



DEGREE PROJECT IN ELECTRICAL ENGINEERING,
SECOND CYCLE, 30 CREDITS
STOCKHOLM, SWEDEN 2016

Model Predictive Control for Autonomous Driving of Over-Actuated Research Vehicle

GONÇALO COLLARES PEREIRA

Acknowledgements

I would like to thank my supervisor Jonas Mårtensson, for giving me an interesting topic and guidance throughout the whole thesis. I would also like to thank Pedro Lima, for all his valuable insights and tremendous help with all aspects of the thesis. Last but not least, I would like to thank all the team members that helped me make the research concept vehicle (RCV) work, Stefanos Kokogias, Lars Svensson, Rui Oliveira, Jonathan Fagerström and Petter Tomner.

Sammanfattning

Ända sedan personfordon blev tillgängliga för allmänheten, har trafikolyckor varit vanliga. För att motverka den här trenden talas det mer och mer om kolonkörning och samarbete mellan kommunicerande fordon, vilket kan komma att öka kapaciteten på vägarna samt minska utsläpp av växthusgaser. Grand Cooperative Driving Challenge (GCDC) är en tävling som hölls i Maj 2016 i Nederländerna, med fokus på den här sortens funktionalitet. Kungliga tekniska högskolan, KTH, deltog med sin research concept vehicle och tävlade mot andra Europeiska universitet.

Syftet med detta examensarbete var att designa och implementera longitudinell och lateral reglering av fordonet i.e reglering av vridmoment och styrvinkelar för att få fordonet att följa en referens. Denna funktionalitet gjorde att fordonet kunde framföras helt autonomt och möjliggjorde deltagande i alla tävlingsmoment.

Den laterala regulatorn blev aldrig implementerad och testad på RCV, utan presenteras istället i form av simuleringsresultat. Resultaten presenterar regulatorns prestanda i flera signifika scenarion ssom filbyte, korrigering av lateralt fel genom krabbstyrning, i.e sidledsförflyttning.

Den longitudinella regulatorn implementerades och testades i bilen och uppnådde statiskt fel nära 0 både i relativ hastighet och avstånd till framförvarande fordon. Fordonet reagerade som förväntat när den tvingades bromsa hastigt för att undvika kollision med framförvarande fordon, samt när målfordonet ändrades under gång.

Abstract

Since automobiles became available to everyone, road traffic and traffic accidents became common. To fight this trend, platooning and cooperative driving are more and more talked about. It helps increase the traffic capacity of roads and decreases greenhouse gas emissions. The grand cooperative driving challenge, GCDC, is a competition that was held May 2016 in the Netherlands focusing on this type of driving. The Royal Institute of Technology, KTH, participated in the competition with the research concept vehicle competing against other teams from different universities of Europe.

The objective of this thesis is to design and implement longitudinal and lateral control on the research concept vehicle, i.e., motor control by torque request and steering control by wheel angle request. Which makes it possible to enter the competition with a fully autonomous vehicle and participate in the three different scenarios.

The lateral controller, is not implemented and tested in the RCV, but it is still simulated, where the capabilities of the controller, such as, for example, lane changing and error correction using crabbing, i.e., sideways motion, are presented.

The longitudinal controller is fully implemented and tested, obtaining static errors very close to 0 in both velocity and distance to the front vehicle. The vehicle behaved as expected when it had to brake fast due to the front vehicle and it always kept more than the considered safety distance, this is rapidly corrected, to the right value, as soon as the front vehicle stops decelerating. It also behaved as expected, when the target being followed changes, i.e., when there is a jump in the distance being followed and it becomes small (0m), it decelerated smoothly and recovered the safety distance.

Contents

Glossary	viii
1 Introduction	1
1.1 Background	2
1.1.1 Research Concept Vehicle	2
1.2 Grand Cooperative Driving Challenge	4
1.2.1 Highway Scenario	4
1.2.2 Intersection Scenario	5
1.2.3 Emergency Scenario	5
1.3 Scope	9
1.4 Problem Formulation	9
1.5 Objective	10
1.6 Related Work/State of Art	10
1.7 Outline	12
2 Modeling	13
2.1 Vehicle Model	13
2.1.1 External Forces	15
2.1.2 Model Constant Values	17
2.2 Wheel Angles Transformation	17
3 Control Design	21
3.1 Model Predictive Control	21
3.2 Linear MPC	22
3.3 MPC for Reference Tracking	23
3.4 Quadratic Programming Formulation	24
3.5 Formulation	25
3.5.1 Longitudinal Controller	26
3.5.2 Lateral Controller	31
3.6 Reference Generation	35
3.6.1 Longitudinal	35
3.6.2 Lateral	37

4 Implementation	40
4.1 Controllers I/Os	40
4.2 Software	42
4.3 Prediction horizon and Controller Frequency	42
4.4 Controllers Specifications	43
4.4.1 Longitudinal	43
4.4.2 Lateral Controller	44
5 Results	46
5.1 Longitudinal	46
5.1.1 Platooning	47
5.1.2 Cruise Control	53
5.2 Lateral	60
6 Conclusions	63
7 Future Work	64
Bibliography	68
A A and B Matrices for Lateral Control	69

Glossary

ACC Adaptive Cruise Control

ADAS Advanced driver assistance systems

CACC Cooperative Adaptive Cruise Control

CAN Controller Area Network

CC Cruise Control

CoG Center of Gravity

GCDC Grand Cooperative Driving Challenge

LMPC Linear Model Predictive Control

LQR Linear-Quadratic regulator

LTV-MPC Linear Time-Varying Model Predictive Control

MABX Micro Auto Box

MPC Model Predictive Control

NMPC Nonlinear Model Predictive Control

PID Proportional Integral Derivative

QP Quadratic Programming

RCV Research Concept Vehicle

RSU Road Side Unit

V2V Vehicle to Vehicle

Nomenclature

$(.)_f$	Front Wheel
$(.)_r$	Rear Wheel
$(.)_{ilk}$	Time instant i predicted at instant k
$(.)_{ref}$	Reference Value
α	Slip angle
β	Crabbing Angle
\ddot{x}, \ddot{y}	Vehicle acceleration
δ	Wheel angle
$\dot{\psi}$	Vehicle yaw rate
\dot{x}, \dot{y}	Vehicle velocity
μ	Road friction coefficient
ψ	Vehicle yaw
ρ	Mass density of the air
θ	Angle of inclination of the road
a, b	Distance of front and rear wheels from CoG
A_F	Frontal Area of the vehicle
c	Curvature
C_d	Aerodynamic drag coefficient
C_f, C_r	Cornering stiffness of front and rear wheels
f	Prediction model
f	Roll resistance coefficient
F_a	Air drag force

F_l, F_c Longitudinal and Lateral tire Forces

F_r Rolling resistance force

F_x, F_y Forces in car body frame

F_z Normal tire load

g Gravitational Constant

H Prediction Horizon

I Vehicle Inertia

J Cost function

m Vehicle mass

m_I Equivalent mass of rotating parts due to inertia

R Radius of curvature

r Wheel radius

s Slip ratio

u Control signals

X, Y Vehicle position in world frame

x, y Vehicle position in local frame

z State of the vehicle

Chapter 1

Introduction

Nowadays, owning a car has become the new standard, which leads to more road traffic and traffic accidents. The increase in traffic is not only a problem because it leads to a waste of time, but also because it causes more air pollution, more energy consumption and it leads to more accidents [3, 4]. Besides the traffic congestion, traffic accidents also became more common with the increase in the total number of vehicles. Many accidents are caused by human error and by the way humans deal with situations or do not deal with them at all (e.g., reaction time is not enough, wrong reaction).

Technology keeps improving and these issues can be addressed. Cooperative vehicle lets vehicles interact with each other preventing traffic shock waves, where drivers are forced to decelerate [5]. Furthermore cooperative driving allows for the introduction of systems that replace the driver. The human error is taken out of the equation allowing for a more challenging driving approach, such as platooning at short distances, which saves energy and reduces pollution.

Almost every new vehicle has cruise control (CC) and they start having a variation of the latter, adaptive cruise control (ACC). ACC maintains a certain speed, as the normal CC, but is also able to follow a slower predecessor, which allows for adjustments in velocity, due to other vehicles deceleration, without the driver interaction. A more advanced ACC is cooperative adaptive cruise control (CACC). CACC does the same thing as ACC, but it also makes use of vehicle-to-vehicle (V2V) communication, which allows to take into consideration more vehicles than just the one in front.

One way, to increase and stimulate the research in the areas of cooperative and autonomous driving, is by having competitions. The first and oldest major competition was defense advanced research projects agency (DARPA) grand and urban challenges that were held in 2004, 2005 and 2007 [6]. Nowadays, a new competition is held, the grand cooperative driving challenge (GCDC), which had its first showcase in 2011 and it was done again in May 2016. The work being done in this thesis, is used to enter the GCDC 2016, which is described in more detail in Section 1.2.

This chapter starts by explaining some basic concepts, needed for understanding the work done, and by presenting the research concept vehicle (RCV). Next, a brief introduction to the GCDC 2016 is given. In Section 1.4 and 1.5 the problem formulation and the objective of the thesis is presented, respectively. Finally, related work is discussed.

1.1 Background

Autonomous vehicles will allow platoons to be formed, which will help, for example, prevent accidents and increase vehicle efficiency. A platoon is a group of vehicles that can travel safely with small inter-vehicle distances at high speeds. In every platoon, there is a lead vehicle and all following vehicles respond to the lead's movement. Platoons are only a possibility, because computers take over the control of the vehicle, which takes the human error out of the equation. Platooning improves the throughput of the roads, provides a more steady state flow of traffic, improves the aerodynamic effectiveness and performance of the vehicles and reduces road accidents, which, in turn, leads to reduced fuel consumption and less pollution.

Platooning is only possible if the vehicles communicate between each other and between the road infrastructures. So, it is important to introduce two concepts: vehicle-to-vehicle (V2V) communication and vehicle to infrastructure (V2I) communication. V2V is a wireless network where vehicles send messages to other vehicles with information about what they are doing, future intentions and status of the vehicle, i.e., for example, data such as speed, position, direction, braking and intentions of travel. V2I is the same thing as V2V, but it happens between vehicles and infrastructures such as, for example, traffic lights that sends its status, i.e., if it is red, yellow or green, which allows the vehicle to act accordingly.

Besides communication, autonomous vehicles need to be controlled. Cruise control (CC) is a system that takes over the throttle control of the vehicle and controls the speed of the vehicle, maintaining a speed reference set by the driver. This system is installed in most vehicles and, in general, it allows for a more comfortable ride and reduces driver fatigue. As said before, adaptive cruise control (ACC) can be seen as an improvement to CC, this system not only follows the reference speed defined by the driver, but also maintains a certain distance and behavior pattern set by a front vehicle, i.e., ACC makes the vehicle decelerate when to close to a front vehicle and vice-versa. Finally, a more advanced ACC is cooperative adaptive cruise control (CACC). CACC does the same thing as ACC, but it also makes use of V2V communication, which allows to take into consideration more vehicles than just the one in front and, ultimately, it allows to form platoons.

There are many type of controllers that can be used to achieve a fully autonomous vehicle, but this thesis only focus on model predictive control (MPC). MPC relies on dynamic models to predict the future states of the system and optimizes the current control action, while keeping future time slots in account. This is achieved by solving an optimization problem, that takes into consideration a finite time horizon. The biggest advantages of MPC are its ability to take into consideration future states of the system, acting accordingly, and also its ability to cope with both input and state constraints. There are different ways to formulate MPC problems, one of which is linear time-varying MPC (LTV-MPC). LTV-MPC consists of the linearization of the MPC problem, which basically can be obtained by computing an error model with respect to a reference. Different formulations and how they are obtained are further explained in Chapter 3.

1.1.1 Research Concept Vehicle

The research concept vehicle (RCV) is shown in Figure 1.1.

The RCV is a 2-seat 4-wheel steer-by-wire electric vehicle. It has two batteries, one main battery that powers the wheel motors and the other battery that powers all the other systems in the vehicle.

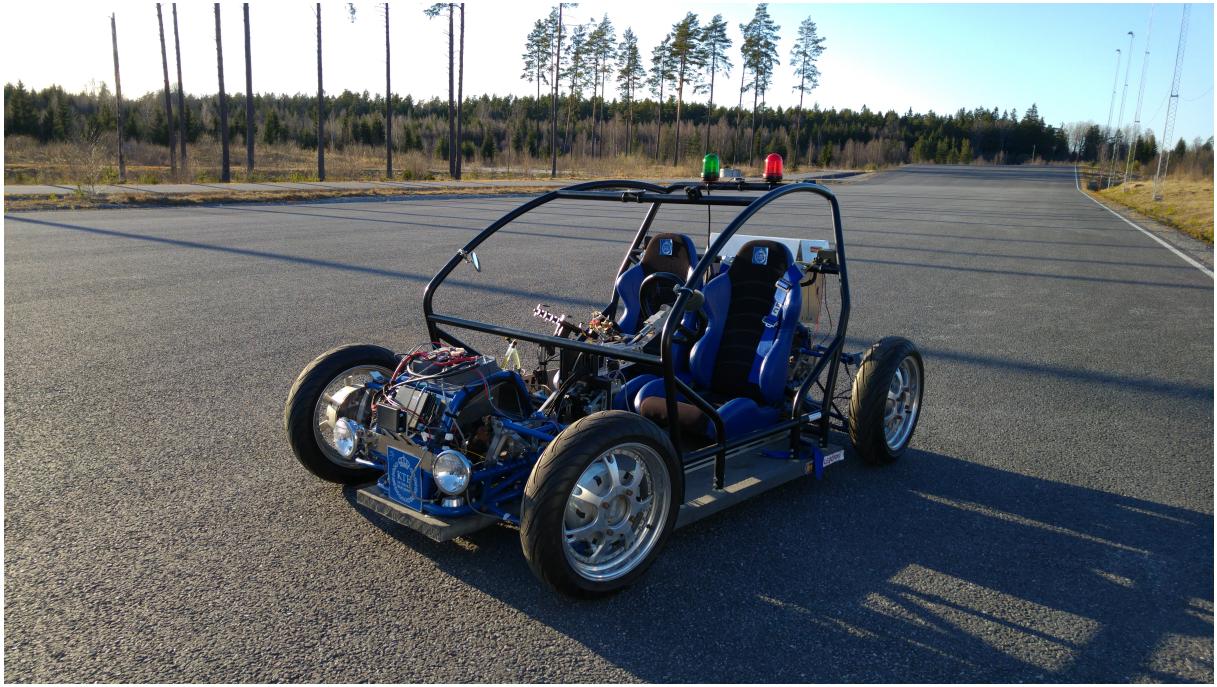


Figure 1.1: Research Concept Vehicle.

It is equipped with a variety of sensors, such as LIDARs, RADARs, cameras, V2V and V2I communication, Trimble GPS, IMU, among others.

Each wheel has a wheel hub motor with a stall torque at 150Nm and a rated maximum speed of 520rpm , giving a maximum vehicle speed of approximate $55\frac{\text{km}}{\text{h}}$, with a two person load depending on the tire radius and battery circuit power cap [17]. Each wheel is steered and camber actuated by linear actuators. The steering actuators can turn the wheels around 20 degrees inwards and 15 degrees outwards in reference to the car body, the camber actuators can tilt the wheels 10 degrees outwards and 15 degrees inwards.

The RCV is an over-actuated vehicle, allowing for more types of movements than a normal vehicle. One of the main challenges of this thesis is to use one of the extra degrees of freedom. The vehicle is able to steer each wheel independently, this allows for the vehicle to perform a type of motion called crabbing. This motion consists of the vehicle moving sideways, i.e., while the vehicle is facing a direction it moves to the side, because all the wheels are turned in the same way. Although the RCV allows for this type of motion, it still has limitations on the wheels turning angle, so, it cannot, for example, do parallel parking just by moving sideways.

Two actuators were added to the RCV allowing to activate the brakes by-wire, this was mounted in series after the normal brake pedal so that the driver can always brake in case it is needed. Besides braking the vehicle, when the driver presses the brakes it deactivates the longitudinal control. For the lateral control the deactivation happens when the driver actuates on the steering wheel. Both of these deactivations were set up for safety reasons.

The main component of the vehicle's electrical control system is the dSPACE 1401/1501 MicroAutoBox (MABX) computer. The MABX processes all signals from actuators, motors and driver inputs [17].

Most parts of the system are connected by two controller area network (CAN) buses, other components, such as gas pedal, horn, tie rod force sensors and the instrument board, are connected through ADC and bit I/O [17].

The controllers developed run on a mobile real-time target machine called Speedgoat. This machine is designed

to run simulink real-time, which makes it easy to program directly from MATLAB software. The Speedgoat is connected to the RADAR, V2V and V2I communication system, cameras and MABX. The most important information about how the system works, is that everything used by the controllers is processed inside the speedgoat, and can either come directly from a sensor to the Speedgoat or it comes through the MABX. The controllers outputs are sent to the MABX and, from there, either to the steering actuators or the motor controllers.

Now that some basic concepts and hardware used is described, the GCDC 2016 can be introduced.

1.2 Grand Cooperative Driving Challenge

The GCDC is a competition coordinated by the Netherlands organization for applied scientific research(TNO) with three other partners, Eindhoven university of technology (TU/e), the Spanish Applus+ IDIADA and Viktoria Swedish ICT. The aim of the competition is to accelerate and improve the real-life implementation and interoperability of wireless communication based automated driving [7]. It was held at the Brabant Brainport region, in the heart of the Dutch automotive cluster, at the automotive campus in Helmond, the Netherlands.

The competition is divided in two main scenarios and one optional scenario, which are explained next in detail.

1.2.1 Highway Scenario

The first scenario takes place in a highway and demonstrates advanced cooperative maneuvers, which is an evolution from the simple platooning scenarios performed in GCDC 2011 [1]. This scenario consists of two platoons going side by side on a highway. At some point in time the platoons receive a message warning them about a construction site up ahead on the road, meaning the platoons must merge before that. To do the merging, the vehicles use V2V communication to inform the other vehicles of their intentions.

Figure 1.2 provides a visual description of the scenario. Both platoons are approaching a construction site that is not visible at the start of the scenario, they receive information from the road side unit (RSU) that they are approaching the construction site and the message contains position and speed limit on the construction site. At this point in time, the two platoons must merge into the free lane and they must do it within a specified area called competition zone 1 (CZ-1). CZ-2 starts at an imaginary line at the beginning of the construction site and, in this zone, the vehicles should be traveling as a single platoon and within the speed limit.

The goal of the scenario is to increase throughput, but guarantee safety as well. So, the closer a vehicle follows the preceding one while maintaining the desired distance, the better. The desired distance can be calculated as,

$$d_{\text{des}} = h \cdot v + r_{\text{safe}}, \quad (1.1)$$

where h is the time headway, v is the vehicle velocity and r_{safe} is the standstill safety distance. Both h and r_{safe} were set by the organizers to 1s and 10m, respectively.

One of the judging criteria is the error,

$$e_i(k) = d_{\text{meas}_i}(k) - d_{\text{safe}_i}(k), \quad (1.2)$$

which corresponds to the error in the desired distance for the i^{th} vehicle for the k^{th} time instance. The score is obtained from this error and it depends on a scaling factor that determines a safety distance, e.g. $d_{\text{safe}_i}(k) = 0.7d_{\text{des}_i}(k)$.

The score function is described in Figure 1.3, where $T_h = (1 - s_{\text{safe}}) d_{\text{des}_i}(k)$, where s_{safe} is the scaling factor, and β represent the slope of the scoring function at a given distance error. Obviously, the more negative the error is, the faster the score goes to zero, because it is more dangerous to be close to the front vehicle than further way from it.

The other judging criteria are keeping the speed limit and merging, in time, inside the CZ-1.

This scenario presents some challenges for the control:

- CACC - the distance to the front vehicle and the behavior of the platoon have to be taken into consideration;
- Gap opening - when changing platoon if the vehicle is in the free lane it has to open a gap to the front vehicle so another vehicle can merge into that space;
- Merging - when changing platoon the vehicle has to change lane.

1.2.2 Intersection Scenario

The second scenario takes place at an intersection and, once again, aims at testing the cooperation between the vehicles.

Figure 1.4 provides a visual description of the scenario. The three competing yellow vehicles approaching the intersection are supposed to enter the CZ at the same time and are not allowed to respond to any communication from other vehicles before this moment. Once every vehicle is in the CZ, they cooperate to let vehicle V1 enter the main road in a safe manner. As soon as this vehicle enters the road, all the vehicles are suppose to go through the intersection as fast as possible. The scenario ends when all the vehicles are out of the CZ.

The evaluation criteria is, once again, keeping the speed limit and safety distances, but also the time it takes to perform the full scenario and the fact that no vehicle comes to a full stop during the scenario. The vehicles are supposed to perform the scenario in an optimal and efficient way, which means decelerate the minimum possible.

This scenario presents some challenges for the control:

- Approaching the CZ - the vehicle has to be at the CZ line at a specific speed at a specific point in time, which is done by an on-line planner and CC;
- CACC with virtual vehicle - when entering the CZ the distance to the front vehicle goes to 0m, but the ego vehicle should not slam the brakes, instead, it should decelerate smoothly, because it is just a virtual distance.

1.2.3 Emergency Scenario

The last scenario is optional and is not evaluated in the competition, but it serves as a demonstration of another possibility of this technology. This scenario takes place in a congested road when an emergency vehicle, for example, an ambulance, wants to pass and all the vehicles in front need to open a gap for it. Cooperative driving is perfect for this situation, because not only all the vehicles know much sooner of the emergency vehicle approaching, but also, they will be able to coordinate and open up the required gap in the fastest way possible.

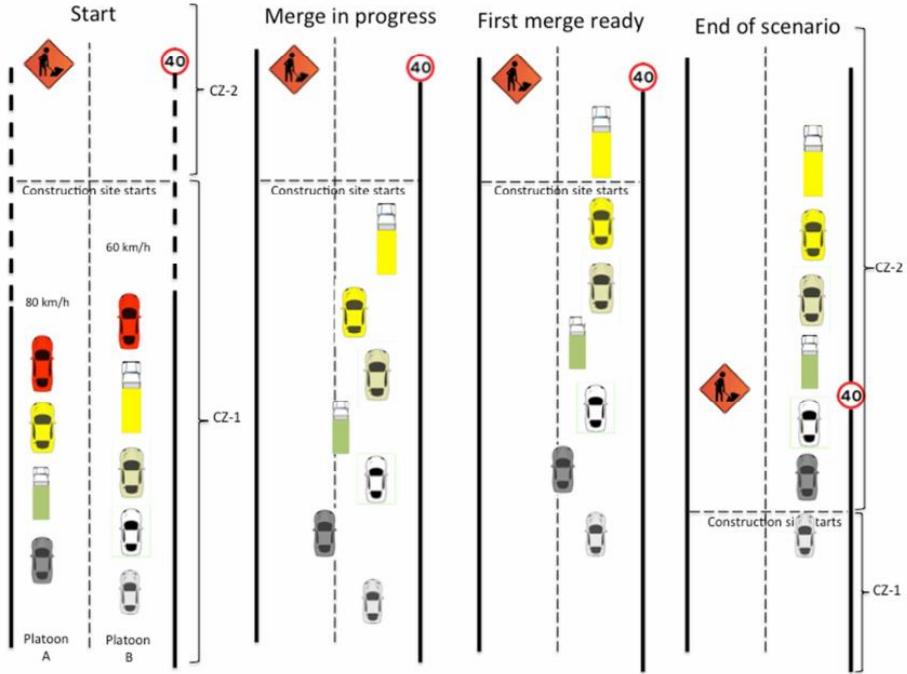


Figure 1.2: Highway scenario description [1]. Both platoons receive information from the RSU that they are approaching a construction site, that is not visible at the start of the scenario. Then, the platoons merge into the free lane and the final platoon travels through the construction site.

