vs. Fig. 4.29). Nevertheless, the fixed parts introduced to the characteristic equation to make the system robust, allowed us to tune the input sensitivity function and limit the actuators re-

sponse, making the controller to behave similar to the optimal one.

Finally, for the estimation used in the robust self-tuning regulator, both approaches were tested on the field (recursive least-squares and online gradient). Even though the recursive least-squares were performing good in the simulations, its real implementation was making the system unstable, and it was not really possible to further proceed with the field test without putting in risk the integrity of the skid-steering robot. In contrast, the online gradient algorithm produced good results (see Fig. 4.24), especially when keeping the learning rate α, small enough so the estimation does not produced unstable results with the combination of turning curves and high soil irregularities.

5.2 Future work

A very important feature that gives the system adaptation, is the online system identification. This topic is very extensive and ideally, it will need a study to check and compare the different approaches, to assess which ones identify the changes in the dynamics of the vehicle more accurately depending on its characteristics (size, form, range of working speeds etc.). Also, we could introduce a disturbance frequency measurement, and automatically use those frequencies to the fixed parts of the sensitivity functions. Another issue to consider as future work is introducing the heading to the controller. At the moment, by controlling the lateral position, we are considering indirectly the heading of the vehicle. Nevertheless, by specifically taking into account the heading as a control output, we could improve the performance on the turning points as well as on the track. This implies indirectly the need of working with a MIMO system. Another topic that can be considered for future work is to extend the controller introducing predictive control to cover for slow time responses in bigger machines. For instance a tractor with two-meter-diameter

wheels, will react slower to an input step and the machine will lose 1 or 2 s to get in track again. Also, a closed loop adaptive non-linear system could be an alternative to this work, by using e.g. artificial neural networks. Nevertheless, the question regarding this approach remains, if the non-linearities can be effectively identified online, and if the system remains stable. Finally, the focus of the problem should be extended from path following to trajectory tracking to include the forward speed and increase therefore the performance and accuracy of the autonomous vehicle.

Bibliography

Ahmed, S. A. and Petrov, M. G. (2015). Trajectory Control of Mobile Robots using Type-2 Fuzzy-Neural PID Controller. *IFAC-PapersOnLine*, 48(24):138–143.

Aithal, H. and Janardhanan, S. (2013). Trajectory tracking of two wheeled mobile robot using higher order sliding mode control. In *2013 International Conference on Control, Computing, Communication and Materials (ICCCCM)*, pages 1–4.

Al-Milli, S., Seneviratne, L. D., and Althoefer, K. (2010). Track-terrain modelling and traversability prediction for tracked vehicles on soft terrain. *Journal of Terramechanics*, 47(3):151–160.

Amer, N. H., Zamzuri, H., Hudha, K., Aparow, V. R., Kadir, Z. A., and Abidin, A. F. Z. (2016). Modelling and trajectory following of an armoured vehicle. In *2016 SICE International Symposium on Control Systems (ISCS)*, pages 1–6.

Amer, N. H., Zamzuri, H., Hudha, K., and Kadir, Z. A. (2017). Modelling and control strategies in path tracking control for autonomous ground vehicles: A review of state of the art and challenges. *Journal of Intelligent & Robotic Systems*, 86(2):225–254.

Amidi, O. (1990). Integrated mobile robot control. Technical Report CMU-RI-TR-90-17, Carnegie Mellon University, Pittsburgh, PA.

Arslan, S. and Temeltas, H. (2011). Robust motion control of a four wheel drive skid-steered mobile robot. *2011 7th International Conference on Electrical and Electronics Engineering (ELECO)*, pages II–415–II–419.

Åström, K. J. and Wittenmark, B. (1994). *Adaptive Control*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 2nd edition.

Åström, K. J. and Wittenmark, B. (1997). *Computer-controlled Systems: Theory and Design*. Prentice-Hall information and system sciences series. Prentice Hall.

Auat Cheein, F. and Carelli, R. (2013). Agricultural Robotics: Unmanned Robotic Service Units in Agricultural Tasks. *IEEE Industrial Electronics Magazine*, 7(3):48–58.

Backman, J., Oksanen, T., and Visala, A. (2012). Navigation system for agricultural machines: Nonlinear Model Predictive path tracking. *Computers and Electronics in Agriculture*, 82:32–43.

Bahadorian, M., Savkovic, B., Eaton, R., and Hesketh, T. (2012). Robust Model Predictive Control for automated trajectory tracking of an Unmanned Ground Vehicle. *2012 American Control Conference (ACC)*, page 4251.

Barton, M. J. (2001). *Controller Development and Implementation for Path Planning and Following in an Autonomous Urban Vehicle*. Bachelor's Thesis, School of Aerospace, Mechanical and Mechatronic Engineering, The University of Sydney.