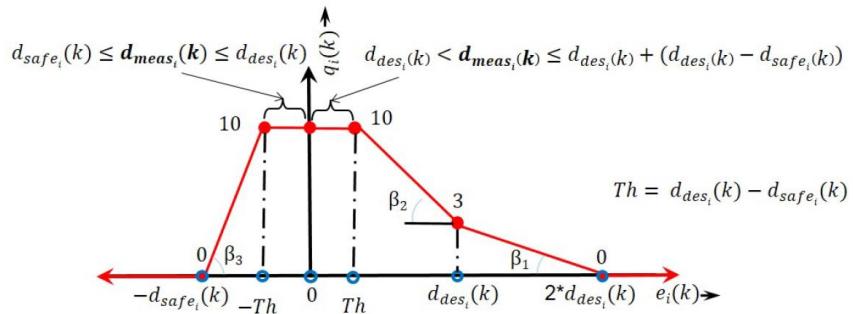


Figure 1.3: Scoring function q for desired distance criteria [2]. The score varies with the distance error and it is maximized when this error is at around zero, more specifically, in between $\pm T_h$.

Figure 1.5 provides a visual description of the scenario. The emergency vehicle approaches congested traffic and broadcasts its intent of passing through the traffic in a given position. The vehicles know at which time the emergency vehicle will pass and act accordingly to create the space for it to pass. It is important to notice that all the vehicles are supposed to slow down while the emergency vehicle passes. As soon as the vehicle has passed, the rest of vehicles resume previous speeds and normal positions.

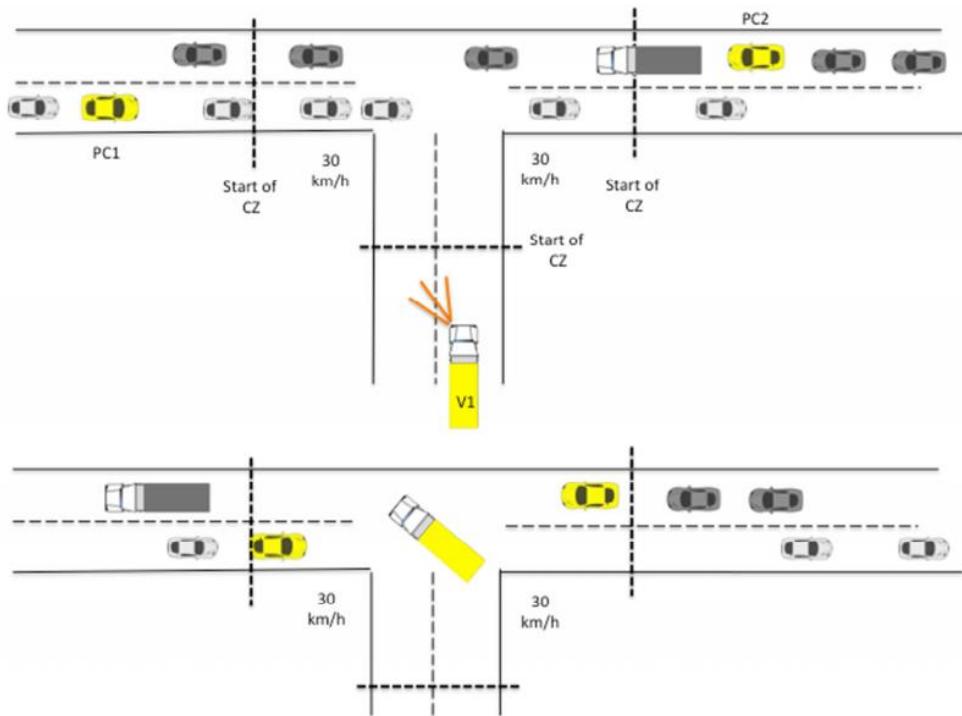


Figure 1.4: Cooperative intersection scenario description [1]. The three yellow vehicles approach the intersection and have to cross the CZ line at the same time, then, since vehicle V1 has priority, the other two vehicles have to slow down to let V1 cross first.

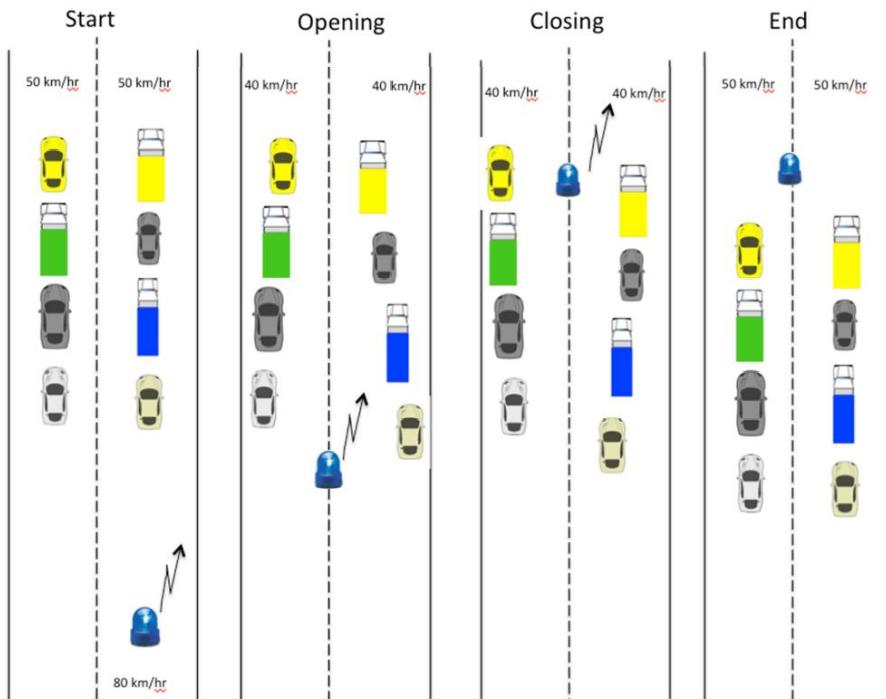


Figure 1.5: Emergency vehicle scenario description [1]. An emergency vehicle approaches two platoons broadcasting its state of emergency, the platoons are supposed to cooperate and make room so the emergency vehicle can go through.

1.3 Scope

The controllers designed are in between the supervisory layer and the low-level controllers that communicate directly with the different actuators. Figure 1.6 shows a scheme of the architecture of the RCV, the red box is the part in which this thesis focuses on.

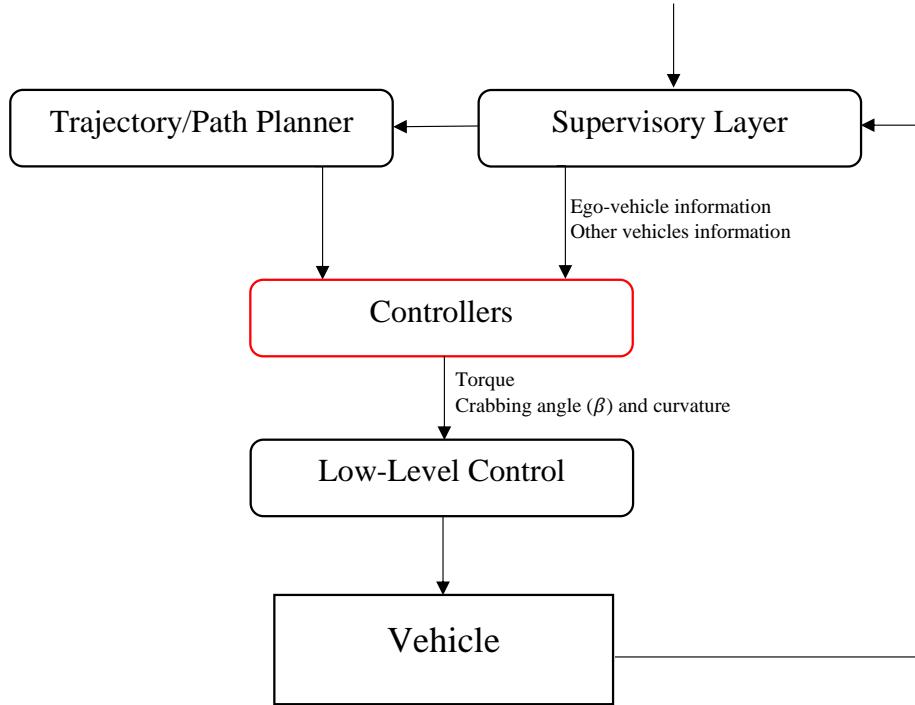


Figure 1.6: Scheme of the architecture of the RCV

The supervisory layer receives information from the vehicle, driver inputs, other vehicles and road infrastructures. It processes and passes that information to the planner and the implemented controllers, which also receive the planned path. All of this is implemented in the Speedgoat. The controllers output torque, crabbing angle and curvature requests, which are received by the MABX and acted upon by low-level controllers which in turn act directly on the motors or the steering actuators of the vehicle.

1.4 Problem Formulation

Being defined what is expected of the vehicle in the competition, it is possible to formulate the problem that is addressed in this thesis:

- The vehicle needs to have longitudinal control - CACC, ACC and CC;
- The vehicle needs to have lateral control - path following;
- The vehicle needs to keep safety distances, speed limits and acceleration limits;
- The control must be performed online in real-time.

1.5 Objective

The objective of this thesis is to design, implement and evaluate controllers for the RCV with the secondary objective of entering the GCDC 2016. It is important to notice that by entering the competition the controllers are optimized to score as much as possible according to the evaluation criteria. However, the controllers should still be general enough so they can be used for other purposes, in which the vehicle can drive in unstructured environments, fundamentally different from highway driving, with much sharper and more complex maneuvers.

1.6 Related Work/State of Art

The automotive industry is fully focused in automated vehicles and, some vehicles, have already implemented advanced driver assistance systems (ADAS). ADAS are systems developed for safety and better driving. One example of such a system is ACC.

To achieve stable platoons it is essential that the following control systems are string stable, if it aims to follow at short distances. In [8], J. Ploeg et al., showed theoretically and experimentally that it is possible to create a CACC that allows for time gaps smaller than 1s while maintaining string stability. On their tests they proved that a test fleet of six vehicles showed string stable behavior with a time headway of 0.7s. This time was determined by the latency of the communication system and the controller parameters used.

In [9], Achour Amazouz compares three different control approaches to solve the longitudinal control problem of platooning. Although model predictive control performed better than the other two approaches, linear-quadratic regulator (LQR) and proportional integral derivative (PID) controllers, MPC was too computationally heavy for the hardware used and, because of that, it was not a feasible solution. LQR is a feedback controller that optimizes a cost function at the current time. LQR performed better than PID with minimal increase in computational cost.

In [10], P. Falcone et al. prove that MPC can be used for lateral control. The MPC controller is effective up to $21 \frac{m}{s}$ on icy roads to perform a double lane change. Other possible example is [11], where I. Maurovic et al. use MPC to do trajectory tracking with a differential drive mobile robot. Another example that MPC is a good approach is [12], where D. Gu and H. Hu use MPC to make a robot follow a virtual leader.

MPC is going to be the approach considered in this thesis. It has many advantages over other possible solutions:

- Straightforward formulation, based on well understood concepts;
- Explicit handling of constraints;
- Explicit use of a model;
- Well understood and defined tuning parameters;
- Ability to predict future behavior of the system and take it into consideration.

The major disadvantage of MPC is the computation time. A big horizon and a complex model make it hard for the solvers. One way to deal with this problem is to use a linear time-varying MPC (LTV-MPC) approach, where the model of the system is linearized, which makes the problem less complex and easier to solve. Although LTV-MPC is less computationally heavy than nonlinear MPC, it is still more time consuming than other approaches, something that is kept in mind when defining the problem and parameters.

One of the objectives of this thesis is to be able to perform CACC, so, it is important to consider what has been done in this area. In [13], Fanping Bu et al. prove that it is possible to implement CACC on a regular vehicle using MPC. The results improved the commercial application of ACC of the vehicle used, which had a minimum headway time of $1.1s$, to a minimum headway time of $0.6s$. This reinforces the idea that the MPC approach is suitable to tackle the problem in hands.

In [14], Turri et al. demonstrate clearly that MPC is suitable for this problem. A linear MPC formulation is used to address the lane keeping and obstacle avoidance problems for a passenger vehicle driving on low curvature roads. The MPC controller is able to avoid obstacles in the middle of the road with an entry velocity of $50 \frac{km}{h}$ and return back to the center of the road effectively. One of the interesting things they do is to separate the longitudinal and the lateral control into different optimization problems, which is a solution explored in this thesis.

In [15], Künhe et al. explain how to convert the nonlinear problem into the linear one on their work of trajectory tracking with a mobile robot, where they also offer some insights of the differences between both approaches. The great advantage of LTV-MPC comparing with nonlinear MPC is that the linearized version is faster to solve than the other, and given that the problem needs to be solved on-line this is important. Also, nonlinear MPC is not guaranteed to get a solution, whilst LTV-MPC will always get a solution.

Considering that MPC is fully based on a model, it is also important to consider vehicle models used in previous works. The works referred use different types of vehicle models, some only use a kinematic model and others use, more complex, force models. This proves that, to use MPC, one does not need to have a complex vehicle model. The model used in [10], by P. Falcone et al., is a simplified bicycle model, where the vertical load of the vehicle, the longitudinal and lateral tire forces are taken into consideration. Furthermore, the aerodynamic forces are neglected, because, considering they wanted to do double lane changes on slippery surfaces, they were more concerned with the lateral dynamics of the vehicle.

In [16], Keviczky et al. successfully designed an MPC controller to stabilize a vehicle along a desired path while rejecting wind gusts and, on another scenario, perform double lane changes. It was able to stabilize the vehicle up to side wind speeds of $10.1 \frac{m}{s}$ on the wind rejection scenario and $17 \frac{m}{s}$ of entry speed in the double lane change maneuvers. However, the feedback control policy was computationally complex and not all the simulations could be experimentally validated, because of this complexity and the time it takes to solve it. They used a similar vehicle model to the one P. Falcone et al. used in [10], but added side wind forces. It is easy from these vehicle models, which are more concerned with the lateral dynamics, to incorporate the vehicle aerodynamics, which will mostly influence the longitudinal control.

Considering that the work done in this thesis is going to be implemented in the RCV, it is important to consider what has been done with it, its limitations and constraints. In [17], Petter Tomner uses a bicycle model to describe the dynamics of the vehicle, which proved to be good in the experimental results. The work consisted of logging and measuring data to evaluate and validate the different components of the RCV, some of the tests he performed to achieve the desired results were a circle test, step steer, straight line with full torque to estimate acceleration and torque characteristics, roll out test to measure the retardation forces acting on the RCV, among others.

Finally, it is important to know what has been done in the previous GCDC as a baseline of the minimum result expected. Some of the teams participating in the previous challenge used an MPC approach to solve the platooning problem. Team AnnieWAY [18] minimized the error of the safety distance, velocity difference between vehicles

and also the acceleration input over an horizon using MPC. It is important to notice that they, to platoon, were only following one vehicle at any given time. In other words, the MPC was designed to follow the "worst" vehicle in front and minimize everything in relation to that one, if every vehicle was performing perfectly their controller would follow the platoon leader.

1.7 Outline

In Chapter 2, the vehicle model used and the wheel angle transformations are presented. In Chapter 3, the control of the vehicle is addressed, first a background to MPC is presented, after, the specific MPC approach is described and, finally, the reference calculation is explained. In Chapter 4, the vehicle implementation is presented and numerical values are attributed to the control parameters. In Chapter 5, the longitudinal controller tests are presented and the lateral capabilities are demonstrated with simulations. In Chapter 6, the conclusions are presented. Finally, in Chapter 7, possible future work and improvements to the thesis and controllers are discussed.

Chapter 2

Modeling

One of the most important aspects of MPC is the model used to predict the future states of the system. The more complex the model used, the closer it represents reality, but also the more computationally heavy it is to predict the states and solve the optimization problem. So, a good model is one that represents the most important dynamics of the system in order to have insignificant static errors, but, at the same time, is not very complex in order to run in real-time.

In this thesis, a dynamic bicycle model is used, which is further described in Section 2.1. This model is used not only to predict the vehicle motion, but also to simulate the vehicle. After, the wheel angles transformations are presented in Section 2.2.

2.1 Vehicle Model

The bicycle model is commonly used to describe a vehicle since it simplifies a four-wheeled vehicle to a two-wheeled one and it still models the longitudinal, lateral and yaw motion well. To achieve this, each pair of two wheels is simplified to one wheel at the center of the axle. The wheel angles obtained from this model are recalculated before being sent to the steering actuators to account for all four wheels and their real position. This way, the desired curvature and crabbing motion are not affected by the differences between the model and the vehicle, this is further explained in Section 2.2.

The vehicle model used is based on [10, 16, 17].

A constant normal tire load is assumed when modeling the dynamics of the vehicle, i.e., $F_{z_f} = F_{z_r} = \text{constant}$, where $(.)_f$ and $(.)_r$ mean front and rear wheel, respectively. Figure 2.1 depicts an illustration of the vehicle model.

It is possible to describe the longitudinal, lateral and turning degrees of freedom with,

$$M\ddot{x} = M\dot{y}\dot{\psi} + 2F_{x_f} + 2F_{x_r}, \quad (2.1a)$$

$$M\ddot{y} = -M\dot{x}\dot{\psi} + 2F_{y_f} + 2F_{y_r}, \quad (2.1b)$$

$$I\ddot{\psi} = 2aF_{y_f} - 2bF_{y_r}, \quad (2.1c)$$

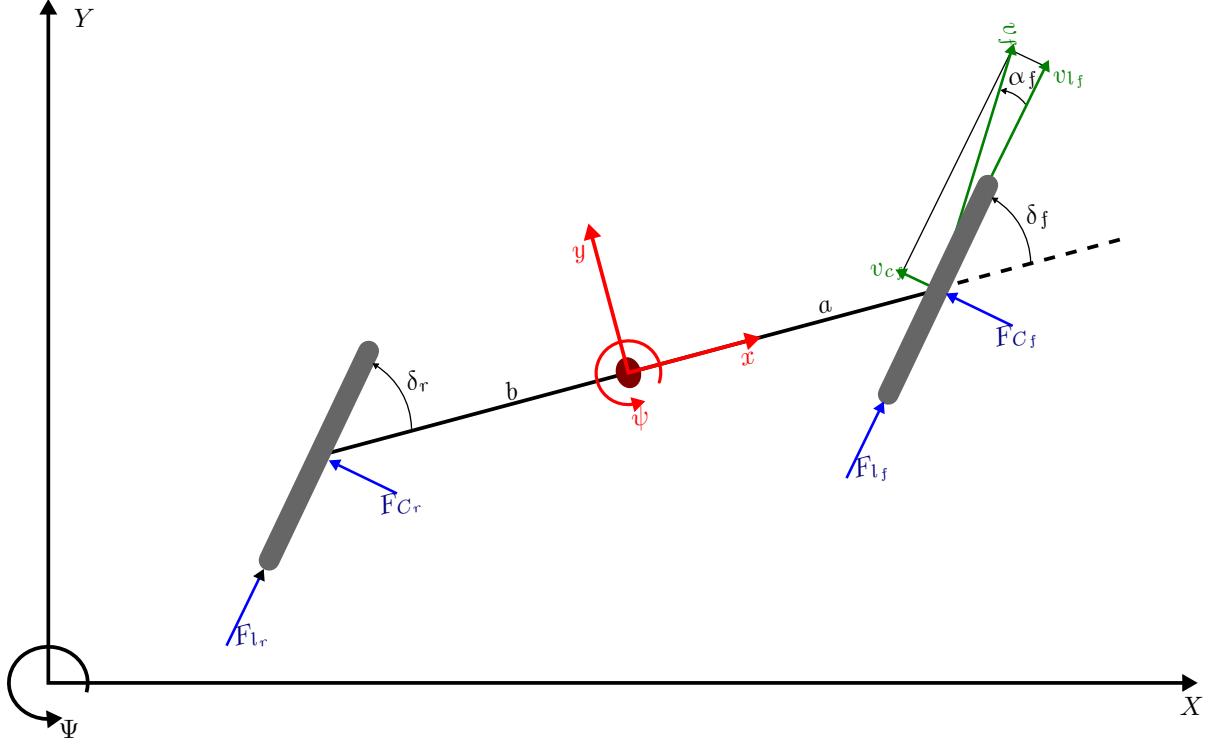


Figure 2.1: Bicycle Model, where, X, Y and Ψ are world coordinates, x, y and ψ are vehicle local coordinates, a and b are the distances from the CoG to front and rear wheel, respectively, F_C and F_l are the cornering and longitudinal forces, respectively, acting on the wheels, α is the slip angle and, finally, δ is the angle of the wheel.

where, x and y are the vehicle position in local coordinates, ψ is the vehicle yaw in relation to the world frame, $M = m + m_I$ is the total mass, where, m is the vehicle mass, m_I is the equivalent mass of rotating parts due to inertia, I the vehicle inertia around the z-axis, a and b are the distances from the front and rear wheels to the CoG, respectively, and F_x and F_y are the longitudinal and lateral forces acting on CoG of the vehicle. The vehicle's equations of motion in the world frame are

$$\dot{X} = \dot{x}\cos(\psi) - \dot{y}\sin(\psi), \quad (2.2a)$$

$$\dot{Y} = \dot{x}\sin(\psi) + \dot{y}\cos(\psi). \quad (2.2b)$$

The forces acting on the CoG can be described by

$$F_{x_f} = F_{l_f} \cos(\delta_f) - F_{C_f} \sin(\delta_f), \quad (2.3a)$$

$$F_{y_f} = F_{l_f} \sin(\delta_f) - F_{C_f} \cos(\delta_f), \quad (2.3b)$$

where, F_l and F_C represent the longitudinal and lateral force of the tire and δ is the angle of the wheel. Equivalent equations can be written for the rear wheel, which leads to

$$F_x = F_{x_f} + F_{x_r}, \quad (2.4a)$$

$$F_y = F_{y_f} + F_{y_r}, \quad (2.4b)$$

The longitudinal and lateral force of the tire can be expressed as

$$F_l = f(\alpha, \mu, s, F_z), \quad (2.5a)$$

$$F_C = f(\alpha, \mu, s, F_z), \quad (2.5b)$$

where, α is the angle between the direction of the wheel and the direction of its velocity, as illustrated in Figure 2.1, called the slip angle, μ is the friction coefficient for the road, s is the difference between the wheel ground point velocity and the equivalent rotational velocity called the slip ratio, and F_z is the vertical load acting on the wheels of the vehicle.

According to Rajamani [19], the lateral tire forces for small slip angles are proportional to α , which means

$$F_{C_f} = C_f \alpha_f, \quad (2.6a)$$

$$F_{C_r} = C_r \alpha_r, \quad (2.6b)$$

where, C_f and C_r are the cornering stiffness for the front and rear wheel, respectively. The slip angles can be calculated as

$$\alpha_f = \delta_f - \arctan\left(\frac{\dot{y} + a\dot{\psi}}{\dot{x}}\right), \quad (2.7a)$$

$$\alpha_r = \delta_r - \arctan\left(\frac{\dot{y} - b\dot{\psi}}{\dot{x}}\right). \quad (2.7b)$$

Assuming that the wheels ground point velocity equals their equivalent rotational velocity, the longitudinal slip ratio equals to zero and the longitudinal tire forces can be calculated as

$$F_{l_f} = \frac{T_f}{r}, \quad (2.8a)$$

$$F_{l_r} = \frac{T_r}{r}, \quad (2.8b)$$

where, r is the wheel radius. In this thesis torque vectoring is not considered, so $T_f = T_r$. The wheels forces are now defined but the vehicle is still influenced by external forces.

2.1.1 External Forces

Two main external forces are considered, namely air drag (F_a) and rolling resistance (F_r).

According to Rajamani [19], the aerodynamic drag force on a vehicle can be represented as

$$F_a = \frac{1}{2} \rho C_d A_F (\dot{x} + v_{\text{wind}})^2, \quad (2.9)$$

where, ρ is the mass density of air, C_d is the aerodynamic drag coefficient, A_F is the frontal area of the vehicle, which is the projected area of the vehicle in the direction of travel, and v_{wind} is the wind velocity in the longitudinal

direction (positive for headwind and negative for tailwind). The air drag force taken into consideration only influences the longitudinal velocity of the vehicle. Side winds are considered as disturbances and the air drag force caused by the lateral velocity of the car is not taken into consideration, because the lateral velocity is always small. Besides, v_{wind} is assumed to be zero which simplifies (2.9) to

$$F_a = \frac{1}{2} \rho C_d A_f \dot{x}^2, \quad (2.10)$$

The rolling resistance force can be modeled by

$$F_r = f_{\text{roll}} mg \cos(\theta), \quad (2.11)$$

where, f_{roll} is the roll resistance coefficient, g is the gravitational acceleration and θ is the angle of inclination of the road. Assuming that the vehicle is used in flat roads the previous equation can be simplified to

$$F_r = f_{\text{roll}} mg. \quad (2.12)$$

Finally, the states of the vehicle can be described as

$$\dot{X} = \dot{x} \cos(\psi) - \dot{y} \sin(\psi), \quad (2.13a)$$

$$\dot{Y} = \dot{x} \sin(\psi) + \dot{y} \cos(\psi), \quad (2.13b)$$

$$M \ddot{x} = M \dot{y} \dot{\psi} + 2F_{x_f} + 2F_{x_r} - F_r - F_a, \quad (2.13c)$$

$$M \ddot{y} = -M \dot{x} \dot{\psi} + 2F_{y_f} + 2F_{y_r}, \quad (2.13d)$$

$$I \ddot{\psi} = 2aF_{y_f} - 2bF_{y_r}, \quad (2.13e)$$

which can also be described by the discrete equations

$$X_{k+1} = X_k + \Delta T (\dot{x}_k \cos(\Psi_k) - \dot{y}_k \sin(\Psi_k)), \quad (2.14a)$$

$$Y_{k+1} = Y_k + \Delta T (\dot{x}_k \sin(\Psi_k) + \dot{y}_k \cos(\Psi_k)), \quad (2.14b)$$

$$\Psi_{k+1} = \Psi_k + \Delta T \dot{\psi}_k, \quad (2.14c)$$

$$\dot{x}_{k+1} = \dot{x}_k + \Delta T \left(\frac{2F_{x_{f_k}} + 2F_{x_{r_k}} - F_r - F_{a_k}}{M} + \dot{y}_k \dot{\psi}_k \right), \quad (2.14d)$$

$$\dot{y}_{k+1} = \dot{y}_k + \Delta T \left(\frac{2F_{y_{f_k}} + 2F_{y_{r_k}}}{M} - \dot{x}_k \dot{\psi}_k \right), \quad (2.14e)$$

$$\dot{\psi}_{k+1} = \dot{\psi}_k + \Delta T \frac{2aF_{y_{f_k}} - 2bF_{y_{r_k}}}{I}. \quad (2.14f)$$

The control variables are the steering angles of the front and rear wheel, δ_f and δ_r , respectively, and the torque of each wheel motor, T .

2.1.2 Model Constant Values

The constants used in the model are the ones used in [17]. Small changes are made to the values used by Peter Tomner, because since his work the vehicle was changed, e.g. the main battery was changed which affected the total weight of the vehicle. To make these changes small tests, with constant torque and steering inputs, are made and the model is slightly tuned to better fit the results obtained. The final values are presented in Table 2.1.

Abbreviation	Value	Unit
m	608	kg
m_I	40	kg
g	9.81	$\frac{m}{s^2}$
a	1.0921	m
b	0.9079	m
I	1000	kgm^2
C_f	25668.509	$\frac{N}{rad}$
C_r	25668.509	$\frac{N}{rad}$
r	0.3	m
f	0.0027	-
ρ	1.23	$\frac{kg}{m^3}$
C_d	1.0834	-
A_F	2.25	m^2

Table 2.1: Bicycle model constant values.

2.2 Wheel Angles Transformation

Considering that the real vehicle has four wheels instead of two, it is important to specify and describe how the steering angles of the front and rear wheels of the bicycle model convert to $\delta_{f_{left}}, \delta_{f_{right}}, \delta_{r_{left}}$ and $\delta_{r_{right}}$ respectively the angle of the front left, front right, rear left and rear right wheels. Note that this transformation is done by low-level controllers, but it is important to prove that controllability is not lost by not considering a four-wheeled model in the MPC.

Figure 2.2 depicts the geometric relation between the different angles of a four-wheeled vehicle. The model is based on the Ackerman steering geometry. However, the considered vehicle allows for the center of rotation to be placed anywhere in the 2D-plane, i.e., the vehicle can move sideways and curve at the same time.

After defining a crabbing angle β and the angle γ for each wheel, the δ for each wheel can be calculated.

From Figure 2.3 it is possible to obtain the following relations for $\gamma_{f_{left}}$, in a similar fashion it is possible to obtain for the other angles

$$\gamma_{f_{left}} = \text{atan} \left(\frac{d_{f_{left}} \cos(\lambda_{f_{left}} - \beta)}{R - d_{f_{left}} \sin(\lambda_{f_{left}} - \beta)} \right), \quad (2.15a)$$

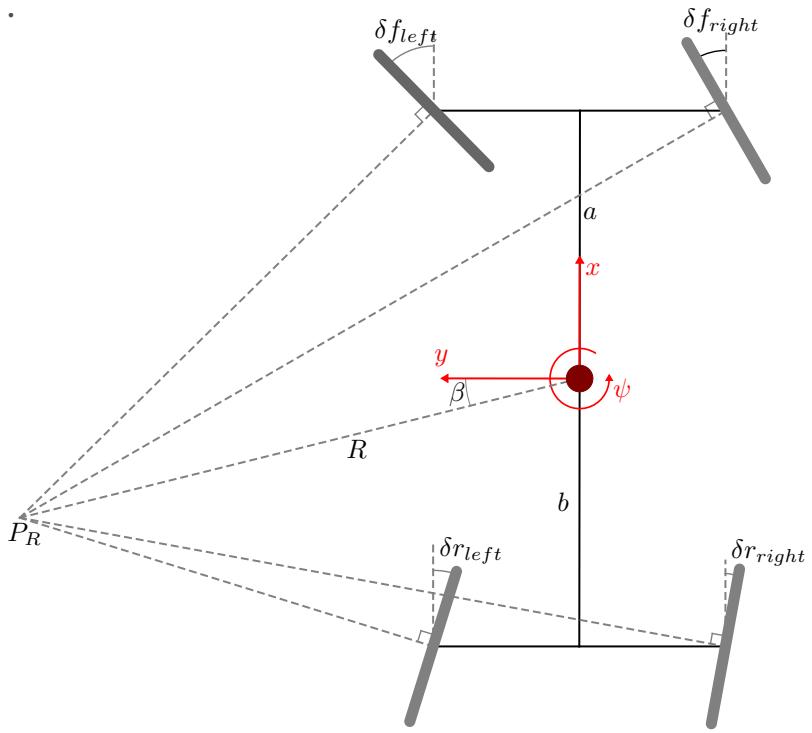


Figure 2.2: Four wheels direction from point of rotation, where, β is the crabbing angle, P_R is the point of rotation and R is the radius of the rotation.

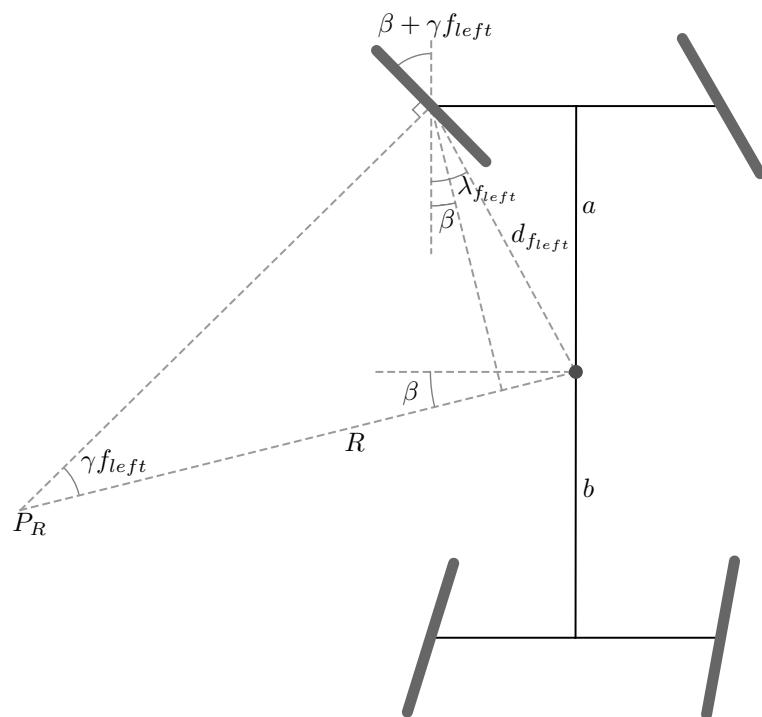


Figure 2.3: Geometric relations between $\delta_{f_{left}}$ and point of rotation.

$$\gamma_{f_{\text{right}}} = \text{atan} \left(\frac{d_{f_{\text{right}}} \cos(\lambda_{f_{\text{right}}} + \beta)}{R + d_{f_{\text{right}}} \sin(\lambda_{f_{\text{right}}} + \beta)} \right), \quad (2.15b)$$

$$\gamma_{r_{\text{left}}} = \text{atan} \left(\frac{d_{r_{\text{left}}} \cos(\lambda_{r_{\text{left}}} + \beta)}{R - d_{r_{\text{left}}} \sin(\lambda_{r_{\text{left}}} + \beta)} \right), \quad (2.15c)$$

$$\gamma_{r_{\text{right}}} = \text{atan} \left(\frac{d_{r_{\text{right}}} \cos(\lambda_{r_{\text{right}}} - \beta)}{R + d_{r_{\text{right}}} \sin(\lambda_{r_{\text{right}}} - \beta)} \right), \quad (2.15d)$$

and for δ

$$\delta_{f_{\text{left}}} = \beta + \gamma_{f_{\text{left}}}, \quad (2.16a)$$

$$\delta_{f_{\text{right}}} = \beta + \gamma_{f_{\text{right}}}, \quad (2.16b)$$

$$\delta_{r_{\text{left}}} = \beta - \gamma_{r_{\text{left}}}, \quad (2.16c)$$

$$\delta_{r_{\text{right}}} = \beta - \gamma_{r_{\text{right}}}. \quad (2.16d)$$

Considering that the controller does not output directly the curvature and the β angle, it is important to define how these are going to be obtained from δ_f and δ_r , the outputs of the controller. In Figure 2.4 it is possible to visualize this geometric relation.

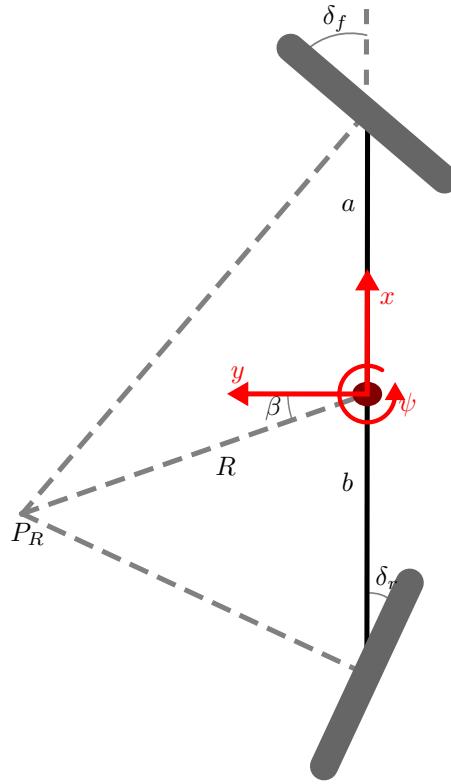


Figure 2.4: Point of rotation from wheels direction.

The intersection point of the lines perpendicular to the wheels direction corresponds to the vehicle center of rotation. By calculating the distance of this point to the CoG one obtains the radius of rotation

$$R = \sqrt{P_R(1)^2 + P_R(2)^2}, \quad (2.17)$$

with the radius of rotation it is possible to compute the curvature of the vehicle,

$$k = \frac{1}{R}. \quad (2.18)$$

Finally, the crabbing angle can be calculated from the point of rotation,

$$\beta = \begin{cases} -\left| \tan^{-1} \left(\frac{P_{R_x}}{P_{R_y}} \right) \right| & \text{if } P_R \in 1^{st} \text{ and } 3^{rd} \text{ quadrant} \\ \left| \tan^{-1} \left(\frac{P_{R_x}}{P_{R_y}} \right) \right| & \text{if } P_R \in 2^{nd} \text{ and } 4^{th} \text{ quadrant} \end{cases}. \quad (2.19)$$

Chapter 3

Control Design

The objective of this thesis is to develop controllers that allow the research concept vehicle to become fully autonomous. The controllers are designed within the model predictive control (MPC) framework. This choice is made based on the research done where many advantages were found:

- Straightforward formulation, based on well understood concepts;
- Explicit handling of constraints;
- Explicit use of a model;
- Well understood and defined tuning parameters;
- Ability to predict future behavior of the system and take it into consideration.

It is important to constrain the control signals, because the vehicle is not only constrained mechanically, but also externally. Besides the constraints that the motors, battery, steering actuators, etc. have, the vehicle might also be constrained, for example, to a maximum velocity. This is a state constrain and it is easy to deal with using MPC.

The outline of this chapter is as follows. It starts by explaining the basics of MPC, then it explains a derivation of the normal MPC, which is linear time-varying MPC (LTV-MPC), after it explains how LTV-MPC is used to do reference tracking, then formulates the problem as a quadratic programming problem. After, it explains the control design for the RCV and, finally, the reference generation is presented.

3.1 Model Predictive Control

The MPC algorithm is based on the prediction of future states of a dynamic system by utilizing its mathematical model. Based on these predictions, the MPC finds the control sequence that minimizes a cost function, while taking into consideration the constraints on control signals and system states.

The formulation of the MPC problem may differ regarding model, cost function, constraints and horizon used but the basic formulation is as follows

$$\begin{aligned}
& \underset{u}{\text{minimize}} && \sum_{i=1}^H J(z_i, u_i) \\
& \text{subject to} && z_{i+1} = f(z_i, u_i), \\
& && z_i \in Z, \\
& && u_i \in U,
\end{aligned} \tag{3.1}$$

where, the notation a_i expresses the predicted value for variable a at time instant i , $J(z_i, u_i)$ is the cost function for the optimization, Z and U are constraint sets for the system states and control inputs, respectively, and f represents the prediction model. The most common choice for the cost function is quadratic in order to reduce the complexity of the problem. The variable u expresses the control signal, H expresses the prediction horizon and z is the system state vector.

The prediction in (3.1) is made over H , this determines the number of future system states and control signals to take into consideration. A common way of reducing the complexity while keeping a big horizon is to use two different horizons, one prediction horizon, H_p and one control horizon, H_c . The control horizon is defined shorter than the prediction horizon which significantly reduces the complexity of the problem while maintaining a long state prediction horizon. In order to do this, the last control signal is maintained constant throughout the states that do not have a time-matching control signal, i.e., the control signals $u_{H_c+1} \dots u_{H_p} = u_{H_c}$ [20].

Solving (3.1) results in H number of control signals. However, only the first control signal is used, i.e., only the control signal for the current time instant is used as an input to the system. On the next time instant, the optimization problem is solved again and a new input is calculated for the system.

The cost function J determines what the control is going to achieve, considering the generic case of controlling the system to the origin, a common cost function is

$$J = z_H^T Q_f z_H + \sum_{i=1}^{H-1} (z_i^T Q z_i + u_i^T R u_i), \tag{3.2}$$

where, Q_f is a positive-semidefinite terminal state weight matrix, Q is a positive-semidefinite state weight matrix and R is a positive-definite control weight matrix.

This formulation of MPC, depending on the model used, which in this case is nonlinear, can also be called nonlinear MPC (NMPC) and requires a nonlinear programming problem to be solved on-line at each time step to determine the control inputs. The biggest challenge is to obtain an on-line solution of the nonlinear problem, because it is generally non-convex [21].

3.2 Linear MPC

In order to reduce the computational effort, one needs to reduce the complexity. One way of achieving this, but still keeping the good results of NMPC, is by using a linear MPC formulation [10, 15]. This consists of reducing the complexity by choosing linear constraints and prediction model, additionally to the quadratic cost function mentioned in Section 3.1.

This can be formulated as

$$\begin{aligned}
\underset{u}{\text{minimize}} \quad & z_H^T Q_f z_H + \sum_{i=1}^{H-1} (z_i^T Q z_i + u_i^T R u_i) \\
\text{subject to} \quad & z_{i+1} = A z_i + B u_i, \\
& C z_i + D u_i \leq b,
\end{aligned} \tag{3.3}$$

where, $z_{i+1} = A z_i + B u_i$ is a linear formulation of the system model and $C z_i + D u_i \leq b$ is the linear formulation of the constraints.

As shown in Chapter 2, the model used is nonlinear, so, to be able to use this MPC formulation, it has to be linearized. The linearization of function $f(z, u)$ at the point $(z_{\text{ref}}, u_{\text{ref}})$ is:

$$f(z, u) \approx f(z_{\text{ref}}, u_{\text{ref}}) + \frac{\partial f(z, u)}{\partial z} \bigg|_{z_{\text{ref}}, u_{\text{ref}}} (z - z_{\text{ref}}) + \frac{\partial f(z, u)}{\partial u} \bigg|_{z_{\text{ref}}, u_{\text{ref}}} (u - u_{\text{ref}}) \tag{3.4}$$

Until this point the MPC problem formulation assumes that the objective is to take the system to the origin, i.e., $(z_{\text{ref}} = u_{\text{ref}}) = 0$ so, it is now important to demonstrate how to change the formulation to allow the system to track any state, different than the origin.

3.3 MPC for Reference Tracking

In both (3.1) and (3.3) the minimization is directly made over the states and control inputs, which, in the perfect case, makes not only the states but also the control inputs go to zero. However, the objective is to track a reference that might be different than zero. To do this, the state variables and the control inputs are replaced by two new variables, $\tilde{z} = z - z_{\text{ref}}$ and $\tilde{u} = u - u_{\text{ref}}$. This results in

$$\begin{aligned}
\underset{\tilde{u}}{\text{minimize}} \quad & \tilde{z}_H^T Q_f \tilde{z}_H + \sum_{i=1}^{H-1} (\tilde{z}_i^T Q \tilde{z}_i + \tilde{u}_i^T R \tilde{u}_i) \\
\text{subject to} \quad & \tilde{z}_{i+1} = A \tilde{z}_i + B \tilde{u}_i, \\
& \tilde{z}_i = z_i - z_{\text{ref}}, \\
& \tilde{u}_i = u_i - u_{\text{ref}}, \\
& C \tilde{z}_i + D \tilde{u}_i \leq b,
\end{aligned} \tag{3.5}$$

It is clear that the optimization function is minimized with respect to \tilde{u} meaning that the optimal control input is $u^* = \tilde{u} + u_{\text{ref}}$.

In (3.5) it was also introduced another concept that is time varying references, i.e., z_{ref} , this formulation is the so called linear time-varying MPC and allows for every prediction made to track a different reference value and not only one constant value for all predictions. The reference generation is very important for the optimal control and this is further addressed later in this chapter.

3.4 Quadratic Programming Formulation

Even with the linearization, solving the problem on-line is still an issue. So, to minimize more the computational time, the problem is posed as a quadratic programming (QP) problem, because the solver used is optimized to solve this type of formulation. The general QP problem has the form [15],

$$\begin{aligned}
& \underset{u}{\text{minimize}} && \frac{1}{2}u^T H u + f^T u \\
& \text{subject to} && D_{ineq} u \leq b_{ineq}, \\
& && D_{eq} u = b_{eq}, \\
& && lb \leq u \leq ub.
\end{aligned} \tag{3.6}$$

It is clear that, to have the MPC problem (3.5) as QP, the summation term has to be removed. To do this, two vectors are introduced

$$\bar{z}_{i+1} = \begin{bmatrix} \tilde{z}_{i+1} \\ \tilde{z}_{i+2} \\ \tilde{z}_{i+3} \\ \vdots \\ \tilde{z}_{i+H} \end{bmatrix}, \tag{3.7}$$

$$\bar{u}_i = \begin{bmatrix} \tilde{u}_i \\ \tilde{u}_{i+1} \\ \tilde{u}_{i+2} \\ \vdots \\ \tilde{u}_{i+H-1} \end{bmatrix}. \tag{3.8}$$

Thus, the cost function in 3.5 can be rewritten as

$$J = \bar{z}_{i+1}^T \bar{Q} \bar{z}_{i+1} + \bar{u}_i^T \bar{R} \bar{u}_i, \tag{3.9}$$

where, $\bar{Q} = \text{diag}(Q, \dots, Q, Q_f)$ and $\bar{R} = \text{diag}(R, \dots, R)$. Therefore it is now possible to write the prediction model (3.5) as

$$\bar{z}_{i+1} = \bar{A}_i \bar{z}_i + \bar{B}_i \bar{u}_i, \tag{3.10}$$

where, \bar{A} and \bar{B} are

$$\bar{A}_i = \begin{bmatrix} A_i \\ A_{i+1} A_i \\ A_{i+2} A_{i+1} A_i \\ \vdots \\ A_{i+H-1} \dots A_{i+1} A_i \end{bmatrix}, \tag{3.11}$$

$$\bar{B}_i = \begin{bmatrix} B_i & 0 & 0 & \dots & 0 \\ A_{i+1}B_i & B_{i+1} & 0 & \dots & 0 \\ A_{i+2}A_{i+1}B_i & A_{i+2}B_{i+1} & B_{i+2} & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ A_{i+H-1} \dots A_{i+1}B_i & A_{i+H-1} \dots A_{i+2}B_{i+1} & A_{i+H-1} \dots A_{i+3}B_{i+2} & \dots & B_{i+H-1} \end{bmatrix}. \quad (3.12)$$

From (3.9) and (3.10) and after some algebraic manipulations it is possible to write (3.5) cost function as

$$\bar{J}_i = \frac{1}{2} \bar{u}_i^T H_i \bar{u}_i + f_i^T \bar{u}_i + d_i, \quad (3.13)$$

where

$$H_i = 2(\bar{B}_i^T \bar{Q} \bar{B}_i + \bar{R}), \quad (3.14a)$$

$$f_i = 2\bar{B}_i^T \bar{Q} \bar{A}_i \tilde{z}_i, \quad (3.14b)$$

$$d_i = \tilde{z}_i^T \bar{A}_i^T \bar{Q} \bar{A}_i \tilde{z}_i, \quad (3.14c)$$

where, the matrix H_i is a hessian matrix, that is always positive-definite. It describes the quadratic part of the objective function, and the vector f_i the linear part. The vector \tilde{z}_i represents the current error of the states, this is the error that the optimization problem compensates for. Considering d_i is independent of \tilde{u} it does not influence the optimization problem, so it is possible to discard this term and rewrite (3.13) as

$$\bar{J}'_i = \frac{1}{2} \bar{u}_i^T H_i \bar{u}_i + f_i^T \bar{u}_i. \quad (3.15)$$

Finally, after defining the basic formulations of MPC and how they work, it is possible to explain in detail how the problem was formulated for the specific vehicle and objectives.