Baslamisli, S. C., Polat, I., and Kose, I. E. (2007). Gain Scheduled Active Steering Control Based on a Parametric Bicycle Model. *2007 IEEE Intelligent Vehicles Symposium*, 2(1):1168–1173.

Bayar, G., Bergerman, M., Konukseven, E. i., and Koku, A. B. (2016a). Improving the trajectory tracking performance of autonomous orchard vehicles using wheel slip compensation. *Biosystems Engineering*, 146:149–164.

Bayar, G., Bergerman, M., Konukseven, E. i., and Koku, A. B. (2016b). Improving the trajectory tracking performance of autonomous orchard vehicles using wheel slip compensation. *Biosystems Engineering*, 146:149–164.

Bergerman, M., Maeta, S. M., Zhang, J., Freitas, G. M., Hamner, B., Singh, S., and Kantor, G. (2015). Robot farmers: Autonomous orchard vehicles help tree fruit production. *IEEE Robotics and Automation Magazine*, 22(1):54–63.

Blender, T., Buchner, T., Fernandez, B., Pichlmaier, B., and Schlegel, C. (2016). Managing a Mobile Agricultural Robot Swarm for a seeding task. In *IECON 2016 – 42nd Annual Conference of the IEEE Industrial Electronics Society*, pages 6879–6886.

Caldwell, T. M. and Murphey, T. D. (2011). Switching mode generation and optimal estimation with application to skid-steering. *Automatica*, 47(1):50–64.

Canale, M., Fagiano, L., Ferrara, A., and Vecchio, C. (2009). Comparing internal model control and sliding-mode approaches for vehicle yaw control. *IEEE Transactions on Intelligent Transportation Systems*, 10(1):31–41.

Candeloro, M., Lekkas, A. M., Sørensen, A. J., and Fossen, T. I. (2013). Continuous curvature path planning using voronoi diagrams and fermat's spirals. In *IFAC Proceedings Volumes (IFAC-PapersOnline)*, volume 9, pages 132–137.

Cao, Q., Zhou, Z., and Zhang, M. (2017). Fuzzy Adaptive Shift Schedule of Tractor Subjected to Random Load. *Mathematical Problems in Engineering*, 2017:1–8.

Caracciolo, L., de Luca, A., and Iannitti, S. (1999). Trajectory tracking control of a four-wheel differentially driven mobile robot. In *Proceedings 1999 IEEE International Conference on Robotics and Automation (Cat. No.99CH36288C)*, volume 4, pages 2632–2638 vol.4.

Chen, B. (2011). Adaptive Path Following Control of Car-like Mobile Robot using Dynamic Model. *2011 6th IEEE Conference on Industrial Electronics and Applications*, pages 239–244.

Chen, C. (2014). *On the Robustness of the Linear Quadratic Regulator via Perturbation Analysis of the Riccati Equation*. PhD thesis, School of Electronic Engineering, Dublin City University, Glasnevin, Dublin 9, Dublin, Irland.

Chen, C. and Holohan, A. (2014a). A revisit of the stability robustness of linear quadratic regulators. In *25th IET Irish Signals Systems Conference 2014 and 2014 China-Ireland International Conference on Information and Communications Technologies (ISSC 2014/CIICT 2014)*, pages 310–315.

Chen, C. and Holohan, A. (2014b). Robustness results of LQR-based systems with structured uncertainty. In *25th IET Irish Signals Systems Conference 2014 and 2014 China-Ireland International Conference on Information and Communications Technologies (ISSC 2014/CIICT 2014)*, pages 70–73.

Chen, Q., Sun, Z., Liu, D., and Li, X. (2014). Ribbon model based path tracking method for autonomous ground vehicles. *Journal of Central South University*, 21(5):1816–1826.

Choset, H. (2001). Coverage for robotics – A survey of recent results. *Annals of Mathematics and Artificial Intelligence*, 31(1–4):113–126.

Chou, C. and Verhaegen, M. (1997). Subspace algorithms for the identification of multivariable dynamic errors-in-variables models. *Automatica*, 33(10):1857–1869.

Coulter, R. C. (1992). Implementation of the pure pursuit path tracking algorithm. Technical Report CMU-RI-TR-92-01, Carnegie Mellon University, Pittsburgh, PA.

Dagci, O. H., Umit, Y., and Ozgiiner, U. (2003). Path Following Controller Design Using Sliding Mode Control Theory. *American Control Conference, Denver, Colorado*, pages 903–908.

Das, T., Kar, I. N., and Chaudhury, S. (2006). Simple neuron-based adaptive controller for a nonholonomic mobile robot including actuator dynamics. *Neurocomputing*, 69(16–18):2140–2151.

De Luca, A., Oriolo, G., and Samson, C. (1998). *Feedback control of a nonholonomic car-like robot*, pages 171–253. Springer Berlin Heidelberg, Berlin, Heidelberg.

Derrick, J. and Bevly, D. (2008). Adaptive control of a farm tractor with varying yaw dynamics accounting for actuator dynamics and saturations. In *Control Applications, 2008. CCA 2008. IEEE International Conference on*, pages 547–552.

Derrick, J., Bevly, D., and Rekow, A. (2008). Model-reference adaptive steering control of a farm tractor with varying hitch forces. In *American Control Conference, 2008*, pages 3677–3682.

Derrick, J. B. and Bevly, D. M. (2009). Adaptive steering control of a farm tractor with varying yaw rate properties. *Journal of Field Robotics*, 26(6-7):519–536.

Dubins, L. E. (1957). On curves of minimal length with a constraint on average curvature, and with prescribed initial and terminal positions and tangents. *American Journal of Mathematics*, 79(3):497–516.

Falcone, P., Borrelli, F., Asgari, J., Tseng, H. E., and Hrovat, D. (2007). Predictive active steering control for autonomous vehicle systems. *IEEE Transactions on Control Systems Technology*, 15(3):566–580.

Fang, H., Lenain, R., Thuilot, B., and Martinet, P. (2005). Trajectory tracking control of farm vehicles in presence of sliding. In *2005 IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS*, volume 54, pages 1720–1725.

Faress, K. N., El hagry, M. T., and El kosy, A. A. (2005). Trajectory tracking control for a wheeled mobile robot using fuzzy logic controller. In *Proceedings of the 9th WSEAS International Conference on Systems*, ICS'05, pages 9:1–9:5, Stevens Point, Wisconsin, USA. World Scientific and Engineering Academy and Society (WSEAS).

Fernandez, B., Cerrada, J., and Gross, J. (2016). Tractor-Implement Real Time Interactive 3D Simulation Based on Openframeworks and OpenGL. In *International Conference on Machine Control & Guidance, 2016*, pages 3677–3682.

Fernandez, B., Herrera, P. J., and Cerrada, J. A. (2018a). Self-tuning regulator for a tractor with varying speed and hitch forces. *Computers and Electronics in Agriculture*, 145:282–288.

Fernandez, B., Herrera, P. J., and Cerrada, J. A. (2018b). Robust digital control for autonomous skid-steered agricultural robots. *Computers and Electronics in Agriculture*, 153:94–101.

Fierro, R. and Lewis, F. L. (1995). Control of a nonholonomic mobile robot: backstepping kinematics into dynamics. In *Proceedings of 1995 34th IEEE Conference on Decision and Control*, volume 4, pages 3805–3810.

Foley, J. (2017). Feeding the World - National Geographic. http://www.nationalgeographic.com/foodfeatures/feeding-9-billion/.

Forssell, U. and Ljung, L. (1999). Closed-loop identification revisited. *Automatica*, 35(7):1215–1241.

Fukao, T., Nakagawa, H., and Adachi, N. (2000). Adaptive tracking control of a nonholonomic mobile robot. *IEEE Transactions on Robotics and Automation*, 16(5):609–615.

Galceran, E. and Carreras, M. (2013). A survey on coverage path planning for robotics. *Robotics and Autonomous Systems*, 61(12):1258–1276.

Gartley, E. (2005). On-line estimation of implement dynamics for adaptive steering control of farm tractors. Master's thesis.

Gartley, E. and Bevly, D. (2008). Online Estimation of Implement Dynamics for Adaptive Steering Control of Farm Tractors. *IEEE/ASME Transactions on Mechatronics*.

Gerrish, J. and Surbrook, T. (1984). Mobile robots in agriculture. In *ASAE Publication*, pages 30–41. ASAE.

Gerrish, J. B., Fehr, B. W., Ee, G. R. V., and Welch, D. P. (1997). Self-steering tractor guidance by computer-vision. In *Applied Engineering in Agriculture*, volume 13, pages 559–563.

Gerrish, J. B., Stockman, G. C., Mann, L., and Hu, G. (1986). Path-finding by image processing in agricultural field operations. *SAE Transactions*, 95:540–554.

Gillespie, T. (1992). *Fundamentals of Vehicle Dynamics*. PA: Society of Automotive Engineers, Warrendale.

Guo, T. and Peng, H. (2013). A simplified skid-steering model for torque and power analysis of tracked small unmanned ground vehicles. In *2013 American Control Conference*, pages 1106–1111.

Hellström, T. and Ringdahl, O. (2006). Follow the past: a path tracking algorithm for autonomous vehicles. *International Journal of Vehicle Autonomous Systems*, 4(2/3/4):216–224.