3.5 Formulation

It is possible to divide the problem defined in Section 1.4 in longitudinal and lateral control. This division has some advantages, namely:

- Easier to tune the parameters and weight matrices. Each problem has different states and control inputs to be tuned;
- More robust. If one of the controllers fails the other is still able to function with certain limitations;
- Two less complex problems. The complexity grows quadratically with the number of states;
- Easier to test. The controllers can be tested independently.

But it also has some disadvantages, namely:

- Harder to design. It requires two problems to be defined;
- Harder to integrate. Both controllers need to communicate in between each other and match their inputs at every time step;
- Loss of optimality. Both controllers cannot calculate their outputs at the same time, so only the last one to calculate its control outputs is calculated based on the exact references.

The advantages outweigh the disadvantages, so the division is made.

3.5.1 Longitudinal Controller

The main objectives of this controller is to platoon, while keeping a distance to other vehicles and also to be able to follow a cruise velocity, a reference that can be set by the driver/passenger.

So, to accomplish these objectives there are three important variables:

- Distance to the front vehicle. The vehicle needs to always keep a safety distance to the front vehicle, so there is never a risk of collision;
- Velocity of the platoon. Allows to predict the current movement of the platoon;
- Acceleration of the platoon. Allows to predict the future movement of the platoon.

Considering that another objective of this thesis is to enter the GCDC 2016, it is also important to take into consideration the way the vehicle is evaluated. The score is calculated by the error (1.2) between the desired distance to the front vehicle (1.1) and the distance measured. In Figure 1.3, it is possible to verify that the maximum score is given when the error is around 0.

The states that are optimized in the longitudinal controller are

$$z_{\text{long}} = \begin{bmatrix} \Delta d \\ d \\ v \end{bmatrix}, \quad (3.16)$$

where, Δd , d and v are the error of the distance to the front vehicle in relation to the desired distance, the distance to the front vehicle and the ego velocity in the local x-axis of the vehicle, respectively. These states can be calculated by

$$\Delta d_i = d_i - (r_{\text{safe}} + hv_i), \quad (3.17a)$$

$$d_{i+1} = d_i + (v_{\text{platoon}_i} - v_i) \Delta T, \quad (3.17b)$$

$$v_{i+1} = v_i + \Delta T \left(\frac{2F_{x_{f_i}} + 2F_{x_{r_i}} - F_r - F_{a_i}}{M} + \dot{y}_i \dot{\psi}_i \right), \quad (3.17c)$$

where, ΔT is a time constant and v_{platoon_i} is the velocity of the platoon at instant i and can be expressed as

$$v_{\text{platoon}_{i+1}} = v_{\text{platoon}_i} + \Delta T a_{\text{platoon}_i}, \quad (3.18)$$

where, a_{platoon_i} is the acceleration of the platoon. It is, now, possible to define the optimization problem as

$$\begin{aligned}
\text{minimize}_{\tilde{u}_{\text{long}}, \Delta s} \quad & \tilde{z}_{\text{long}_H}^T Q_{\text{long}_f} \tilde{z}_{\text{long}_H} + \tilde{u}_{\text{long}_H}^T R_{\text{long}_f} \tilde{u}_{\text{long}_H} + q_{\Delta s} \Delta s + \sum_{i=1}^{H-1} (\tilde{z}_{\text{long}_i}^T Q_{\text{long}_i} \tilde{z}_{\text{long}_i} + \tilde{u}_{\text{long}_i}^T R_{\text{long}_i} \tilde{u}_{\text{long}_i}) \\
\text{subject to} \quad & \tilde{z}_{\text{long}_{i+1}} = A_{\text{long}_i} \tilde{z}_{\text{long}_i} + B_{\text{long}_i} \tilde{u}_{\text{long}_i}, \\
& \tilde{z}_{\text{long}_i} = z_{\text{long}_i} - z_{\text{long}_i}^{\text{ref}}, \\
& \tilde{u}_{\text{long}_i} = u_{\text{long}_i} - u_{\text{long}_i}^{\text{ref}}, \\
& D_{\text{long}_{\text{ineq}}} [\tilde{u}_{\text{long}_1}; \tilde{u}_{\text{long}_2}; \dots; \tilde{u}_{\text{long}_H}; \Delta s] \leq b_{\text{long}_{\text{ineq}}}, \\
& lb_{\text{long}_i} \leq \tilde{u}_{\text{long}_i} \leq ub_{\text{long}_i}, \\
& lb_{\text{long}_{H+1}} \leq \Delta s \leq ub_{\text{long}_{H+1}}, \\
& T_{\min_i} \leq \tilde{u}_{\text{long}_i} \leq T_{\max_i},
\end{aligned} \tag{3.19}$$

and in the QP form as

$$\begin{aligned}
\text{minimize}_{\bar{u}_{\text{long}}, \Delta s} \quad & [\bar{u}_{\text{long}}; \Delta s]^T H_{\text{long}} [\bar{u}_{\text{long}}; \Delta s] + f_{\text{long}}^T [\bar{u}_{\text{long}}; \Delta s] \\
\text{subject to} \quad & D_{\text{long}_{\text{ineq}}} [\bar{u}_{\text{long}}; \Delta s] \leq b_{\text{long}_{\text{ineq}}}, \\
& lb_{\text{long}} \leq [\bar{u}_{\text{long}}; \Delta s] \leq ub_{\text{long}}, \\
& T_{\min} \leq \bar{u}_{\text{long}} \leq T_{\max},
\end{aligned} \tag{3.20}$$

where Δs is a slack variable that is used to guarantee the feasibility of the problem, $q_{\Delta s}$ is the weight of the slack variable and u is the torque of each motor. Note that it is assumed, when formulating the model, that all the motors have the same applied torque.

States Linearization

To formulate the problem (3.20), it is necessary to calculate matrices H_{long} and f_{long} . To do this, it is necessary to calculate matrices A_{long} and B_{long} of the linearized state space error model

$$\tilde{z}_{\text{long}_{i+1}} = A_{\text{long}_i} \tilde{z}_{\text{long}_i} + B_{\text{long}_i} \tilde{u}_{\text{long}_i}, \tag{3.21}$$

which can be obtained by taking the partial derivatives at the reference points

$$A_{\text{long}_i} = \begin{bmatrix} 0 & 1 & & -h \\ 0 & 1 & & -\Delta T \\ 0 & 0 & 1 - \Delta T \frac{2C_f \frac{\dot{y}_i + \dot{\psi}_i a}{(\dot{y}_i + \dot{\psi}_i a)^2 + v_i^2} \sin(\delta_{f_i}) + 2C_r \frac{\dot{y}_i - \dot{\psi}_i b}{(\dot{y}_i - \dot{\psi}_i b)^2 + v_i^2} \sin(\delta_{r_i}) + v_i \rho C_d A_F}{M} & \end{bmatrix}, \tag{3.22}$$

$$B_{\text{long}_i} = \begin{bmatrix} 0 \\ 0 \\ \frac{2\Delta T(\cos(\delta_{f_i}) + \cos(\delta_{r_i}))}{rM} \end{bmatrix}. \quad (3.23)$$

The current error vector, \tilde{z}_i , depends on the operation mode of the controller. There are two modes, the CACC and the CC. For CACC, the error vector can be calculated as

$$\tilde{z}_{\text{long}_i} = [d_i - hv_i - r_{\text{safe}}; -(v_{\text{platoon}_i}h + r_{\text{safe}}) + d_i; -v_{\text{platoon}_i} + v_i], \quad (3.24)$$

the difference between the error calculation of the distance score and the distance, is that the distance is based on the velocity of the platoon which is or will be the steady state velocity and the score is based on the current velocity of the ego vehicle, because that is how the GCDC is judged. For CC, the only state that matters is velocity, so, the error vector only takes in consideration this state and can be expressed as

$$\tilde{z}_{\text{long}_i} = [0; 0; -v_{\text{platoon}_i} + v_i]. \quad (3.25)$$

Besides the error vector changing with the controller operation mode, the weighing matrices also change, so for CACC, $Q_{\text{long}_i} = \text{diag}(q_{\Delta d_i}, q_{d_i}, q_{v_i})$, and for CC, $Q_{\text{long}_i} = \text{diag}(0, 0, q_{v_i})$, once again, only the velocity state matters in this operation mode, so only this state is given weight. The weight for the control signal is the same for both operation modes, $R_{\text{long}_i} = q_{T_i}$.

Finally, by using (3.11), (3.12) and (3.14) it is possible to calculate H_{long} and f_{long} .

Constraints

The explicit specification of constraints in the optimization problem is one of the great advantages of MPC. The vehicle has not only mechanical constraints, but also regulation and comfort constraints:

- Regulation constraint. Maximum velocity;
- Comfort and regulation constraints. Maximum and minimum acceleration;
- Comfort constraints. Maximum and minimum jerk;
- Mechanical constraints. Maximum and minimum torque requests.

The maximum velocity constraint has to do with speed limits on roads, for example, a normal speed limit on a highway is $120 \frac{\text{km}}{\text{h}}$ and on a city street is $50 \frac{\text{km}}{\text{h}}$. So, to guarantee that the vehicle does not brake the law, it is important that these speed limits are taken into consideration. Considering that one of the states is directly the longitudinal velocity of the vehicle, this is an explicit constraint on the state of the vehicle. This constraint has the particularity of not being a hard constraint, i.e., it can be not satisfied in case it is needed. For example, if the vehicle was in a $120 \frac{\text{km}}{\text{h}}$ zone and changes to a $50 \frac{\text{km}}{\text{h}}$, the constraint might not be satisfied, because the vehicle does not instantaneously change its velocity. Therefore, it is necessary to add a slack variable to make sure that the

problem is feasible at all times

$$v_i \leq v_{\max} + \Delta s. \quad (3.26)$$

As it was shown in Chapter 2, v_i can be expressed as

$$v_{i+1}(T) = v_i(T_{i-1}) + \Delta T \left(\frac{2 \left(\frac{T_i}{r} \cos(\delta_{f_i}) - F_{C_{f_i}} \sin(\delta_{f_i}) \right) + 2 \left(\frac{T_i}{r} \cos(\delta_{r_i}) - F_{C_{r_i}} \sin(\delta_{r_i}) \right) - F_r - F_{a_i}}{M} + \dot{y}_i \dot{\psi}_i \right), \quad (3.27)$$

where, $T = u_{\text{long}} = \tilde{u}_{\text{long}} + u_{\text{long}_{\text{ref}}}$ is the control signal. The other "free variables" are δ_{f_i} and δ_{r_i} , which are inputs to the controller. So, it is now possible to write (3.26) with respect to \tilde{u}

$$\frac{2\Delta T}{rM} (\cos(\delta_{f_i}) + \cos(\delta_{r_i})) \tilde{u}_{\text{long}} \leq v_{\max} + \Delta s - v_{i+1} \left(u_{\text{long}_i}^{\text{ref}} \right). \quad (3.28)$$

The state v_i is not a prediction, it is the latest value measured and it is an initial condition. It is possible to write the previous equation in matrix form for the prediction horizon

$$D_{\text{velocity}} = \frac{2}{rM} \begin{bmatrix} \Delta T(\cos(\delta_{f_{i+1}}) + \cos(\delta_{r_{i+1}})) & 0 & \dots & 0 & -1 \\ \Delta T(\cos(\delta_{f_{i+1}}) + \cos(\delta_{r_{i+1}})) & \Delta T(\cos(\delta_{f_{i+2}}) + \cos(\delta_{r_{i+2}})) & \dots & 0 & -1 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ \Delta T(\cos(\delta_{f_{i+1}}) + \cos(\delta_{r_{i+1}})) & \Delta T(\cos(\delta_{f_{i+2}}) + \cos(\delta_{r_{i+2}})) & \dots & \Delta T(\cos(\delta_{f_{i+H-1}}) + \cos(\delta_{r_{i+H-1}})) & -1 \end{bmatrix}, \quad (3.29)$$

$$b_{\text{velocity}} = \begin{bmatrix} v_{\max} - v_{i+1} \left(u_{\text{long}}^{\text{ref}} \right) \\ v_{\max} - v_{i+2} \left(u_{\text{long}}^{\text{ref}} \right) \\ \vdots \\ v_{\max} - v_{i+H-1} \left(u_{\text{long}}^{\text{ref}} \right) \end{bmatrix}. \quad (3.30)$$

There is no direct measure of jerk, but it can be approximated by differences in acceleration

$$\frac{\dot{v}_{i+1} - \dot{v}_i}{\Delta T} \leq j_{\max}. \quad (3.31)$$

From (3.27), it is possible to express the acceleration as

$$\dot{v}_i(T_i) = \frac{2 \left(\frac{T_i}{r} \cos(\delta_{f_i}) - F_{C_{f_i}} \sin(\delta_{f_i}) \right) + 2 \left(\frac{T_i}{r} \cos(\delta_{r_i}) - F_{C_{r_i}} \sin(\delta_{r_i}) \right) - F_r - F_{a_i}}{M} + \dot{y}_i \dot{\psi}_i. \quad (3.32)$$

Following the same logic as before, it is possible to write the maximum jerk constraint in matrix form

$$D_{j_{\max}} = \frac{2}{rM} \begin{bmatrix} (\cos(\delta_{f_{i+1}}) + \cos(\delta_{r_{i+1}})) & 0 & \dots & 0 & 0 \\ 0 & (\cos(\delta_{f_{i+2}}) + \cos(\delta_{r_{i+2}})) & \dots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \dots & (\cos(\delta_{f_{i+H-1}}) + \cos(\delta_{r_{i+H-1}})) & 0 \end{bmatrix}, \quad (3.33)$$

$$b_{j_{\max}} = \begin{bmatrix} j_{\max} \Delta T + \dot{v}_i - \dot{v}_{i+1} (u_{\text{long}}^{\text{ref}}) \\ j_{\max} \Delta T + \dot{v}_{i+1} (u_{\text{long}}^{\text{ref}}) - \dot{v}_{i+2} (u_{\text{long}}^{\text{ref}}) \\ \vdots \\ j_{\max} \Delta T + \dot{v}_{i+H-2} (u_{\text{long}}^{\text{ref}}) - \dot{v}_{i+H-1} (u_{\text{long}}^{\text{ref}}) \end{bmatrix}. \quad (3.34)$$

The minimum jerk constraint can be written as

$$\begin{aligned} \frac{\dot{v}_{i+1} - \dot{v}_i}{\Delta T} &\geq j_{\min} \\ \Leftrightarrow \quad -\frac{\dot{v}_{i+1} - \dot{v}_i}{\Delta T} &\leq -j_{\min}. \end{aligned} \quad (3.35)$$

Again, by using the same logic it is possible to get the matrix form of the constraint

$$D_{j_{\min}} = -D_{j_{\max}}, \quad (3.36)$$

$$b_{j_{\min}} = \begin{bmatrix} -j_{\min} \Delta T - \dot{v}_i + \dot{v}_{i+1} (u_{\text{long}}^{\text{ref}}) \\ -j_{\min} \Delta T - \dot{v}_{i+1} (u_{\text{long}}^{\text{ref}}) + \dot{v}_{i+2} (u_{\text{long}}^{\text{ref}}) \\ \vdots \\ -j_{\min} \Delta T - \dot{v}_{i+H-2} (u_{\text{long}}^{\text{ref}}) + \dot{v}_{i+H-1} (u_{\text{long}}^{\text{ref}}) \end{bmatrix}. \quad (3.37)$$

Finally, it is possible to write $D_{\text{long}_{\text{ineq}}}$ and $b_{\text{long}_{\text{ineq}}}$

$$D_{\text{long}_{\text{ineq}}} = \begin{bmatrix} D_{\text{velocity}} \\ D_{j_{\max}} \\ D_{j_{\min}} \end{bmatrix}, \quad (3.38) \quad b_{\text{long}_{\text{ineq}}} = \begin{bmatrix} b_{\text{velocity}} \\ b_{j_{\max}} \\ b_{j_{\min}} \end{bmatrix}. \quad (3.39)$$

The maximum and minimum acceleration constraint is not only about comfort, but also about regulation of the GCDC 2016. This constraint can be expressed as

$$\dot{v}_{\min} \leq \dot{v}_i \leq \dot{v}_{\max} \quad (3.40)$$

Applying the same algebraic manipulation as before, it is possible to obtain

$$a_{\max_i} = \frac{r \left((\dot{v}_{\max} - \dot{y}_i \dot{\psi}_i) M + 2F_{C_{f_i}} + 2F_{C_{r_i}} + F_r + F_{a_i} \right)}{2 (\cos(\delta_{f_i}) + \cos(\delta_{r_i}))} - T_{\text{ref}}, \quad (3.41)$$

$$a_{\min_i} = \frac{r \left((\dot{v}_{\min} - \dot{y}_i \dot{\psi}_i) M + 2F_{C_{f_i}} + 2F_{C_{r_i}} + F_r + F_{a_i} \right)}{2 (\cos(\delta_{f_i}) + \cos(\delta_{r_i}))} - T_{\text{ref}}. \quad (3.42)$$

The upper and lower bound for the slack variable are Δs_{\max} and Δs_{\min} , respectively. These bounds are set due to the solver. To set up the solver, it did not compile and create the desired code when the slack variable is left completely free. So, bounds are imposed, but these bounds are always large enough so they are never reached and the slack variable is completely "free".

Finally, it is possible to express the vectors ub_{long} and lb_{long} ,

$$ub_{\text{long}} = \begin{bmatrix} a_{\max_i} \\ a_{\max_{i+1}} \\ \vdots \\ a_{\max_{i+H-1}} \\ \Delta s_{\max} \end{bmatrix}, \quad (3.43) \quad lb_{\text{long}} = \begin{bmatrix} a_{\min_i} \\ a_{\min_{i+1}} \\ \vdots \\ a_{\min_{i+H-1}} \\ \Delta s_{\min} \end{bmatrix}. \quad (3.44)$$

The maximum and minimum torque requests are due to constraints in the motors and are an explicit constraint on the control signal.

3.5.2 Lateral Controller

The main objective of the lateral controller is to follow a path, a reference that can be, for example, the road the vehicle is traveling on. The controller is not supposed to follow a trajectory, because it is only controlling the steering angles of the wheels and it is impossible to follow a trajectory with only that control over the vehicle. On the other hand, it is possible to guarantee that the vehicle follows a path, i.e., it is possible to make the vehicle, for example, follow the center line of a lane.

So, to follow a path there are three important variables:

- X position of the vehicle;
- Y position of the vehicle;
- Ψ heading of the vehicle.

The reference path is given with respect to the vehicle current position and the states used in the MPC are

$$z_{\text{lat}} = \begin{bmatrix} x \\ y \\ \psi \\ \dot{x} \\ \dot{y} \\ \dot{\psi} \end{bmatrix}. \quad (3.45)$$

These states can be calculated as

$$x_{i+1} = x_i + \Delta T (v_i \cos(\psi_i) - \dot{y}_i \sin(\psi_i)), \quad (3.46a)$$

$$y_{i+1} = y_i + \Delta T (v_i \sin(\psi_i) + \dot{y}_i \cos(\psi_i)), \quad (3.46b)$$

$$\psi_{i+1} = \psi_i + \Delta T \dot{\psi}_i, \quad (3.46c)$$

$$\dot{x}_{i+1} = \dot{x}_i + \Delta T \left(\frac{2F_{x_{f_i}} + 2F_{x_{r_i}} - F_r - F_{a_i}}{M} + \dot{y}_i \dot{\psi}_i \right), \quad (3.46d)$$

$$\dot{y}_{i+1} = \dot{y}_i + \Delta T \left(\frac{2F_{y_{f_i}} + 2F_{y_{r_i}}}{M} - \dot{x}_i \dot{\psi}_i \right), \quad (3.46e)$$

$$\dot{\psi}_{i+1} = \dot{\psi}_i + \Delta T \frac{2aF_{y_{f_i}} - 2bF_{y_{r_i}}}{I}. \quad (3.46f)$$

It is possible to define the optimization problem for the lateral controller as

$$\begin{aligned} \underset{\tilde{u}_{\text{lat}}}{\text{minimize}} \quad & \tilde{z}_{\text{lat}_H}^T Q_{\text{lat}_f} \tilde{z}_{\text{lat}_H} + \tilde{u}_{\text{lat}_H}^T R_{\text{lat}_f} \tilde{u}_{\text{lat}_H} + \sum_{i=1}^{H-1} (\tilde{z}_{\text{lat}_i}^T Q_{\text{lat}} \tilde{z}_{\text{lat}_i} + \tilde{u}_{\text{lat}_i}^T R_{\text{lat}} \tilde{u}_{\text{lat}_i}) \\ \text{subject to} \quad & \tilde{z}_{\text{lat}_{i+1}} = A \tilde{z}_{\text{lat}_i} + B \tilde{u}_{\text{lat}_i}, \\ & \tilde{z}_{\text{lat}_i} = z_{\text{lat}_i} - z_{\text{lat}_i}^{\text{ref}}, \\ & \tilde{u}_{\text{lat}_i} = u_{\text{lat}_i} - u_{\text{lat}_i}^{\text{ref}}, \\ & D_{\text{lat}_{\text{ineq}}} [\tilde{u}_{\text{lat}_1}; \tilde{u}_{\text{lat}_2}; \dots; \tilde{u}_{\text{lat}_H}] \leq b_{\text{lat}_{\text{ineq}}}, \\ & lb_{\text{lat}_i} \leq \tilde{u}_{\text{lat}_i} \leq ub_{\text{lat}_i}, \end{aligned} \quad (3.47)$$

and in the QP form as

$$\begin{aligned}
& \underset{\bar{u}_{\text{lat}}}{\text{minimize}} \quad \bar{u}_{\text{lat}}^T H_{\text{lat}} \bar{u}_{\text{lat}} + f_{\text{lat}}^T \bar{u}_{\text{lat}} \\
& \text{subject to} \quad D_{\text{lat}_{\text{ineq}}} \bar{u}_{\text{lat}} \leq b_{\text{lat}_{\text{ineq}}}, \\
& \quad l b_{\text{lat}} \leq \bar{u}_{\text{lat}} \leq u b_{\text{lat}},
\end{aligned} \tag{3.48}$$

where, u is the input to the vehicle, which in this case are the two steering angles request for the front and rear wheel of the bicycle model

$$u_{\text{lat}_i} = \begin{bmatrix} \delta_{f_i} \\ \delta_{r_i} \end{bmatrix}. \tag{3.49}$$

States Linearization

To formulate the problem (3.48), it is necessary to calculate matrices H_{lat_k} and f_{lat_k} , to do this, first, it is necessary to calculate matrices A_{lat} and B_{lat} of the linearized state space error model

$$\tilde{z}_{\text{lat}_{i+1}} = A_{\text{lat}_i} \tilde{z}_{\text{lat}_i} + B_{\text{lat}_i} \tilde{u}_{\text{lat}_i}, \tag{3.50}$$

which can be obtained by taking the partial derivatives at the reference points. Due to the size of the matrices A_{lat} and B_{lat} they are included in Appendix A.

In the lateral controller case there is no operation mode, it is always performing the same task, so one can express the current error vector as

$$\tilde{z}_{\text{lat}_i} = [-x_{\text{ref}_i}; -y_{\text{ref}_i}; -\psi_{\text{ref}_i}; -\dot{x}_{\text{ref}_i} + \dot{x}_i; -\dot{y}_{\text{ref}_i} + \dot{y}_i; -\dot{\psi}_{\text{ref}_i} + \dot{\psi}_i], \tag{3.51}$$

where, x_{ref_i} , y_{ref_i} and ψ_{ref_i} are defined in the local frame of the vehicle.

The weight matrices can be expressed as, $Q_{\text{lat}_i} = \text{diag}(q_{x_i}, q_{y_i}, q_{\psi_i}, q_{\dot{x}_i}, q_{\dot{y}_i}, q_{\dot{\psi}_i})$ and $R_{\text{lat}_i} = \text{diag}(q_{\delta_{f_i}}, q_{\delta_{r_i}})$. Finally, by using (3.11), (3.12) and (3.14) it is possible to calculate H_{lat_i} and f_{lat_i} .

Constraints

In this case, the constraints are mainly mechanical, but also related to comfort:

- Mechanical constraints. Maximum and minimum wheel angle;
- Mechanical and comfort constraints. Maximum and minimum wheel angle rate.

Since the control signals are directly the angle of the wheels, it is trivial to express the constraints. The maximum wheel angle rate, can be expressed as

$$D_{\dot{\delta}_{\max}} = \begin{bmatrix} 1 & 0 & 0 & 0 & \dots & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & \dots & 0 & 0 & 0 \\ -1 & 0 & 1 & 0 & \dots & 0 & 0 & 0 \\ 0 & -1 & 0 & 1 & \dots & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \dots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & \dots & -1 & 0 & 1 \end{bmatrix}, \quad b_{\dot{\delta}_{\max}} = \begin{bmatrix} \dot{\delta}_{\max} \Delta T + \delta_f - \delta_{f_{\text{ref}_i}} \\ \dot{\delta}_{\max} \Delta T + \delta_r - \delta_{r_{\text{ref}_i}} \\ \dot{\delta}_{\max} \Delta T + \delta_{f_{\text{ref}_i}} - \delta_{f_{\text{ref}_{i+1}}} \\ \dot{\delta}_{\max} \Delta T + \delta_{r_{\text{ref}_i}} - \delta_{r_{\text{ref}_{i+1}}} \\ \vdots \\ \dot{\delta}_{\max} \Delta T + \delta_{f_{\text{ref}_{i+H-2}}} - \delta_{f_{\text{ref}_{i+H-1}}} \\ \dot{\delta}_{\max} \Delta T + \delta_{r_{\text{ref}_{i+H-2}}} - \delta_{r_{\text{ref}_{i+H-1}}} \end{bmatrix}. \quad (3.52) \quad (3.53)$$

By using the same algebraic manipulation that was used for the minimum jerk in the longitudinal control (3.35), it is possible to obtain the minimum wheel angle rate constraints

$$D_{\dot{\delta}_{\min}} = -D_{\dot{\delta}_{\max}}, \quad (3.54)$$

$$b_{\dot{\delta}_{\min}} = \begin{bmatrix} -\dot{\delta}_{\min} \Delta T - \delta_f + \delta_{f_{\text{ref}_i}} \\ -\dot{\delta}_{\min} \Delta T - \delta_r + \delta_{r_{\text{ref}_i}} \\ -\dot{\delta}_{\min} \Delta T - \delta_{f_{\text{ref}_i}} + \delta_{f_{\text{ref}_{i+1}}} \\ -\dot{\delta}_{\min} \Delta T - \delta_{r_{\text{ref}_i}} + \delta_{r_{\text{ref}_{i+1}}} \\ \vdots \\ -\dot{\delta}_{\min} \Delta T - \delta_{f_{\text{ref}_{i+H-2}}} + \delta_{f_{\text{ref}_{i+H-1}}} \\ -\dot{\delta}_{\min} \Delta T - \delta_{r_{\text{ref}_{i+H-2}}} + \delta_{r_{\text{ref}_{i+H-1}}} \end{bmatrix}. \quad (3.55)$$

Finally, it is possible to write $D_{\text{lat}_{\text{ineq}}}$ and $b_{\text{lat}_{\text{ineq}}}$

$$D_{\text{lat}_{\text{ineq}}} = \begin{bmatrix} D_{\dot{\delta}_{\max}} \\ D_{\dot{\delta}_{\min}} \end{bmatrix}, \quad (3.56) \quad b_{\text{lat}_{\text{ineq}}} = \begin{bmatrix} b_{\dot{\delta}_{\max}} \\ b_{\dot{\delta}_{\min}} \end{bmatrix}. \quad (3.57)$$

The maximum and minimum wheel angle is a direct constraint on the control signal and can be expressed as

$$ub_{\text{lat}} = \begin{bmatrix} \delta_{\max} - \delta_{f_{\text{ref}_i}} \\ \delta_{\max} - \delta_{r_{\text{ref}_i}} \\ \delta_{\max} - \delta_{f_{\text{ref}_{i+1}}} \\ \delta_{\max} - \delta_{r_{\text{ref}_{i+1}}} \\ \vdots \\ \delta_{\max} - \delta_{f_{\text{ref}_{i+H-1}}} \\ \delta_{\max} - \delta_{r_{\text{ref}_{i+H-1}}} \end{bmatrix}, \quad (3.58)$$

$$lb_{\text{lat}} = \begin{bmatrix} \delta_{\min} - \delta_{f_{\text{ref}_i}} \\ \delta_{\min} - \delta_{r_{\text{ref}_i}} \\ \delta_{\min} - \delta_{f_{\text{ref}_{i+1}}} \\ \delta_{\min} - \delta_{r_{\text{ref}_{i+1}}} \\ \vdots \\ \delta_{\min} - \delta_{f_{\text{ref}_{i+H-1}}} \\ \delta_{\min} - \delta_{r_{\text{ref}_{i+H-1}}} \end{bmatrix}. \quad (3.59)$$

3.6 Reference Generation

In order to use LTV-MPC for reference tracking, it is necessary to calculate references for all states and control signals used in both controllers.

Since both controllers are supposed to be independent it is important to calculate the references independently for each controller. However, the full functionality of the controllers is only unlocked when they are used together.

The MPC is able to deal with some non feasible paths and references. For example, it is a non feasible path if it does not account for the turning of the wheels when requesting for crabbing, but the MPC will adjust and position the vehicle the best way possible to minimize the state errors.

3.6.1 Longitudinal

As expressed before the states of the longitudinal controller are

$$z_{\text{long}} = \begin{bmatrix} \Delta d \\ d \\ v \end{bmatrix}, \quad (3.60)$$

and the control signal is the torque of the motors. So, it is necessary to calculate one reference value for each variable at every time instant of the horizon.

There are two possible operation modes for the controller, namely CACC and CC. The way the references are calculated depend on which mode is active.

CACC

In this operation mode, as it is shown in Section 3.5.1, all the states are active, which means that all of them need references.

The first state, Δd , always have all its references at zero, this is due to the fact that the vehicle should always be at the desired distance so the reference value for the error in distance is zero.

The second state, d , has different references depending on the platoon/vehicle velocity and acceleration that is to be followed, and can be expressed as

$$d_i^{\text{ref}} = h v_{\text{platoon}_i}^{\text{ref}} + r_{\text{safe}}, \quad (3.61)$$

where,

$$v_{\text{platoon}_{i+1}}^{\text{ref}} = v_{\text{platoon}_i}^{\text{ref}} + a_{\text{platoon}}^{\text{ref}} \Delta T, \quad (3.62)$$

where, $a_{\text{ref platoon}}$ is the acceleration of the platoon and it is assumed constant during the prediction at each time instant. Note that $v_{\text{ref platoon}_i}$ and $a_{\text{ref platoon}}$ is a known value, which is received through communications or estimated from the different values of velocity of the vehicles platooning.

The third state, v , references are equal to $v_{\text{platoon}}^{\text{ref}}$.

Finally, it is possible to calculate the torque references. These references come directly from the desired velocity references through some algebraic manipulations to (3.27) and it can be calculated as

$$T_i^{\text{ref}} = \frac{\left(\frac{v_{\text{platoon}_{i+1}}^{\text{ref}} - v_{\text{platoon}_i}^{\text{ref}}}{\Delta T} - \dot{y}_i \dot{\psi}_i \right) M + F_{C_{f_i}} \sin(\delta_{f_i}) + F_{C_{r_i}} \sin(\delta_{r_i}) + F_r + F_a}{\frac{2}{r} (\cos(\delta_{f_i}) + \cos(\delta_{r_i}))}, \quad (3.63)$$

since the values of \dot{y}_i and $\dot{\psi}_i$ are needed to calculate the references of torque, the vehicle model is used to predict them, i.e., using the values inputted to the controller of current v , \dot{y} , $\dot{\psi}$ and previous torque reference calculated, a state prediction is made and used in the next torque calculation.

Some precision of the controllers is lost when both, the lateral and the longitudinal controllers, are not running at the same time. For example, when the lateral controller is not running, there is no vector containing the horizon prediction for the front and rear steering wheel angles. So, to deal with this problem, it is assumed that the steering angles are zero throughout the horizon. Although this is obviously an approximation, it is good enough since it is assumed that the vehicle is mostly moving in the x direction when the lateral controller is not on.

To calculate the references for the controller, the velocity and acceleration of the platoon are used so, it is important to specify where they come from and how they are processed. There are different ways to get values for these variables:

- V2V communication retrieves all the variables from all the vehicles in the platoon. Then, it is possible to create a weighed sum of these values to estimate the velocity and acceleration of the platoon;
- Sensor measurements, for example, radar gives directly the measures of the vehicle in front of the ego vehicle. In this case, only the front vehicle would be used as a reference;
- Virtual measurements, for example, a ghost vehicle can be created in software to tune the controller and do specific tasks.

CC

In this operation mode only one state is active in the optimization, which is the velocity of the ego vehicle. So, only this state needs reference values.

The reference values are calculated according to what the driver/passenger chooses as final velocity and desired acceleration. With this two values, the reference velocity values are calculated and, then, using (3.63) the torque references are computed.

3.6.2 Lateral

The lateral references are the first calculation at every controller cycle, which means it is necessary to estimate the future trajectory of the vehicle based on the last optimization. To do this, in a normal MPC formulation, where every point in the horizon is isochronal, one would only have to shift the result vector from the last optimization by one time sample. But, in this case, the prediction horizon is not isochronal, as shown in Figure 3.1. So, it is necessary to predict what will be the control signals at the corresponding time instants and, with those, use the vehicle model to predict the states, which, in turn, is used to obtain the reference positions in the given path.

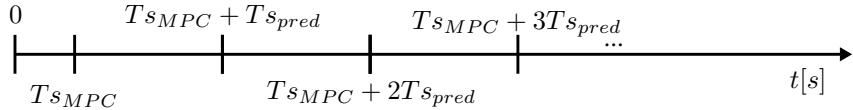


Figure 3.1: Horizon sample times. The first sample is made according to the rate of the controllers and the rest according to the variable Ts_{pred} .

The input to the lateral controller is a local reference for x and y position and a local reference for ψ the vehicle heading.

A valid approximation to what the optimization problem will give as result, on the corresponding time samples, is obtained by linear interpolation of the latest output of the controllers and sample at the times needed, i.e.,

$$u_{i|i} = u_{i-1|i-1} + \frac{Ts_{MPC}}{Ts_{pred}} (u_{i|i-1} - u_{i-1|i-1}), \quad (3.64)$$

where, the notation a_{ilk} means the value of variable a at instant i predicted at instant k , $Ts_{MPC} = \frac{1}{f_{MPC}}$ and f_{MPC} is the frequency at which the controllers are run, Ts_{pred} is the time interval between each point of the horizon, note that the time interval between the current point and the first point in the horizon is still Ts_{MPC} . For the last point, the last value of the horizon is repeated,

$$u_{i+H-1|i} = u_{i+H-1|i-1}. \quad (3.65)$$

Using the vehicle model, it is possible to compute future predicted velocities, which allows to find a trajectory on the given path. Note that to do this, it is assumed the vehicle is following the path, i.e., distance covered in the horizon is distance covered in the path. It is important to calculate this trajectory, because all the requests are calculated to a specific point in time. So, to match the steering requests to the torque requests, one needs to predict where the vehicle is going to be, by following the previous computed torque profile, and calculate the steering

references to those positions.

Now that the trajectory points are calculated, it is possible to calculate references for the front and rear wheel angles of the bicycle model. The reference calculation can be divided in two parts, the crabbing angle and the curvature angle for each wheel.

To calculate the crab angle, β , between two points of the trajectory, it is necessary to rotate the vector between those two points by the direction of movement. This, basically, eliminates the influence of the curvature part of the trajectory and leaves the crabbing movement in the trajectory. It can be computed by

$$Rot(\theta) = \begin{bmatrix} \cos(\theta) & \sin(\theta) \\ -\sin(\theta) & \cos(\theta) \end{bmatrix}, \quad (3.66)$$

$$\begin{bmatrix} x_{\text{crab}_{i+1}}^{\text{traj}} \\ y_{\text{crab}_{i+1}}^{\text{traj}} \end{bmatrix} = Rot(\psi_i^{\text{traj}}) \begin{bmatrix} x_{i+1}^{\text{traj}} - x_i^{\text{traj}} \\ y_{i+1}^{\text{traj}} - y_i^{\text{traj}} \end{bmatrix}, \quad (3.67)$$

$$\beta_{\text{ref}_i} = \text{atan} \left(\frac{y_{\text{crab}_{i+1}}^{\text{traj}} - y_{\text{crab}_i}^{\text{traj}}}{x_{\text{crab}_{i+1}}^{\text{traj}} - x_{\text{crab}_i}^{\text{traj}}} \right), \quad (3.68)$$

where, ψ^{traj} is the reference of direction that the vehicle is suppose to follow, normally it is the direction of the road. This way, the vehicle always faces the road while changing lane, doing other maneuvers and/or correcting error.

To calculate the curvature angle reference for each wheel, the turning radius and rotation point of the curve has to be calculated. To do this, perpendicular lines to the direction of the vehicle are drawn at each two points of the trajectory and the intersection of those two lines is obtained, which is the rotation point. From this point, it is possible to calculate the radius of curvature, which then allows for the calculation of γ , the curvature term of the angle of each wheel. The geometrical relation is similar to the one depicted in Figure 2.3, but the wheels are positioned as the ones in Figure 2.4. It can be calculated as

$$\gamma_{f_{\text{ref}_i}} = \text{sign}(\psi_{i+1}^{\text{traj}} - \psi_i^{\text{traj}}) \text{atan} \left(\frac{\cos(\beta_{\text{ref}_i})a}{\text{sign}(\psi_{i+1}^{\text{traj}} - \psi_i^{\text{traj}}) \sin(\beta_{\text{ref}_i})a + R} \right), \quad (3.69a)$$

$$\gamma_{r_{\text{ref}_i}} = \text{sign}(\psi_{i+1}^{\text{traj}} - \psi_i^{\text{traj}}) \text{atan} \left(\frac{\cos(\beta_{\text{ref}_i})b}{-\text{sign}(\psi_{i+1}^{\text{traj}} - \psi_i^{\text{traj}}) \sin(\beta_{\text{ref}_i})b + R} \right). \quad (3.69b)$$

Finally, it is possible to calculate the full steering angle of each wheel

$$\delta_{f_{\text{ref}_i}} = \beta_{\text{ref}_i} + \gamma_{f_{\text{ref}_i}}, \quad (3.70a)$$

$$\delta_{r_{\text{ref}_i}} = \beta_{\text{ref}_i} - \gamma_{r_{\text{ref}_i}}. \quad (3.70b)$$

The last references that need to be calculated are \dot{x} , \dot{y} and $\dot{\psi}$. The first two are calculated from the crabbing

trajectory and the last is calculated from the desired direction references

$$\dot{x}_{\text{ref}_i} = \frac{x_{\text{crab}_{i+1}}^{\text{traj}} - x_{\text{crab}_i}^{\text{traj}}}{\Delta T}, \quad (3.71\text{a})$$

$$\dot{y}_{\text{ref}_i} = \frac{y_{\text{crab}_{i+1}}^{\text{traj}} - y_{\text{crab}_i}^{\text{traj}}}{\Delta T}, \quad (3.71\text{b})$$

$$\dot{\psi}_{\text{ref}_i} = \frac{\psi_{i+1}^{\text{traj}} - \psi_i^{\text{traj}}}{\Delta T}. \quad (3.71\text{c})$$

Chapter 4

Implementation

In this chapter, the implementation details are presented. It starts by describing the inputs and outputs of the controllers. Then, the software used is detailed. After, the prediction horizon and controller frequency are defined. Finally, controller specifications, such as, constants used, are described and explained.

4.1 Controllers I/Os

In Figure 4.1 it is possible to see how both controllers are connected.

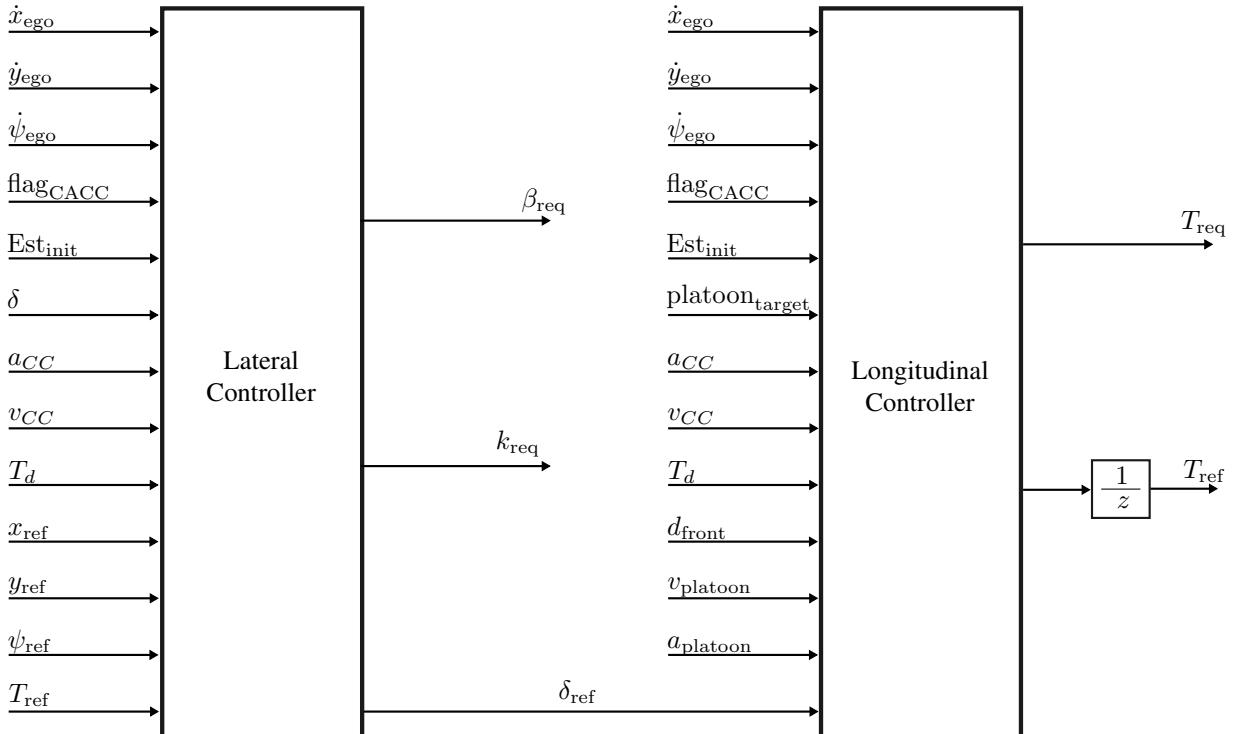


Figure 4.1: Controllers Scheme

The inputs to the lateral controller are:

- \dot{x}_{ego} , \dot{y}_{ego} and $\dot{\psi}_{ego}$ are the current state of the vehicle;

- $\text{flag}_{\text{CACC}}$ is the flag that decides the operation mode of the controllers. If equal to 1 the controller is in platoon mode, which for the lateral controller means the reference calculation is based on the last torque values outputted (T_{ref}) by the longitudinal controller and last steering angles computed. If equal to 0 the controller is doing CC, using acceleration, a_{CC} , and velocity, v_{CC} , set by the driver to determine the distance that is traveled by following the specific velocity profile;
- Est_{init} is the estimator flag that indicates if the state values being inputted are already valid or not, if not the controllers output zeros;
- δ contains both the current δ_f and δ_r ;
- T_d is the time delay of the system. This time constant, if used, allows for the controller to take into consideration the delay of the actuation. This ended up not being used, because it is not necessary, the controllers are able to deal with the small actuation delay without having to know about it;
- x_{ref} , y_{ref} and ψ_{ref} are the path that the vehicle is going to follow.

The outputs are:

- β is the desired crabbing angle;
- k is the desired curvature;
- δ_{ref} contains the predictions for δ_f and δ_r during the horizon, which are used by the longitudinal controller.

The longitudinal controller only has 4 different inputs from the lateral:

- d_{front} is the distance to the most important object, normally the front vehicle;
- v_{platoon} is the current velocity of the platoon;
- a_{platoon} is the current acceleration of the platoon;
- $\text{platoon}_{\text{target}}$ is the id of the vehicle being followed, to whom d_{front} corresponds to. This only matters, because when the vehicle being followed changes, the gains used by the controller also change, to have a smooth transition when changing targets.

The outputs are:

- T_{req} is the torque request sent to the low-level controllers;
- T_{ref} contains the predictions for torque during the horizon, which are used by the lateral controller on the next controller cycle.

It is important to define the software used to implement, considering it has a huge influence on the performance and design of the controllers.

4.2 Software

The main software used is MATLAB and simulink real-time. The controllers optimization problems are solved with a tool made by Jacob Mattingley and Stephen Boyd called CVXGEN [23], since MATLAB solvers are too slow for this type of application. This tool generates fast custom code for small, QP-representable convex optimization problems.

CVXGEN is a tool, where the optimization problem that needs to be solved is set up and, the tool, generates C-code with a custom solver specifically designed for the problem.

To have full functionality on both controllers they have to communicate with each other, to do that it is easier if their prediction horizons match.

4.3 Prediction horizon and Controller Frequency

It is important to have a large horizon, but a large horizon means a big computational cost and since the controllers are suppose to run in real-time, there are hardware limitations that need to be taken into consideration. The horizon is chosen to have 20 samples, i.e., $H = 20$. This means that at every time instant the MPC will predict twenty samples ahead. The faster the controllers run the better, because it increases the reaction time to errors and the number of control signals sent per second, which allows for more precision in the movements. The problem is that the faster it runs, the bigger the load on the CPU and if only one horizon is used it becomes a very small time prediction that does not look into the future that much, for example, if the controller runs at a frequency of $50Hz$ with twenty samples, with a constant time between samples and only one horizon, it only looks ahead $0.4s$, which is a small value.

Instead of using the common way of increasing the horizon time by using two horizons, a control horizon and a prediction horizon, a different approach is taken. Two different time constants are set, Ts_{MPC} and Ts_{pred} , the first one is the inverse of the frequency rate of the controllers (f_{MPC}) and the second one is a free time constant that can be set to whatever value needed to match a desired time of prediction horizon. These constants represent the time in between samples of the horizon. The first point of the horizon has to obey the rate of the controllers, because the control signal being sent has to match the time it is going to actuate. The rest of the points in the horizon can be further apart from each other, so, they obey Ts_{pred} . This way, with $Ts_{pred} > Ts_{MPC}$ the horizon is larger than only using Ts_{MPC} . This is shown in Figure 3.1.

The controllers run at $f_{MPC} = 33.33Hz$, which is the maximum feasible rate before the CPU overloads. So, $Ts_{MPC} = 0.03s$.