Hima, S., Lusetti, B., Vanholme, B., Glaser, S., and Mammar, S. (2011). Trajectory tracking for highly automated passenger vehicles. In Bittanti, Sergio, C. A. Z. S., editor, *18th World Congress of the International Federation of Automatic Control (IFAC 2011)*, volume 18, page elec. proc., Milano, Italy.

Hoffmann, G. M., Tomlin, C. J., Montemerlo, M., and Thrun, S. (2007). Autonomous automobile trajectory tracking for off-road driving: Controller design, experimental validation and racing. In *2007 American Control Conference*, pages 2296–2301.

Hwang, E., Kang, H., Hyun, C., and Park, M. (2013). Robust backstepping control based on a Lyapunov redesign for Skid-Steered Wheeled Mobile Robots. *International Journal of Advanced Robotic Systems*, 10(1):26.

Iagnemma, K., Kang, S., Shibly, H., and Dubowsky, S. (2004). Online terrain parameter estimation for wheeled mobile robots with application to planetary rovers. *IEEE Transactions on Robotics*, 20(5):921–927.

Inoue, R. S., Cerri, J. P., Terra, M. H., and Siqueira, A. A. G. (2013). Robust recursive control of a skid-steering mobile robot. In *2013 16th International Conference on Advanced Robotics (ICAR)*, pages 1–6.

Jansson, M. and Wahlberg, B. (1996). A linear regression approach to state-space subspace system identification. *Signal Processing*, 52(2):103–129. Subspace Methods, Part II: System Identification.

Jeon, S. and Jeong, W. (2015). The Stable Trajectory Tracking Control of a Skid-Steered Mobile Platform with Dynamic Uncertainties. *International Journal of Advanced Robotic Systems*, 12(9):1–9.

Jiangdagger, Z. and Nijmeijer, H. (1997). Tracking control of mobile robots: A case study in backstepping. *Automatica*, 33(7):1393–1399.

Jinlin, X. and Liming, X. (2010). Autonomous agricultural robot and its row guidance. In *Measuring Technology and Mechatronics Automation (ICMTMA), 2010 International Conference on*, volume 1, pages 725–729.

Jun, J.-Y., Hua, M.-D., and Benamar, F. (2014). A trajectory tracking control design for a skid-steering mobile robot by adapting its desired instantaneous center of rotation. In *53rd IEEE Conference on Decision and Control*, pages 4554–4559. IEEE.

Kanade, T., Thorpe, C., and Whittaker, W. (1986). Autonomous land vehicle project at cmu. In *Proceedings of the 1986 ACM Fourteenth Annual Conference on Computer Science*, CSC '86, pages 71–80, New York, NY, USA. ACM.

Kanayama, Y., Kimura, Y., Miyazaki, F., and Noguchi, T. (1990). A stable tracking control method for an autonomous mobile robot. In *Proceedings., IEEE International Conference on Robotics and Automation*, volume 1, pages 384–389.

Karkee, M. and Steward, B. L. (2011). Parameter estimation and validation of a tractor and single axle towed implement dynamic system model. *Computers and Electronics in Agriculture*, 77(2):135–146.

Kayacan, E., Kayacan, E., Ramon, H., and Saeys, W. (2015). Towards agrobots: Identification of the yaw dynamics and trajectory tracking of an autonomous tractor. *Computers and Electronics in Agriculture*, 115:78–87.

Kayacan, E., Saeys, W., Kayacan, E., and Ramon, H. (2012). Intelligent control of a tractor–implement system using type-2 fuzzy neural networks. pages 1–8. IEEE.

Keviczky, T., Falcone, P., Borrelli, F., Asgari, J., and Hrovat, D. (2006). Predictive control approach to autonomous vehicle steering. In *2006 American Control Conference*, pages 6 pp.–.

Khatib, E. I. A., Al–Masri, W. M. F., Mukhopadhyay, S., Jaradat, M. A., and Abdel–Hafez, M. (2015). A comparison of adaptive trajectory tracking controllers for wheeled mobile robots. *Mechatronics and its Applications (ISMA), 2015 10th International Symposium on*, pages 1–6.

Kigezi, T. N., Alexandru, S., Mugabi, E., and Musasizi, P. I. (2015). Sliding mode control for tracking of nonholonomic wheeled mobile robots. In *2015 5th Australian Control Conference (AUCC)*, pages 21–26.

Klančar, G. and Škrjanc, I. (2007). Tracking–error model–based predictive control for mobile robots in real time. *Robotics and Autonomous Systems*, 55(6):460 – 469.

Kohanbash, D., Bergerman, M., Lewis, K. M., and Moorehead, S. J. (2012). A safety architecture for autonomous agricultural vehicles. In *American Society of Agricultural and Biological Engineers Annual Meeting*, Pittsburgh, PA.