After setting the controller rate, different values for the Ts_{pred} were tested in simulation and in the RCV. Good results, where the vehicle behaves as expected and predicts what's going to happen with enough time, were obtained for a $Ts_{pred} = 0.14$. It is important to notice that, if the points are spaced too far apart from each other the discrete model used does not predict the vehicle motion with good accuracy, so, this value can not grow indefinitely.

The horizon is now $t_{horizon} = 19Ts_{pred} + Ts_{MPC} = 2.69s$, which means that for a vehicle moving at $10\frac{m}{s}$ the controller will be predicting $26.9m$ ahead.

Having defined the rate of the controllers and the time of the predictions, the next step is to define the controllers specifications.

Abbreviation	Value	Unit
T_{\min}	$\frac{-130}{\sqrt{2}}$	Nm
T_{\max}	$\frac{130}{\sqrt{2}}$	Nm
\dot{v}_{\min}	-2	$\frac{m}{s^2}$
\dot{v}_{\max}	2	$\frac{m}{s^2}$
v_{\max}	14	$\frac{m}{s}$
j_{\max}	1	$\frac{m}{s^3}$
j_{\min}	-3	$\frac{m}{s^3}$
$q_{\Delta d}$	5	-
q_d	43	-
q_v	90	-
$q_{\Delta d_f}$	7.5	-
q_{d_f}	63	-
q_{v_f}	135	-
q_T	0.01	-
$q_{\Delta s}$	10^{10}	-
$q_{\Delta d_{\text{change}}}$	5	-
$q_{d_{\text{change}}}$	20	-
$q_{\dot{x}_{\text{change}}}$	90	-
n_{change}	200	-
r_{safe}	1	s
h	10	m

Table 4.1: Constant values used in longitudinal controller

4.4 Controllers Specifications

To finalize the controllers, the constraints and weights used need to be defined.

4.4.1 Longitudinal

The values used in the controller are all presented in Table 4.1, where, n_{change} is the number of cycles that the controller will run, when the target id changes, before it changes back to the normal weights and the weights $(.)_{\text{change}}$ are the corresponding weights.

Note that the value for maximum and minimum torque is divided by $\sqrt{2}$, because in the low-level controllers there is a division by that value, so to overcome that problem without changing the low-level control, all the control signals sent out by the controller are multiplied by $\sqrt{2}$. This value is chosen to be 130Nm instead of the maximum of the motors 150Nm , because, once again, there is a cutoff at 130Nm at the low-level controllers.

The acceleration values are chosen to follow the rules of the competition, the jerk and velocity values are chosen for comfort and because of the hardware limitations of the RCV.

To define the gains $(.)_{change}$, it was trial and error with the starting point on the normal operation gains. The reasoning behind is that the ego vehicle needs to care less about distance when changing target. So, the weight of distance is reduced. To define n_{change} , first it is defined how long this gains should actuate for and then divided by the rate of the controller. To define the actuation time, the time the crossing vehicle, in the intersection scenario, takes to reach the center point of the intersection is taken into consideration, this time is 6s, because the vehicle travels at $30 \frac{km}{h}$ for 50m, so $n_{change} = \frac{6}{Ts_{MPC}} = 200$.

Finally, r_{safe} and h are the values used in the GCDC 2016. These values are high and the controllers could use lower values and still maintain safety.

The static error in the acceleration value is big enough to block the torque requests (because of the constraints) which makes the vehicle to not start moving. To solve this issue, instead of using the true value, \dot{v}_i , the predicted value by the model of the last controllers iteration is used. One disadvantage that this fix brings, is that it allows the controller to "think" the vehicle is moving when it is not, i.e., when stopped and the longitudinal control is not on, the torque request value still becomes bigger, which leads to a jerky first movement of the vehicle. To minimize this effect, the initial state of the vehicle is well chosen, i.e., the vehicle starts with a small error to the reference values, this causes the torque requests to be close to zero.

4.4.2 Lateral Controller

The values used in the controller are all presented in Table 4.2, where, the values for lb_{lat} and ub_{lat} were set conservatively to make sure the actuators can handle the request. For the same reason, but also because of comfort $\dot{\delta}_{max}$ and $\dot{\delta}_{min}$ are set with a low value.

The gains are set intuitively, by giving a big value to q_ψ the controllers are forced to follow the heading references and to use crabbing to correct the error in the other states. The same reasoning is valid for q_ψ . The states q_x and q_y have the same value, both of them represent the position, which is what is being corrected in this controller. However, the states $q_{\dot{x}}$ and $q_{\dot{y}}$ do not have the same weight, because $q_{\dot{x}}$ is controlled by the longitudinal controller and not the lateral, but $q_{\dot{y}}$ is controlled by the lateral controller so it has a bigger weight.

These weights practically did not need tuning, because they are only used in simulations. All the weights with 0 value are not needed in simulation so they are not used, but, it is believed, they will be needed on the RCV and that is why they are implemented.

It is important to mention that all the weights, for the lateral controller, will have to be tuned for the RCV.

Abbreviation	Value	Unit
δ_{\min}	-15	deg
δ_{\max}	15	deg
$\dot{\delta}_{\max}$	6	$\frac{\text{deg}}{\text{s}}$
$\dot{\delta}_{\min}$	-6	$\frac{\text{deg}}{\text{s}}$
q_x	100	-
q_y	100	-
q_{ψ}	8000	-
$q_{\dot{x}}$	1	-
$q_{\dot{y}}$	100	-
$q_{\dot{\psi}}$	200	-
q_{δ_f}	0	-
q_{δ_r}	0	-
q_{x_f}	0	-
q_{y_f}	0	-
q_{ψ_f}	0	-
$q_{\dot{x}_f}$	0	-
$q_{\dot{y}_f}$	0	-
$q_{\dot{\psi}_f}$	0	-

Table 4.2: Constant values used in lateral controller

Chapter 5

Results

The RCV is fully built at KTH by multiple students and researchers and, because of this, it is more prone to errors and problems than a normal vehicle. In the Netherlands, while competing in the GCDC 2016, a component from the main battery burned, and although it was replaced, the cause of the problem was in the original mount design of the battery, which had to be redone and could not be fixed there. Due to this problem, and some other mechanical problems, for example, on the steering actuators of the wheels, the development, implementation and tuning of the controllers were delayed and the lateral controller could not be implemented in the RCV in time.

In this chapter, the results from actual tests made on the RCV are presented for the longitudinal controller. For the lateral controller, some simulation results are presented. Many tests are performed, but it is impossible to present them all, so, some tests that show how the controllers work in different situations are chosen.

5.1 Longitudinal

The longitudinal controller was first tested in simulation, which always worked and it was easy to tune. In simulation, it did not require all the gains to be tuned, some could be set to zero and it would still work. This happened, because the model being used in the controller is the same as the model simulating the vehicle, so, the controller was able to predict very precisely what the vehicle would do with a specific input, which allowed for a near perfect control¹. But, since this thesis is about implementing on the RCV, the simulation results are not presented.

The vehicle could not compete in the GCDC 2016, so there are no results from the competition, but given that the scenarios are known, it is possible to fake the scenarios and simulate the different situations to evaluate how the controller handles those situations. The different situations are:

- Platoon, maintaining a certain distance from the front vehicle;
- Platoon, with strange behavior of the vehicles in front, such as hard braking;
- While platooning, change in target vehicle, which might cause the distance to the front vehicle to jump to a small or even negative value;

¹The control is not perfect, because, as mentioned before, the controllers and the simulation run at different rates so the models end up not being exactly the same.

- Cruise control;
- Intersection scenario, which is similar to the change in target while platooning, but adds a time constraint, because the crossing vehicle goes through the intersection at a certain time.

5.1.1 Platooning

To simulate the platoon, a ghost vehicle is created. Note that there is no point in creating multiple ghost vehicles, because the platoon can be described by an acceleration and a velocity, so, one ghost vehicle is enough.

To implement this ghost vehicle its acceleration is controlled and integrated twice to get velocity and distance.

Test 1: Normal behavior of front vehicle

In Figures 5.1, 5.2 and 5.3 the results for the first test are presented.

In Figure 5.1, at around 23s, the ego vehicle changes its acceleration value as expected, since the ghost vehicle is no longer accelerating, the reference values of velocity are now constant which for the controller means that the error is not increasing and it is just a matter of correcting the current error, which allows for a much smoother and calmer approach to reach the desired velocity and compensate for the errors in distance. When accelerating, the ego vehicle does not follow perfectly the velocity of the front vehicle, because it is gaining distance from it, the desired distance to the front vehicle increases with velocity.

After reaching the desired velocity there is a small overshoot, to, once again slowly, correct the distance state, which as it is possible to see in Figure 5.2 is not perfect by the time the ego vehicle reaches the desired velocity. Note that the test had to be finished before the ego vehicle reached a stable state on the distance, because the road ended, but it is pretty close and it would have reached it.

Although Figure 5.3 is called errors, this cannot be viewed as a normal error ($error = actual - desired$). Distance and velocity are both connected so, the fact they are not following the references is not an error. To correct distance the velocity state has to change and this obviously makes the ego vehicle not follow the ghost vehicle velocity, but it cannot be considered an error because it is made on purpose to correct the distance state. The maximum velocity error obtained is just under $0.16 \frac{m}{s}$ and the minimum above $-1.32 \frac{m}{s}$ but, as explained before, this is to correct the distance state.

Also, in Figure 5.3, the maximum and minimum errors in distance are about 0.92 and -2.4 meters. This result is further explained in Test 2.

Test 2: Changes in velocity of front vehicle

In Figures 5.4, 5.5 and 5.6 the results for the second test are presented. In this run, first the ghost vehicle has a constant acceleration of $1 \frac{m}{s^2}$ until it reaches a velocity of $4.21 \frac{m}{s}$ and then after the ego vehicle stabilizes with a distance error of $0.5m$, the ghost vehicle is accelerated once again to around $8.23 \frac{m}{s}$ and finally the velocity was decreased to $3.46 \frac{m}{s}$ with a constant acceleration of $-1 \frac{m}{s^2}$.

The maximum velocity error obtained is just under $1 \frac{m}{s}$ and the minimum just above $-1 \frac{m}{s}$ but, as explained before, this is to correct the distance state.

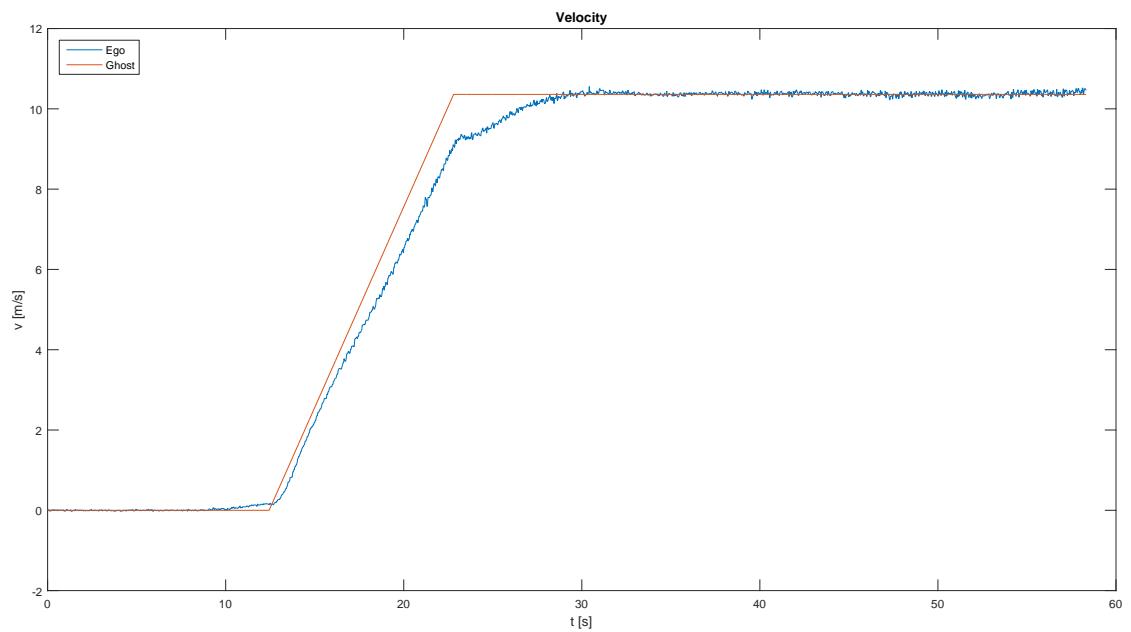


Figure 5.1: Velocity profile of the RCV following a target.

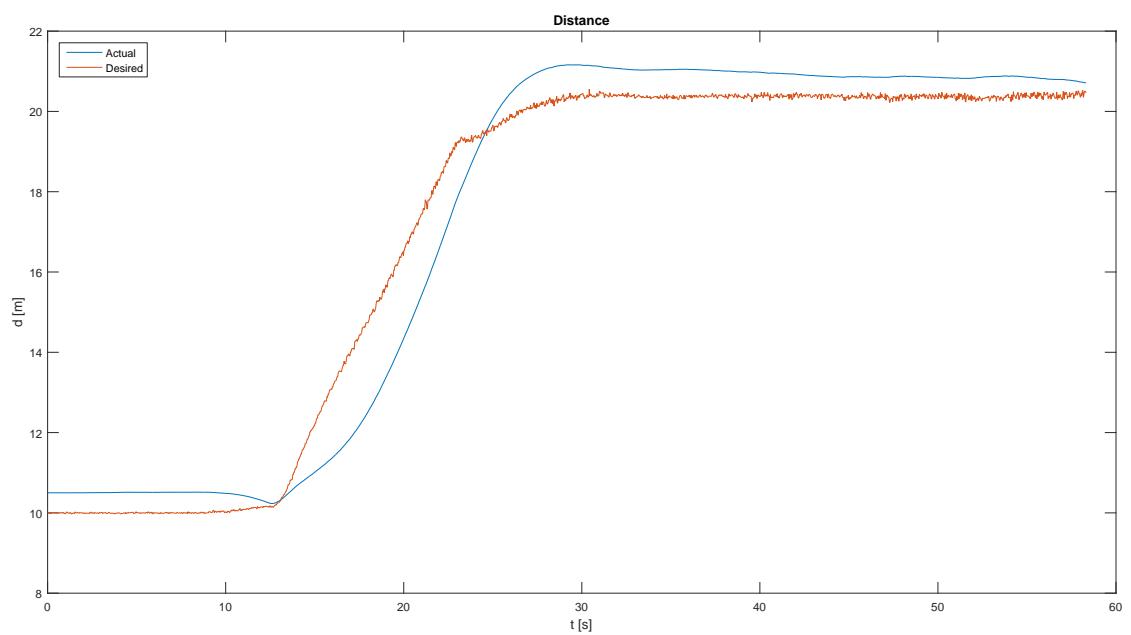


Figure 5.2: Distance profile of the RCV following a target.

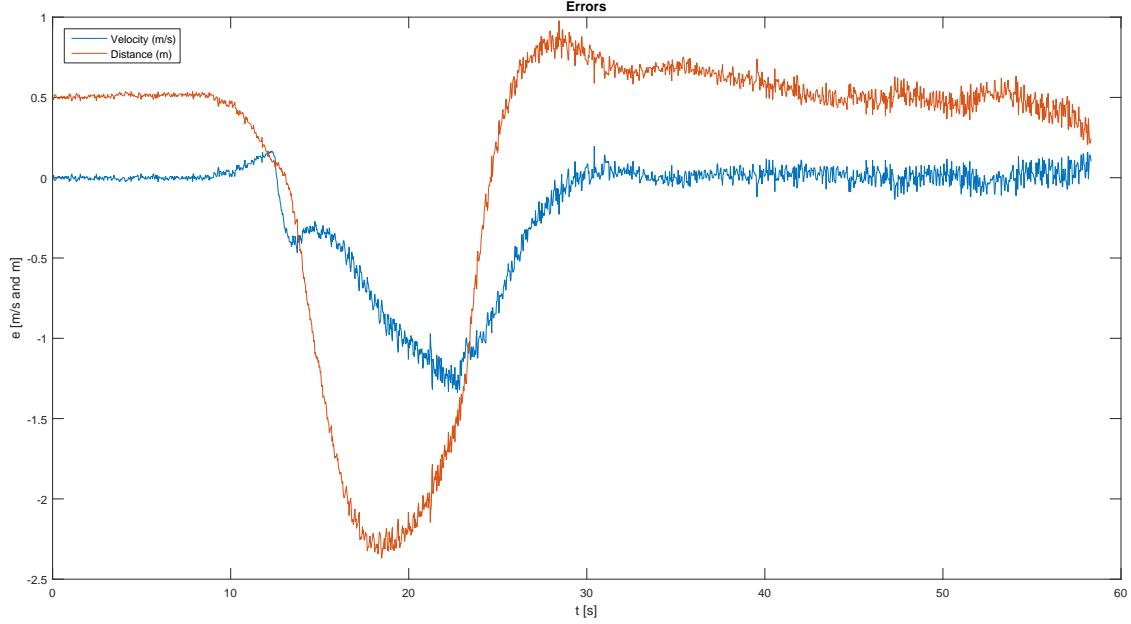


Figure 5.3: Difference between actual and desired values of velocity and distance for the RCV following a target.

On a steady state, where the velocity state matches the desired velocity, the maximum error of distance was around $0.5m$ which shows the vehicle is behaving as expected, and not only following the velocity profile, but also keeping the desired distances. The vehicle battery has influence on the correction of these errors and on the accuracy of the vehicle prediction model, because, with low battery levels, the torque requests sent may not be completely actuated on and this causes a static error, in this case, it is $0.5m$.

In Figure 5.6, it is possible to see that the maximum and minimum errors in distance are about 2 and -2 meters. At $20s$ the error is due to the fact that the ego vehicle is following the velocity of the ghost very accurately and does not care much about distance, which when accelerating leads to a negative error in distance and when decelerating leads to a positive error in distance, this is the desired behavior intended. It is important to notice that when accelerating the ego vehicle can follow the same or very close acceleration, because the vehicle in front is getting away from the ego vehicle, but when the front vehicle decelerates fast the ego vehicle should decelerate at least at the same rate, so it not only keeps the safe distance but is even further away than the desired distance just in case the other vehicle continues to stop or something else happens.

Test 3: Change in target being followed

This test consists of evaluating the behavior of the ego vehicle when the platooning target is changed or, when on the intersection scenario, the ego vehicle starts following the crossing vehicle.

To do the test, the ghost vehicle distance is changed to zero instantaneously when both vehicles are in a steady state. From Figures 5.7, 5.8 and 5.9 it is possible to see that this change happens at $24s$ and by $30.39s$ the ego vehicle is already at the desired distance. This means, that the ego vehicle takes $6.39s$ to reach the desired distance, i.e., after $6.39s$ of changing platoon target, that vehicle is able to merge if it wants to or after $6.39s$ the crossing vehicle is fully safe to cross the intersection.

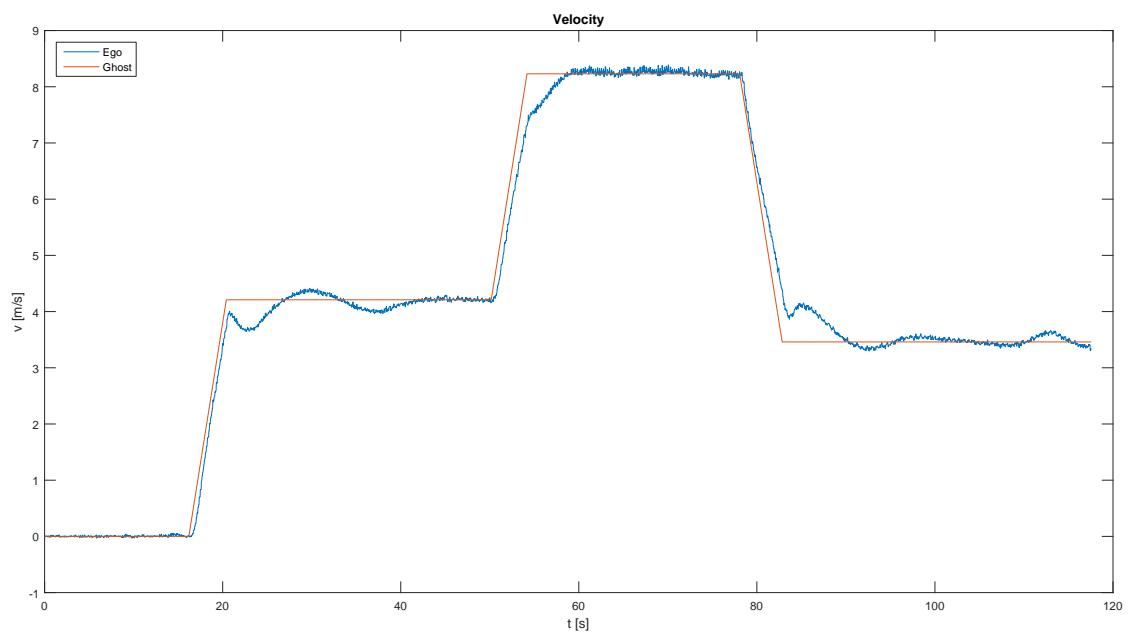


Figure 5.4: Velocity profile of the RCV following a target changing velocity.

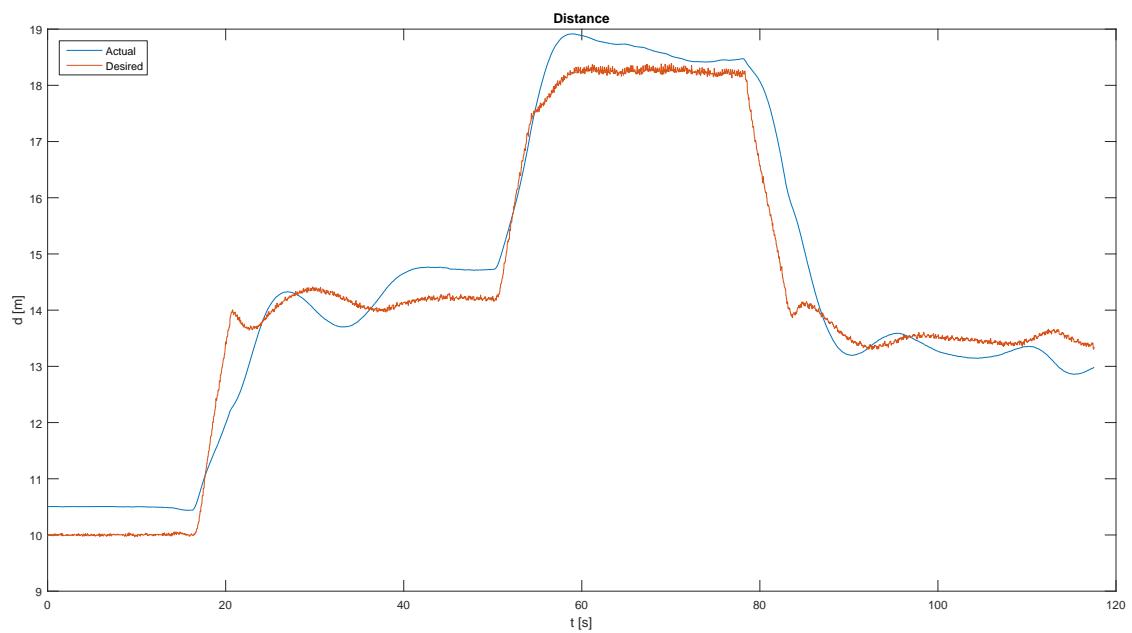


Figure 5.5: Distance profile of the RCV following a target changing velocity.

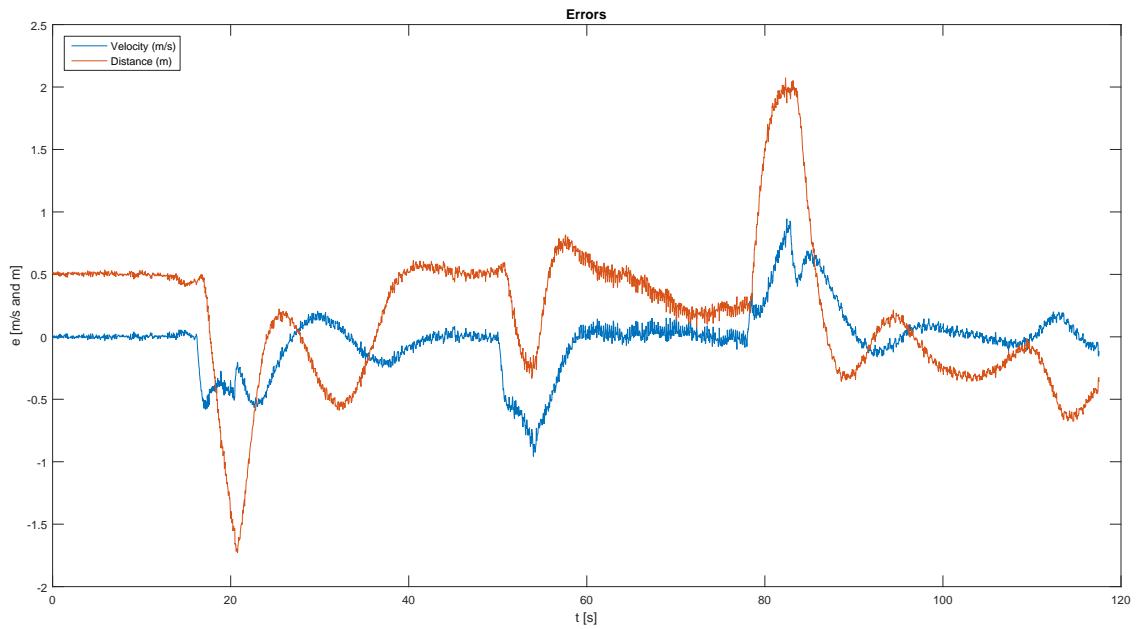


Figure 5.6: Difference between actual and desired values of velocity and distance for the RCV following a target changing velocity.

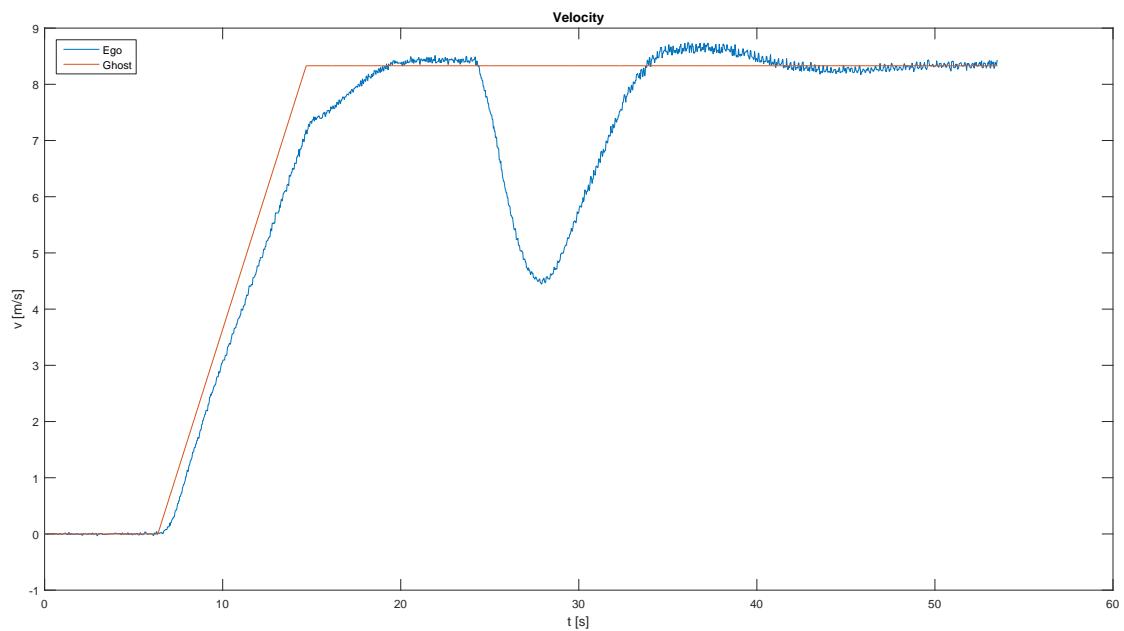


Figure 5.7: Velocity profile of the RCV for a sudden change in distance to the front vehicle.

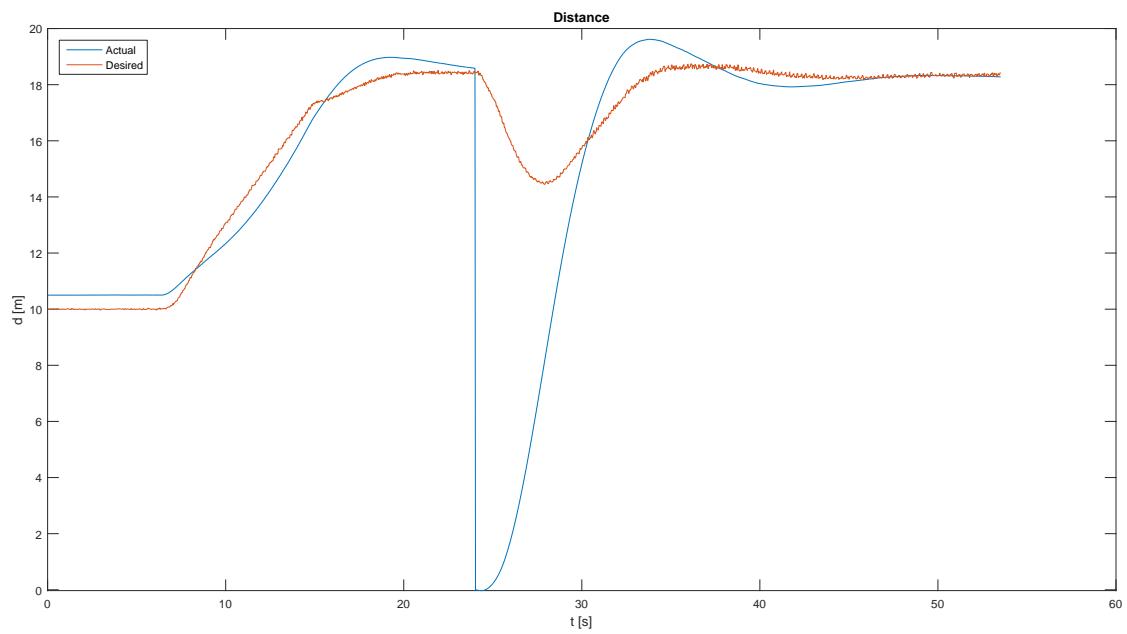


Figure 5.8: Distance profile of the RCV for a sudden change in distance to the front vehicle.

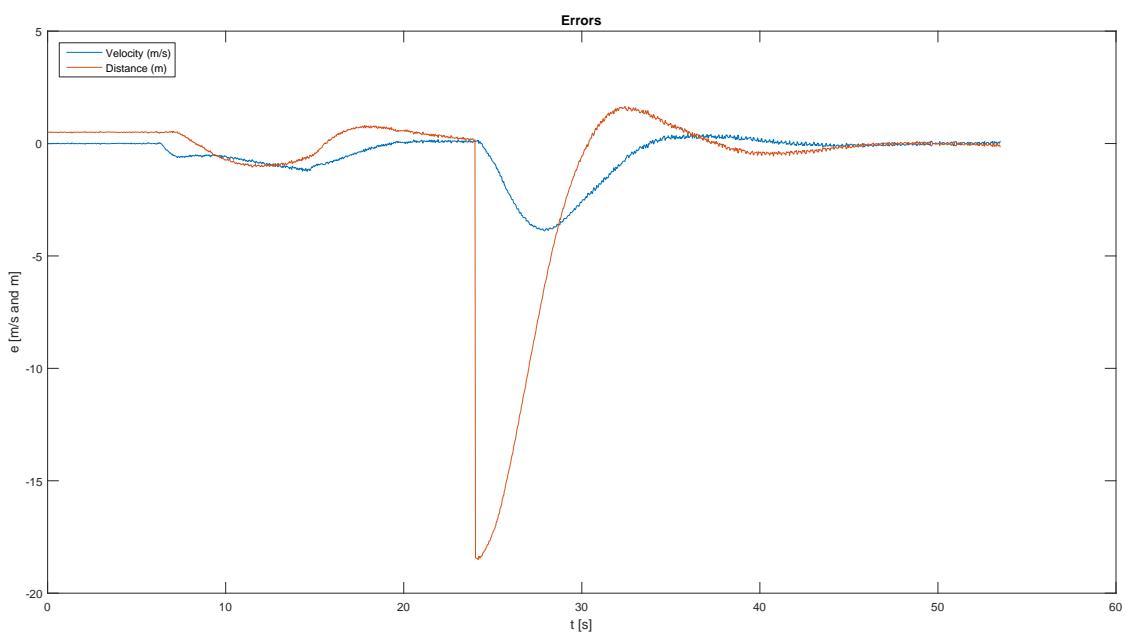


Figure 5.9: Difference between actual and desired values for a sudden change in distance to the front vehicle.

In the crossing scenario, the vehicles are at $50m$ from the middle point of the intersection when they enter the competition zone and they are traveling at a speed of $30\frac{km}{h} \simeq 8.33\frac{m}{s}$. So, the crossing vehicle reaches the intersection point in $6s$, a bit less than what the controller is tuned for to be at the desired distance. Although, it is easy to change this to get a faster response, it was decided not to do it because it would make the reaction more aggressive, i.e., the ego vehicle would have a bigger deceleration and it would not make it safer. $6s$ after the ghost vehicle changed to $0m$, the ego vehicle is more than $15m$ away, which is more than most vehicles did in the competition and their crossings, with smaller distances, were considered safe.

This test was performed while the vehicle battery was still at a good level so, the steady state error, is close to zero. With a good battery level, the prediction model is accurate to lead to the desired behavior of the vehicle, which, ultimately, leads to steady state errors of 0 .

In Figures 5.7, 5.8 and 5.9 the vehicle behaves as expected, it corrects both velocity and distance errors in a smooth way without braking to hard, even when the distance to the front vehicle is set to $0m$. Once again, the overshoots seen in the figures are related to the corrections in velocity and distance and are expected from the start.

Test 4: Signals from front vehicle with noise

This test could not be done at a normal test track so, it is performed at a lower speed, because there is not enough space for the RCV to run at normal speeds, i.e., the speeds of the previous tests.

To simulate noisy measurements from the front vehicle gaussian noise is added to the outputs of the ghost vehicle. Since, what is tested in this case is how well the controller deals with noisy measurements no type of filtering is added to the noisy signals.

In Figures 5.10, 5.11 and 5.12 the results are presented and in Figure 5.13 the noise added to the signals is shown. The noise for velocity of the ghost vehicle has a maximum and a minimum of $1.4137\frac{m}{s}$ and $-1.6636\frac{m}{s}$, respectively , for distance $4.2412m$ and $-4.9907m$ and for acceleration $0.2827\frac{m}{s^2}$ and $-0.3327\frac{m}{s^2}$.

The maximum noise added corresponds to about 30% of the true signals values, which is much more than what normal noisy measurements would have. The amount of noise added is so big, because when tested with lower amounts of noise the RCV behaved nicely without any big difference from the other tests, so, the noise is increased until a point the ego vehicle starts to be affected by it.

Even with the noise added to the ghost vehicle, the ego vehicle is able to follow without noticeable oscillations, although, as it is shown in Figure 5.12, the error is bigger than in the other tests, mostly because the signal being received in the controllers is full of noise. The maximum a minimum error for velocity is $1.5350\frac{m}{s}$ and $-2.5763\frac{m}{s}$ and for distance $6.0453m$ and $-4.0428m$. If the noise is removed from the signals, it is possible to see the actual values being kept, this is shown in Figures 5.14, 5.15 and 5.16. The real maximum and minimum errors are $0.7376\frac{m}{s}$ and $-1.4804\frac{m}{s}$ for velocity and $3.0205m$ and $0.2441m$ for distance, which are much closer to the results obtained in the previous tests and prove that the controller is able to handle measurements with noise.

5.1.2 Cruise Control

The results obtained are similar to each other, with small differences depending on the battery level, inclination of the road and other small perturbations not modeled in the controller which can only be corrected a posteriori.

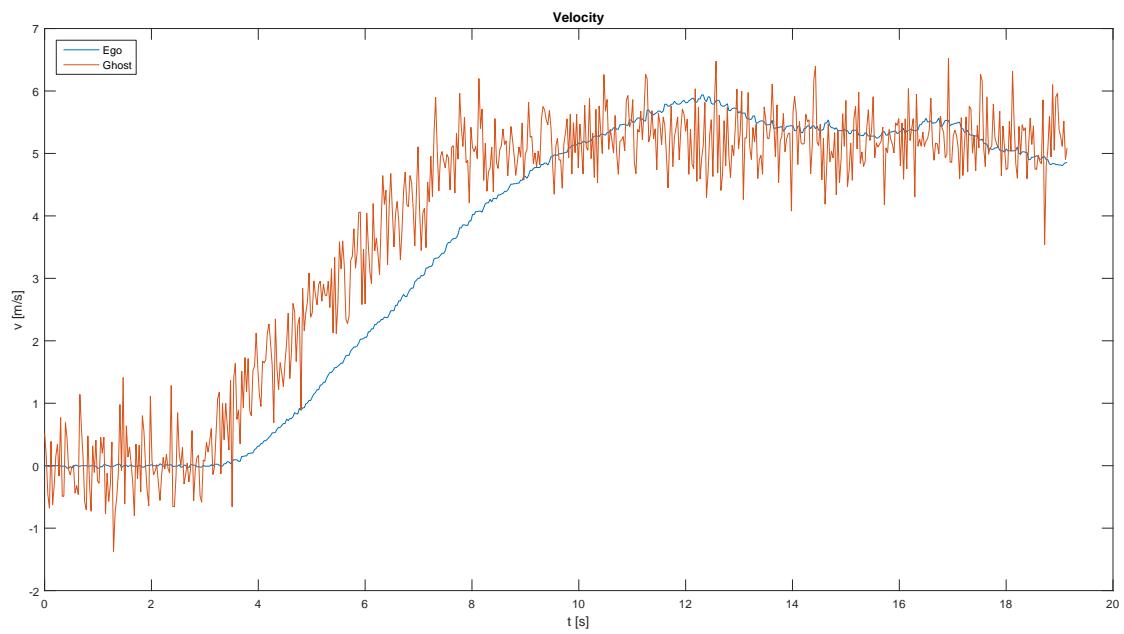


Figure 5.10: Velocity profile of the RCV for noisy signals from the front vehicle.

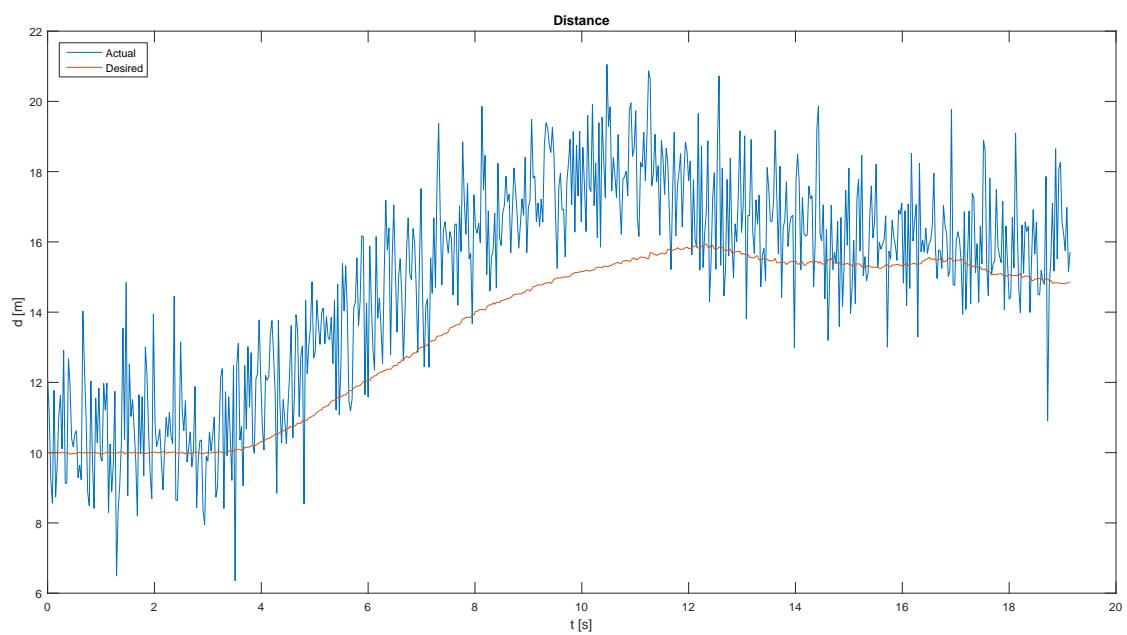


Figure 5.11: Distance profile of the RCV for noisy signals from the front vehicle.

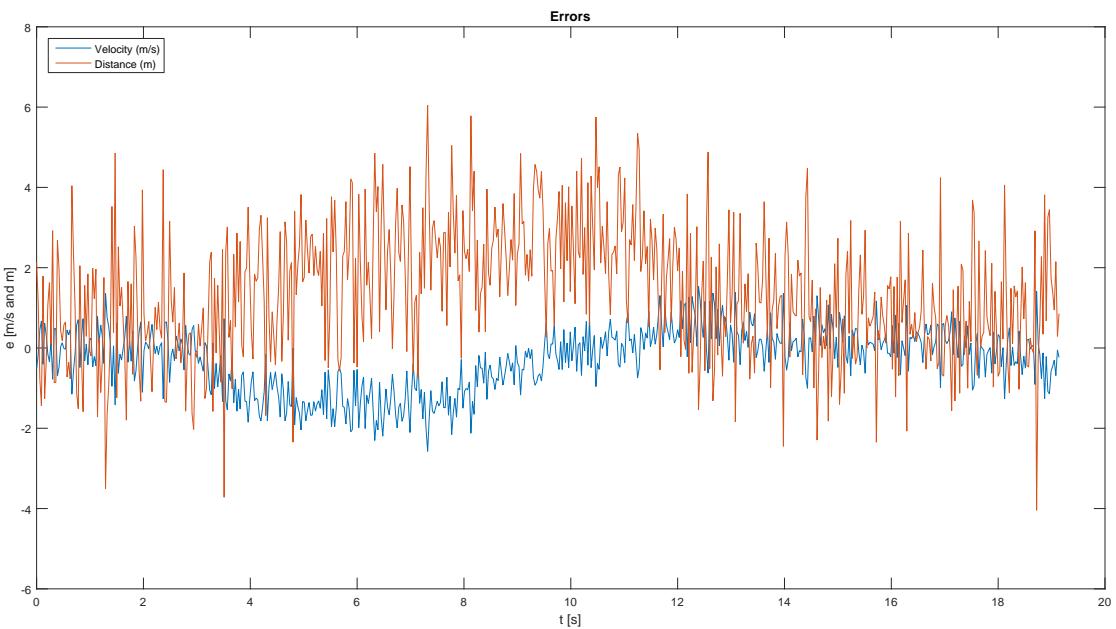


Figure 5.12: Difference between actual and desired values for noisy signals from the front vehicle.

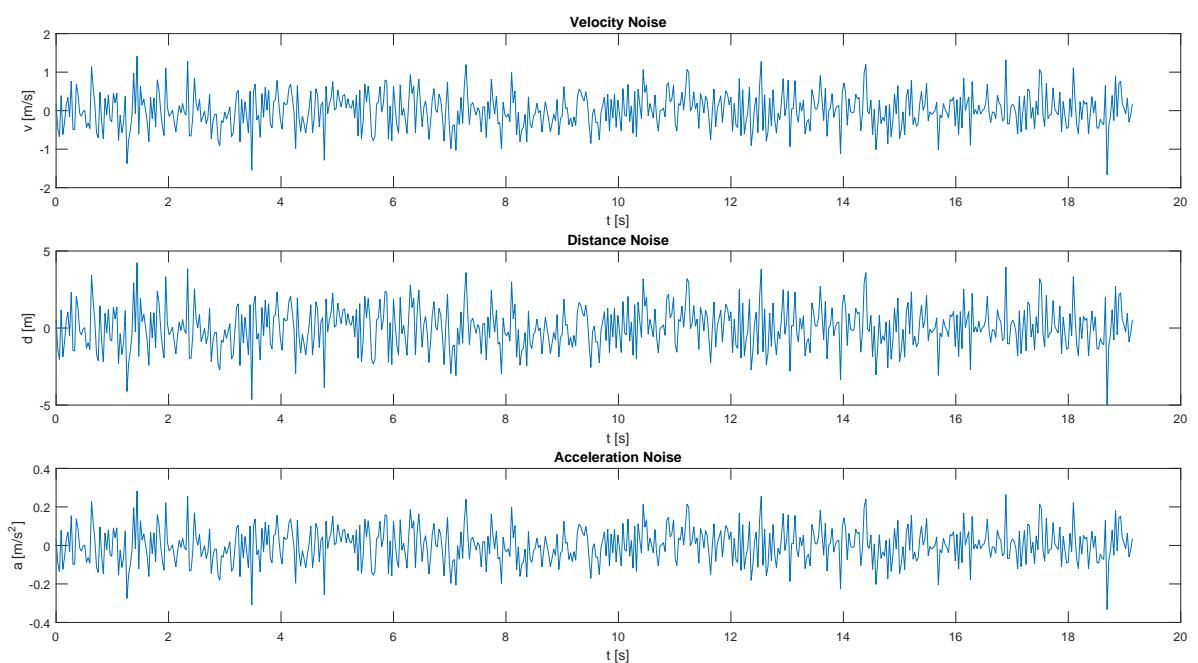


Figure 5.13: Noise added to the signals of the front vehicle.

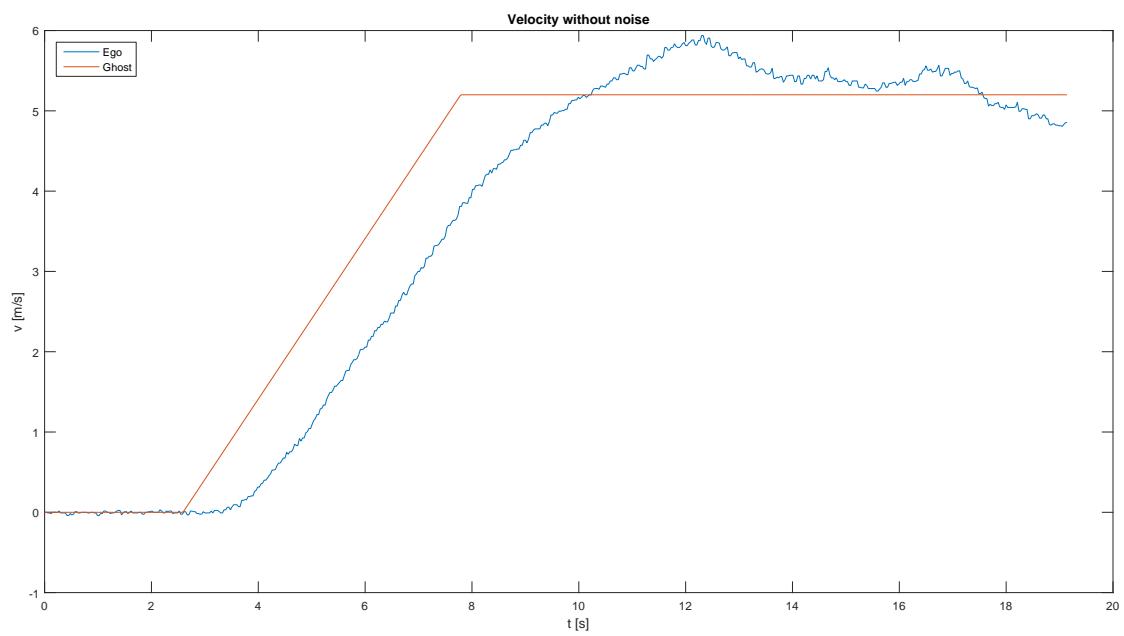


Figure 5.14: Velocity profile of the RCV for true signals from the front vehicle.