Kühne, F. (2005). Mobile robot trajectory tracking using model predictive control. Universidade Federal do Rio Grande do Sul. Departamento de Engenharia Electrica.

Kuhne, F., Lages, W. F., and da Silva, J. M. G. (2005). Point stabilization of mobile robots with nonlinear model predictive control. In *IEEE International Conference Mechatronics and Automation, 2005*, volume 3, pages 1163–1168 Vol. 3.

Landau, I. D. and Karimi, A. (1998). Robust Digital Control using Pole Placement with Sensitivity Function Shaping Method. *International Journal of Robust and Nonlinear Control*, 8:191–210.

Landau, I. D., Lozano, R., M'Saad, M., and Karimi, A. (2011). *Adaptive Control. Algorithms, Analysis and Applications*. Number c. Springer, 2 edition.

Landau, I. D. and Zito, G. (2006). *Digital control systems : design, identification and implementation*. Springer.

Langer, J. and Constantinescu, A. (1999). Pole placement design using convex optimisation criteria for the flexible transmission benchmark. *European Journal of Control*, 5(2):193–207.

Langer, J. and Landau, I. (1999). Combined pole placement/sensitivity function shaping method using convex optimization criteria. *Automatica*, 35(6):1111–1120.

Larimore, W. E. (1990). Canonical variate analysis in identification, filtering, and adaptive control. In *29th IEEE Conference on Decision and Control*, volume 2, pages 596–604.

Lenain, R., Thuilot, B., Cariou, C., and Martinet, P. (2003). Adaptive control for car like vehicles guidance relying on RTK GPS: rejection of sliding effects in agricultural applications. In *ICRA Proc. IEEE International Conference on Robotics and Automation.*, volume 1, pages 115–120.

Lenain, R., Thuilot, B., Cariou, C., and Martinet, P. (2007). Adaptive and predictive path tracking control for off-road mobile robots. *European Journal of Control*, 13(4):419–439.

Li, T. and Hu, J. (2014). Adaptive Sliding Mode Control Method Based on Nonlinear Integral Sliding Surface for Agricultural Vehicle Steering Control. *Mathematical Problems in Engineering*, 2014:1–10.

Li, X., Wang, Z., Zhu, J., and Chen, Q. (2015). Adaptive tracking control for wheeled mobile robots with unknown skidding. In *2015 IEEE Conference on Control Applications (CCA)*, pages 1674–1679.

Liao, X., Sun, Z., Weng, L., Li, B., Song, Y., and Li, Y. (2006). Adaptive neural network path tracking of unmanned ground vehicle. In Wang, J., Yi, Z., Zurada, J. M., Lu, B.-L., and Yin, H., editors, *Advances in Neural Networks – ISNN 2006*, pages 1233–1238, Berlin, Heidelberg. Springer Berlin Heidelberg.

Liu, N. and Alleyne, A. G. (2014). Iterative Learning Identification Applied to Automated Off-Highway Vehicle. *IEEE Transactions on Control Systems Technology*, 22(1):331–337.

Luck, J., Pitla, S., Shearer, S., Mueller, T., Dillon, C., Fulton, J., and Higgins, S. (2010). Potential for pesticide and nutrient savings via map-based automatic boom section control of spray nozzles. *Computers and Electronics in Agriculture*, 70(1):19–26.

MacAdam, C. C. (1980). An Optimal Preview Control for Linear Systems. *Journal of Dynamic Systems, Measurement, and Control*, 102(3):188.

MacAdam, C. C. (1981). Application of an optimal preview control for simulation of closed-loop automobile driving. *IEEE Transactions on Systems, Man, and Cybernetics*, 11(6):393–399.

Maclaurin, B. (2011). A skid steering model using the Magic Formula. *Journal of Terramechanics*, 48:247–263.

Martin, T. C., Orchard, M. E., and Sánchez, P. V. (2013). Design and simulation of control strategies for trajectory tracking in an autonomous ground vehicle. In *6th IFAC Conference on Management and Control of Production and Logistics*, volume 46, pages 118 – 123.

Martínez, J. L., Mandow, A., Morales, J., Pedraza, S., and García-Cerezo, A. (2005). Approximating kinematics for tracked mobile robots. *International Journal of Robotics Research*, 24(10):867–878.

Martins, F. N., Celeste, W. C., Carelli, R., Sarcinelli-Filho, M., and Bastos-Filho, T. F. (2008). An adaptive dynamic controller for autonomous mobile robot trajectory tracking. *Control Engineering Practice*, 16(11):1354–1363.

Matveev, A. S., Hoy, M., Katupitiya, J., and Savkin, A. V. (2013). Nonlinear sliding mode control of an unmanned agricultural tractor in the presence of sliding and control saturation. *Robotics and Autonomous Systems*, 61(9):973–987.

Merabti, H., Belarbi, K., and Bouchemal, B. (2016). Nonlinear predictive control of a mobile robot: a solution using metaheuristcs. *Journal of the Chinese Institute of Engineers*, 39(3):282–290.

Michael O'Connor, T. B., Elkaim, G., and Parkinson, D. B. (1996). Automatic steering of farm vehicles using gps. In *In 3rd Intl. Conf. Precision Agriculture*.

Moorehead, S. J., Wellington, C. K., Gilmore, B. J., and Vallespi, C. (2012). Automating orchards: A system of autonomous tractors for orchard maintenance. In *Workshop on Agricultural Robotics: Enabling Safe, Efficient, and Affordable Robots for Food Production (Collocated with IROS 2012), Vilamoura, Portugal*.

Mousazadeh, H. (2013). A technical review on navigation systems of agricultural autonomous off-road vehicles. *Journal of Terramechanics*, 50(3):211–232.

Noh, K.-M. (1990). *Self-tuning Controller for Farm Tractor Guidance*. PhD thesis.

Noh, K.-M. and Erbach, D. C. (1993). Self-tuning Controller for Farm Tractor Guidance. *Transactions of the ASAE*, 36(6):1583–1594.

Normey-Rico, J. E., Alcala, I., Gomez-Ortega, J., and Camacho, E. F. (2001). Mobile robot path tracking using a robust PID controller. *Control Engineering Practice*, 9(11):1209–1214.

O'Cornnor, M. L. (1997). Carrier-Phase Differential Gps for Automatic Control of Land Vehicles. *Stanford University Press*, (December).

Ogata, K. (1995). *Discrete-time Control Systems (2Nd Ed.)*. Prentice-Hall, Inc., Upper Saddle River, NJ, USA.

Ollis, M. and Stentz, A. (1996). First results in vision-based crop line tracking. In *Proceedings of IEEE International Conference on Robotics and Automation*, volume 1, pages 951–956.

Overschee, P. V. and Moor, B. D. (1994). N4sid: Subspace algorithms for the identification of combined deterministic-stochastic systems. *Automatica*, 30(1):75–93. Special issue on statistical signal processing and control.

Owen, R. H. and Bernard, J. E. (1982). Directional dynamics of a tractor-loader-backhoe. *Vehicle System Dynamics*, 11(5-6):251–265.

Pan, H., Zhang, Y., and Sun, W. (2014). Robust Tracking Control for Vehicle Lateral Dynamics with Uncertain Parameters and External Nonlinearities. *Shock and Vibration*, 2014(Article ID e324018):12.

Pazderski, D. and Kozłowski, K. (2008). Trajectory tracking control of skid-steering robot - experimental validation. *IFAC Proceedings Volumes*, 41(2):5377 - 5382. 17th IFAC World Congress.

Pazderskit, D., Kozlowski, K., and Dixon, W. E. (2004). Tracking and Regulation Control of a Skid Steering Vehicle. In *American Nuclear Society Tenth International Topical Meeting on Robotics and Remote Systems*, number 82797, pages 369–376.

Ping, E. P., Hudha, K., Harun, M. H. B., and Jamaluddin, H. (2010). Hardware-in-the-loop simulation of automatic steering control: Outer-loop and inner-loop control design. In *2010 11th International Conference on Control Automation Robotics Vision*, pages 964–969.

Pomerleau, D. a. (1989). Alvinn: An autonomous land vehicle in a neural network. *Advances in Neural Information Processing Systems 1*, pages 305–313.

Qin, S. J. (2006). An overview of subspace identification. *Computers & Chemical Engineering*, 30(10):1502–1513. Papers form Chemical Process Control VII.

Raffo, G. V., Gomes, G. K., Normey-Rico, J. E., Kelber, C. R., and Becker, L. B. (2009). A predictive controller for autonomous vehicle path tracking. *IEEE Transactions on Intelligent Transportation Systems*, 10(1):92–102.

Rankin, A. L., Crane, C. D., Armstrong, D. G., Nease, A. D., and Brown, H. E. (1996). Autonomous path-planning navigation system for site characterization. In *Proc. SPIE Navigation and Control Technologies for Unmanned Systems*, volume 2738, pages 176–186.

Rolla, D. A. C. (1983). The steering behaviour of off-road vehicles. *Vehicle System Dynamics*, 12(1-3):39-42.

Rossetter, E. J. (2003). *A Potential Field Framework for Active Vehicle Lanekeeping Assistance*. PhD thesis, Stanford, CA, USA.

Roy, S., Nandy, S., Ray, R., and Shome, S. N. (2015). Robust path tracking control of nonholonomic wheeled mobile robot: Experimental validation. *International Journal of Control, Automation and Systems*, 13(4):897-905.

Scaglia, G., Rosales, A., Quintero, L., Mut, V., and Agarwal, R. (2010). A linear-interpolation-based controller design for trajectory tracking of mobile robots. *Control Engineering Practice*, 18(3):318 - 329.

Scaglia, G., Serrano, E., Rosales, A., and Albertos, P. (2015). Linear interpolation based controller design for trajectory tracking under uncertainties: Application to mobile robots. *Control Engineering Practice*, 45:123 - 132.

Scheuer, a. and Fraichard, T. (1997). Continuous-Curvature Path Planning for Multiple Car-Like Vehicles 2 Related Works 1 Introduction. *Recherche*, pages 8-12.

Searcy, S. W. and Reid, J. F. (1986). Vision-based guidance of an agricultural tractor. In *1986 American Control Conference*, pages 931-937.

Shah, A. N., Tanveer, M., Shahzad, B., Yang, G., Fahad, S., Ali, S., Bukhari, M. A., Tung, S. A., Hafeez, A., and Souliyanonh, B. (2017). Soil compaction effects on soil health and cropproductivity: an overview. *Environmental Science and Pollution Research*, pages 1-12.

Sharda, A., Luck, J. D., Fulton, J. P., Mcdonald, T. P., and Shearer, S. A. (2013). Field application uniformity and accuracy of two rate control systems with automatic section capabilities on agricultural sprayers. *Precision Agriculture; Dordrecht*, 14(3):307-322.

Sharp, R. S. (2005). Driver steering control and a new perspective on car handling qualities. *Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science*, 219(10):1041-1051.

Sharp, R. S. (2012). Rider control of a motorcycle near to its cornering limits. *Vehicle System Dynamics*, 50(8):1193-1208.

Snider, J. M. (2009). Automatic steering methods for autonomous automobile path tracking. Technical Report CMU-RI-TR-09-08, Carnegie Mellon University, Pittsburgh, PA.

Solea, R. and Cernega, D. (2015). Super twisting sliding mode controller applied to a nonholonomic mobile robot. In *2015 19th International Conference on System Theory, Control and Computing (ICSTCC)*, pages 87-92.

Solea, R. and Nunes, U. (2007). Trajectory planning and sliding-mode control based trajectory-tracking for cybercars. *Integr. Comput.-Aided Eng.*, 14(1):33–47.

Stombaugh, T. S., Benson, E. R., and Hummel, J. W. (1998). Automatic guidance of agricultural vehicles at high field speeds. ASAE Annual International Meeting. Paper No. 983110.

Sun, Z., Chen, Q., Nie, Y., Liu, D., and He, H. (2012). Ribbon model based path tracking method for autonomous land vehicle. In *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 1220–1226.

Tao, G. (2014). Multivariable adaptive control: A survey. *Automatica*, 50(11):2737–2764.

Tchoń, K., Zadarnowska, K., Juszkiewicz, and Arent, K. (2015). Modeling and control of a skid-steering mobile platform with coupled side wheels. *Bulletin of the Polish Academy of Sciences: Technical Sciences*, 63(3):807–818.

Thrun, S., Montemerlo, M., Dahlkamp, H., Stavens, D., Aron, A., Diebel, J., Fong, P., Gale, J., Halpenny, M., Hoffmann, G., Lau, K., Oakley, C., Palatucci, M., Pratt, V., Stang, P., Strohband, S., Dupont, C., Jendrossek, L.-E., Koelen, C., Markey, C., Rummel, C., van Niekerk, J., Jensen, E., Alessandrini, P., Bradski, G., Davies, B., Ettinger, S., Kaehler, A., Nefian, A., and Mahoney, P. (2006). Stanley: The robot that won the darpa grand challenge. *Journal of Field Robotics*, 23(9):661–692.

U.Nations (2013). World population projected to reach 9.6 billion by 2050, UN DESA, United Nations Department of Economic and Social Affairs.

Verhaegen, M. and Dewilde, P. (1992). Subspace model identification part 1. the output-error state-space model identification class of algorithms. *International Journal of Control*, 56(5):1187–1210.

Vroegindeweij, B. A., Blaauw, S. K., IJsselmuiden, J. M., and van Henten, E. J. (2018). Evaluation of the performance of poultrybot, an autonomous mobile robotic platform for poultry houses. *Biosystems Engineering*, 174:295–315.

Wai, R.-J. and Liu, C.-M. (2009). Experimental verification of dynamic Petri recurrent-fuzzy-neural-network path tracking control design for mobile robot. *2009 IEEE International Conference on Control and Automation*, pages 1359–1364.

Wai, R.-J., Liu, C.-M., and Lin, Y.-W. (2010). Robust path tracking control of mobile robot via dynamic petri recurrent fuzzy neural network. *Soft Computing*, 15(4):743–767.

Wallace, R., Stentz, A., Thorpe, C., Maravec, H., Whittaker, W., and Kanade, T. (1985). First results in robot road-following.

In *Proceedings of the 9th International Joint Conference on Artificial Intelligence - Volume 2*, IJCAI'85, pages 1089–1095, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.

Wang, J., Steiber, J., and Surampudi, B. (2008). Autonomous ground vehicle control system for high-speed and safe operation. In *2008 American Control Conference*, pages 218–223.

Wang, T., Wu, Y., Liang, J., Han, C., Chen, J., and Zhao, Q. (2015). Analysis and experimental kinematics of a skid-steering wheeled robot based on a laser scanner sensor. *Sensors (Switzerland)*, 15(5):9681–9702.

Wit, J., Crane, C. D., and Armstrong, D. (2004). Autonomous ground vehicle path tracking. *Journal of Robotic Systems*, 21(8):439–449.

Wit, J. S. (2000). *Vector Pursuit Path Tracking for Autonomous Ground Vehicles*. PhD thesis, Gainesville, FL, USA.

Xue, T., Li, R., Tokgo, M., Ri, J., and Han, G. (2017). Trajectory planning for autonomous mobile robot using a hybrid improved qpso algorithm. *Soft Computing*, 21(9):2421–2437.

Xueyuan, L., Yu, Z., Jibin, H., Chongbo, J., and Jing, G. (2013). Lateral dynamic simulation of skid steered wheeled vehicle. In *2013 IEEE Intelligent Vehicles Symposium (IV)*, pages 1095–1100. IEEE.

Yang, J.-M. and Kim, J.-H. (1999). Sliding mode control for trajectory tracking of nonholonomic wheeled mobile robots. *IEEE Transactions on Robotics and Automation*, 15(3):578–587.

Ye, J. (2008). Tracking control for nonholonomic mobile robots: Integrating the analog neural network into the backstepping technique. *Neurocomputing*, 71(16):3373 - 3378. Advances in Neural Information Processing (ICONIP 2006) / Brazilian Symposium on Neural Networks (SBRN 2006).

Yi, J., Song, D., Zhang, J., and Goodwin, Z. (2007). Adaptive trajectory tracking control of skid-steered mobile robots. In *Proceedings 2007 IEEE International Conference on Robotics and Automation*, pages 2605–2610.

Yi, J., Wang, H., Zhang, J., Song, D., Jayasuriya, S., and Liu, J. (2009). Kinematic modeling and analysis of skid-steered mobile robots with applications to low-cost inertial measurement unit-based motion estimation. *IEEE Trans. Robotics*, 25(5):1087–1097.

Yoo, S. (2010). Adaptive tracking control for a class of wheeled mobile robots with unknown skidding and slipping. *IET Control Theory & Applications*, 4(10):2109–2119.

Yoo, S. J., Choi, Y. H., and Park, J. B. (2006). Generalized predictive control based on self-recurrent wavelet neural network

for stable path tracking of mobile robots: Adaptive learning rates approach. *IEEE Transactions on Circuits and Systems I: Regular Papers*, 53(6):1381–1394.

Zakaria, M. A., Zamzuri, H., Mamat, R., and Mazlan, S. A. (2013). A path tracking algorithm using future prediction control with spike detection for an autonomous vehicle robot. *International Journal of Advanced Robotic Systems*, 10(8):309.

Zakaria, M. A., Zamzuri, H., Mazlan, S. A., and Zainal, S. M. H. F. (2012). Vehicle path tracking using future prediction steering control. *Procedia Engineering*, 41:473 – 479. International Symposium on Robotics and Intelligent Sensors 2012 (IRIS 2012).

Zhang, M., Ma, W., Liu, Z., and Liu, G. (2013). Fuzzy-adaptive control method for off-road vehicle guidance system. *Mathematical and Computer Modelling*, 58(3-4):551–555.

Zhang, S., Wang, Y., Zhu, Z., Li, Z., Du, Y., and Mao, E. (2018). Tractor path tracking control based on binocular vision. *Information Processing in Agriculture*. https://doi.org/10.1016/j.inpa.2018.07.003.

Zhou, Q. Z. Q., Wang, F. W. F., and Li, L. (2005). Robust sliding mode control of 4WS vehicles for automatic path tracking. In *IEEE Proceedings. Intelligent Vehicles Symposium, 2005.*, pages 819–826.

Zhu, Z. X., Torisu, R., Takeda, J. I., Mao, E. R., and Zhang, Q. (2005). Neural network for estimating vehicle behaviour on sloping terrain. *Biosystems Engineering*, 91(4):403–411.