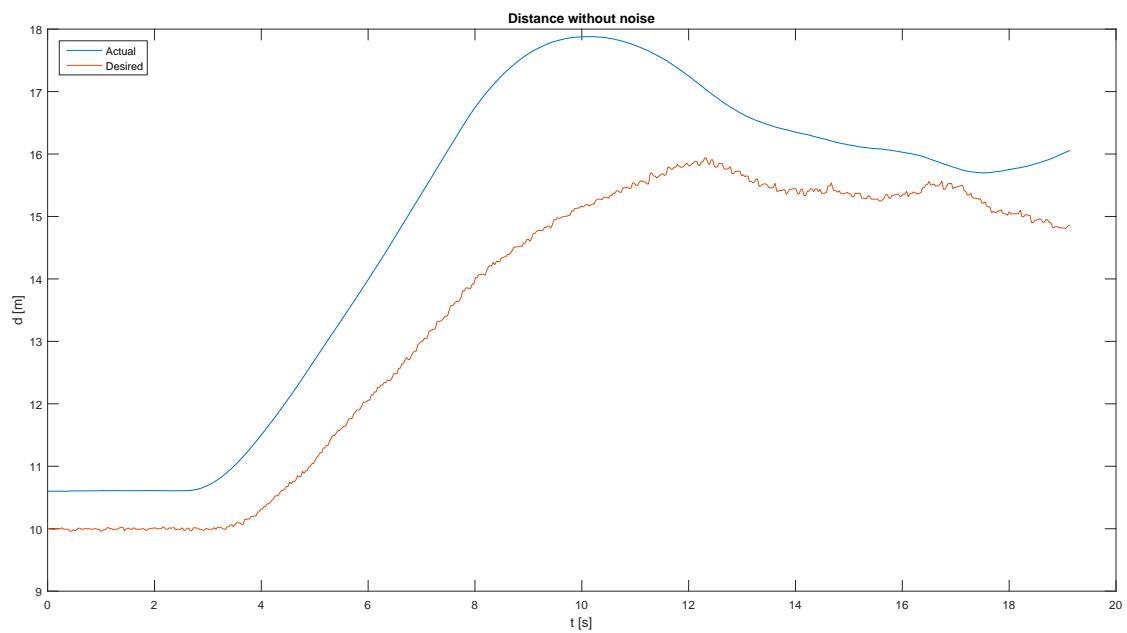


Figure 5.15: Distance profile of the RCV for true signals from the front vehicle.

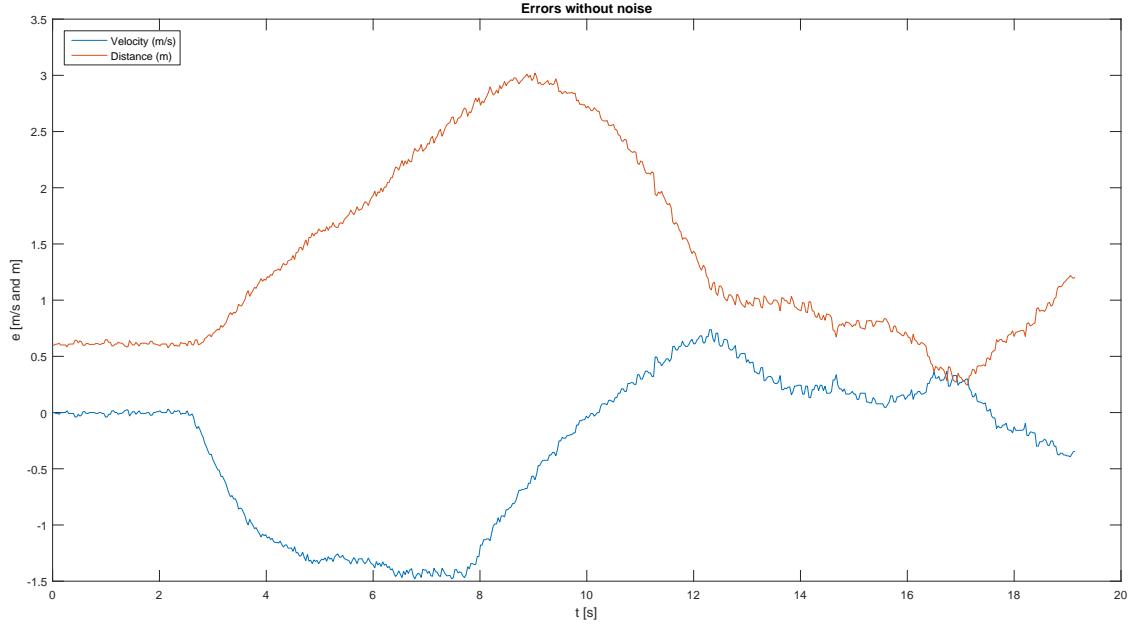


Figure 5.16: Difference between actual and desired values for true signals from the front vehicle.

Test 1: Cruise Control ($7 \frac{m}{s}$)

This test consists of checking how the ego vehicle behaves to a request of a CC speed of $7 \frac{m}{s}$ and an acceleration of $0, 5 \frac{m}{s^2}$. Figures 5.17 and 5.17 present the results.

The vehicle behaved as expected, but since the road was slightly inclined, the vehicle gained more speed than expected, about $0, 3 \frac{m}{s}$, which it ended up correcting after a few seconds. The inclination of the road was not taken into consideration in the model of the vehicle so it is impossible to avoid it, the vehicle has to "see" this inclination by the error of the vehicle and, after detecting it, it is corrected.

Test 2: Cruise Control ($12 \frac{m}{s}$)

This test is similar to the previous, the end velocity is $12 \frac{m}{s}$ and the acceleration is $1 \frac{m}{s^2}$. Figures 5.19 and 5.19 present the results of the test.

It is clear that the vehicle reached the desired velocity but, to reach this velocity, approximately $42 \frac{km}{h}$, some of the torque requests made caused the battery voltage to drop and were cut in the low level controllers², although, the controller is still keeping the theoretical limits. This happened, because the velocity requested is high, which in turn causes the torque requests of the controller to be high and with a low battery level, high torque requests, cause a big voltage drop. The maximum error was around $-1 \frac{m}{s}$ and the steady state error was around $0 \frac{m}{s}$.

In the beginning of the scenario, the RCV is on a road with a small inclination, which caused the vehicle to start moving slowly forward, which then made the vehicle brake, causing a wave in velocity that can be seen at 5s in Figure 5.19.

²The MPC controller does not know which torque requests are cut, because these cuts depend on the battery level, so, it is impossible to know where to stop requesting for more torque and predict the behavior.

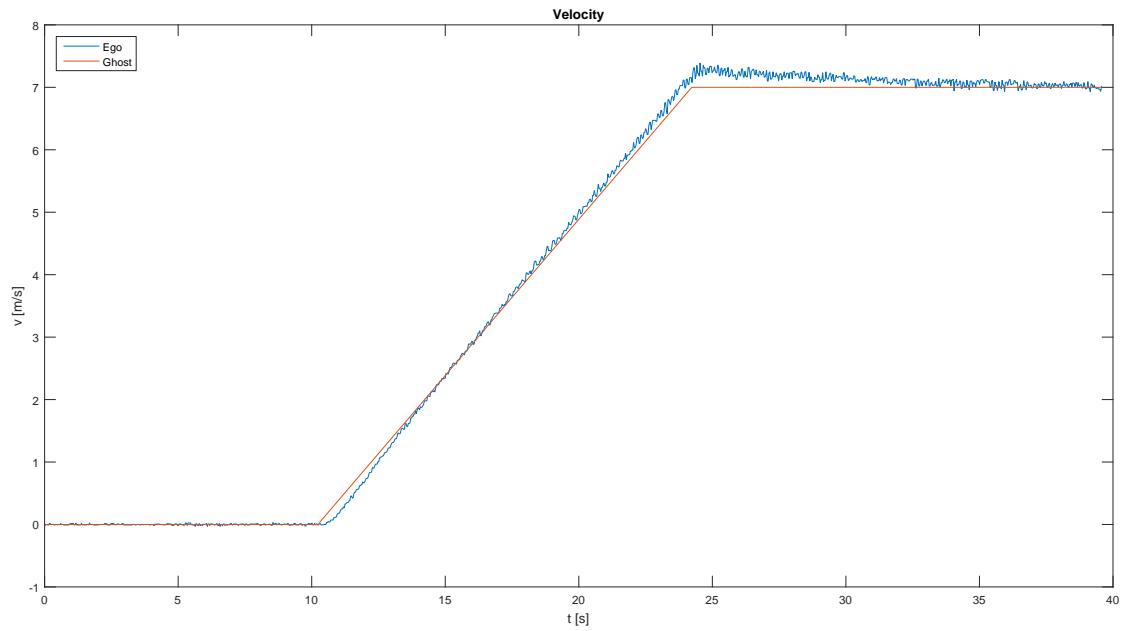


Figure 5.17: Velocity profile of the RCV following a speed of $7 \frac{m}{s}$ and a acceleration of $0.5 \frac{m}{s^2}$.

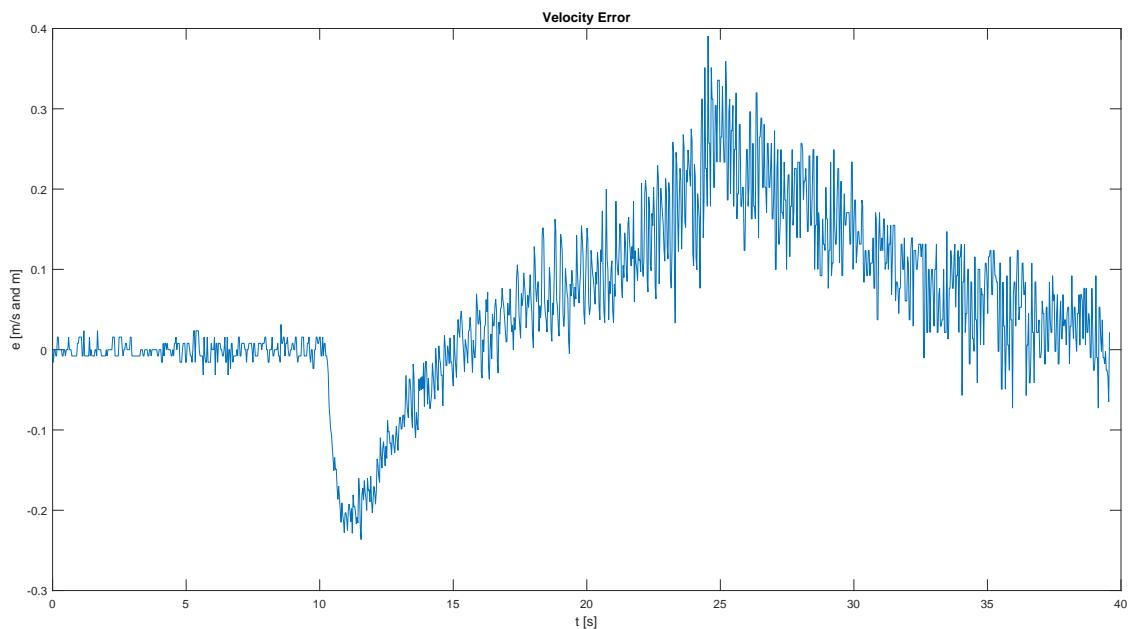


Figure 5.18: Difference between actual and desired values of the RCV following a speed of $7 \frac{m}{s}$ and a acceleration of $0.5 \frac{m}{s^2}$.

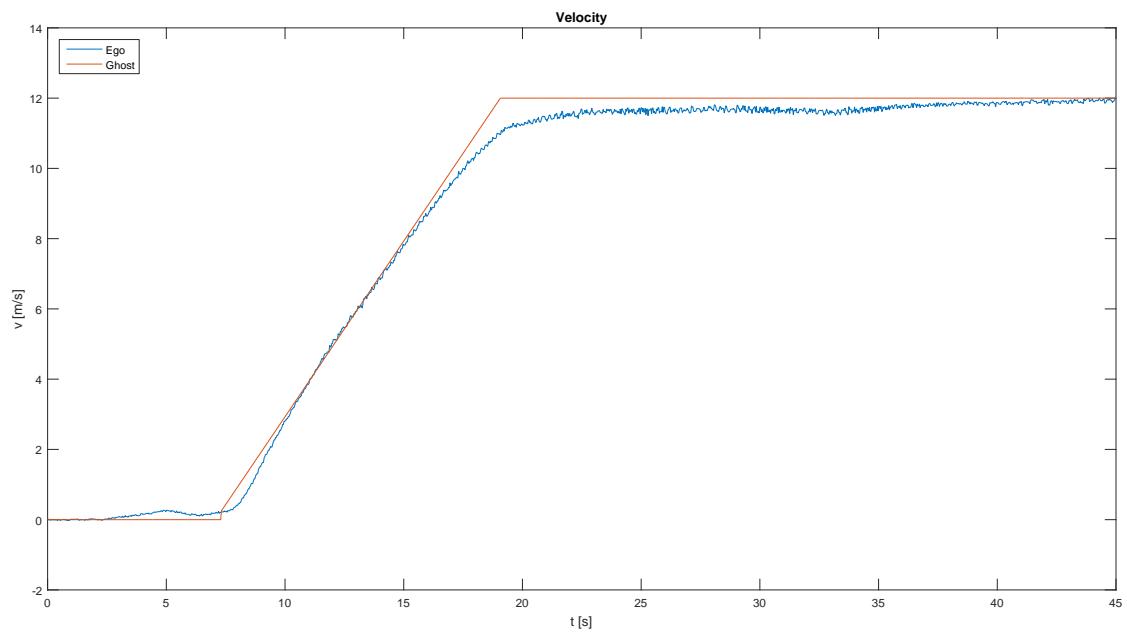


Figure 5.19: Velocity profile of the RCV following a speed of $12 \frac{m}{s}$ and a acceleration of $1 \frac{m}{s^2}$.

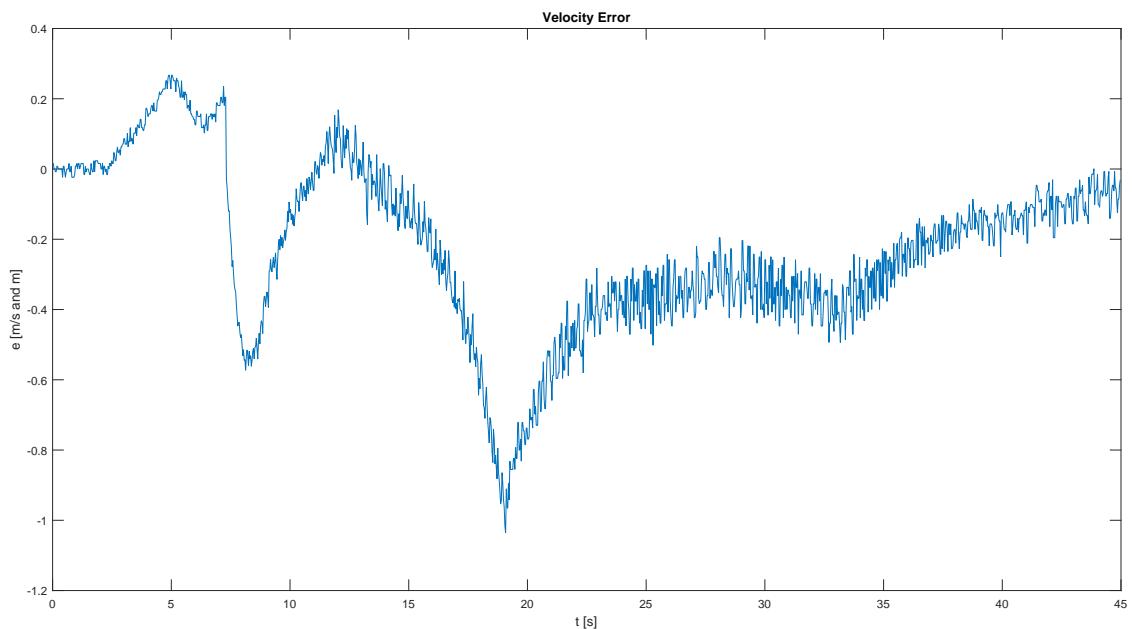


Figure 5.20: Difference between actual and desired values of the RCV following a speed of $12 \frac{m}{s}$ and a acceleration of $1 \frac{m}{s^2}$.

5.2 Lateral

Due to lack of time it was not possible to implement the lateral controller in the RCV. So to present some results, simulations are made to show the capabilities of the controller. In simulation everything is near perfect, which means that the model predicting the vehicle is the same model used in the controller. However, the prediction and the controller are sampled at different rates, the prediction is sampled every 0,01s and the controller every 0,03s.

Four different simulations to the controller are presented. In the first simulation, the vehicle starts perfectly positioned and changes lane while curving. In the second simulation, the vehicle starts with error being inside of the curve. The third simulation, is similar to the second but the start is outside of the curve. Finally, a simulation where the vehicle needs to crab is presented. These are made to show the capabilities of the controller in dealing with different situations and are presented in Figures 5.21, 5.22, 5.23 and 5.24, respectively. In these figures, the vehicle positions are represented by the red *xs* and the heading by the blue arrows.

Once again, these are simulations and the vehicle behaves exactly as expected. In Figure 5.21, the vehicle performs the lane change perfectly while doing a curve, where the outer lane has 20m of radius and the vehicle is traveling at a speed of $9\frac{m}{s}$. The longitudinal controller is also used to keep the speed for these simulations and the speed is the same for all.

Both simulations 5.22 and 5.23, include the correction of initial state errors. The first starts with an initial error of 10.35m on the lateral position of the vehicle in respect to the path and, the second, starts with an error of 14.65m. Since it is simulation, in both simulations the vehicle performed as expected, in the first simulation the vehicle reaches the path in 4s and in the second in 3.22s. Although, 5.23 has more initial error than 5.22, it takes less time to correct this error, because it is an easier and more direct motion to reach the path, so, the controller, ends up being more aggressive without causing any errors.

The last simulation 5.24, is made to show the capabilities of the controllers and how the crabbing motion looks like. The vehicle performs a diagonal practically without changing its heading. The initial and final changes in heading are due to constraints on the steering rate for the front and rear wheels, which makes it impossible for them to reach the required crabbing angle instantaneously, and, because of that, it needs some heading change to keep tracking the path.

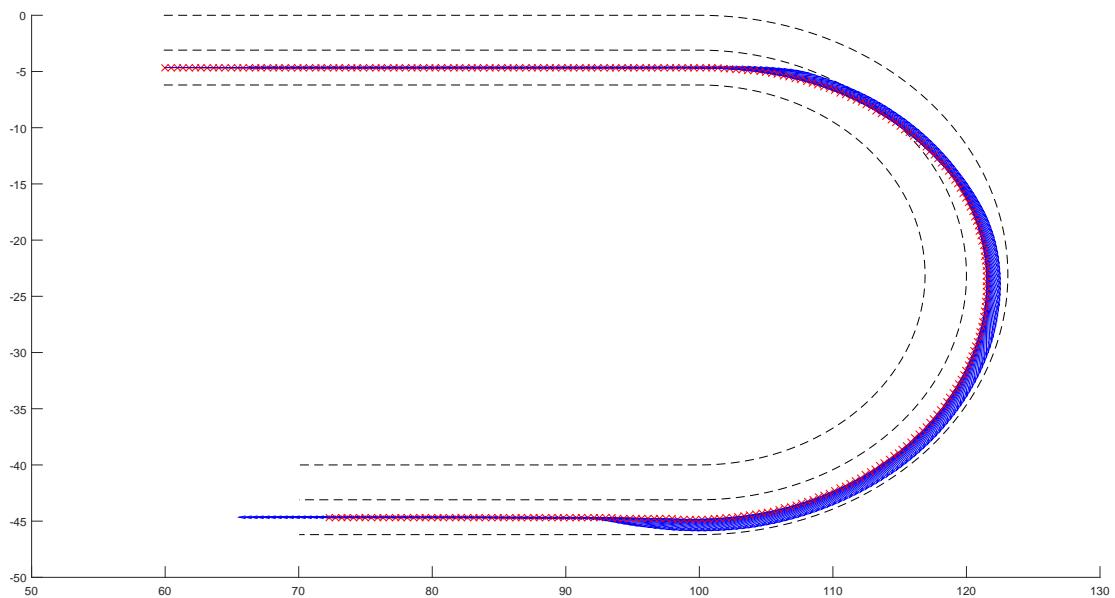


Figure 5.21: Path following and lane change while doing the curve - x in red represents the position of the vehicle and the blue arrow its direction.

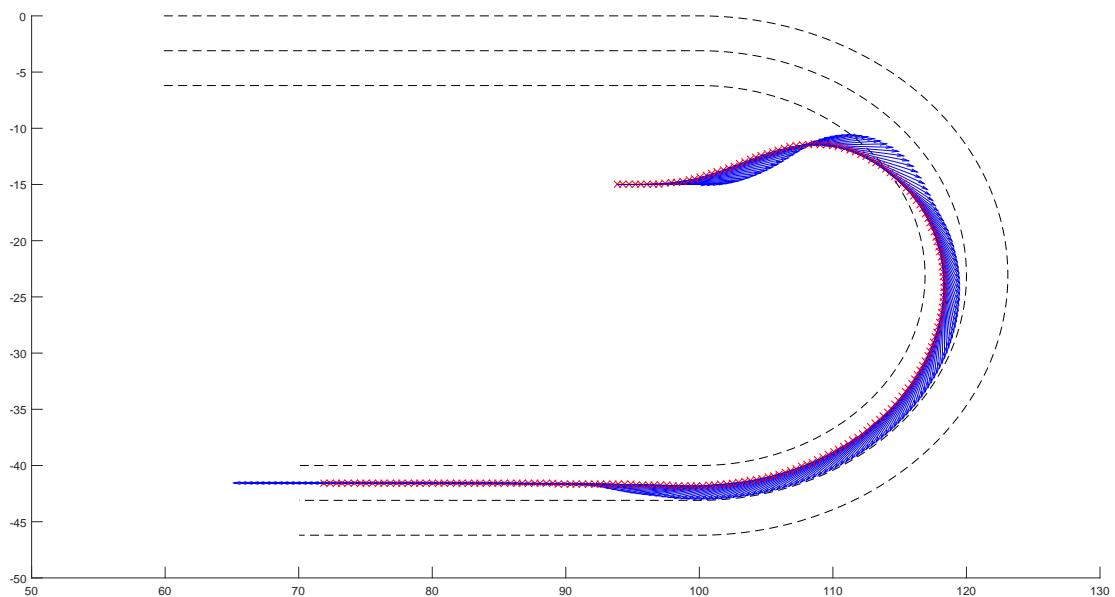


Figure 5.22: Path following and initial error correction - x in red represents the position of the vehicle and the blue arrow its direction.

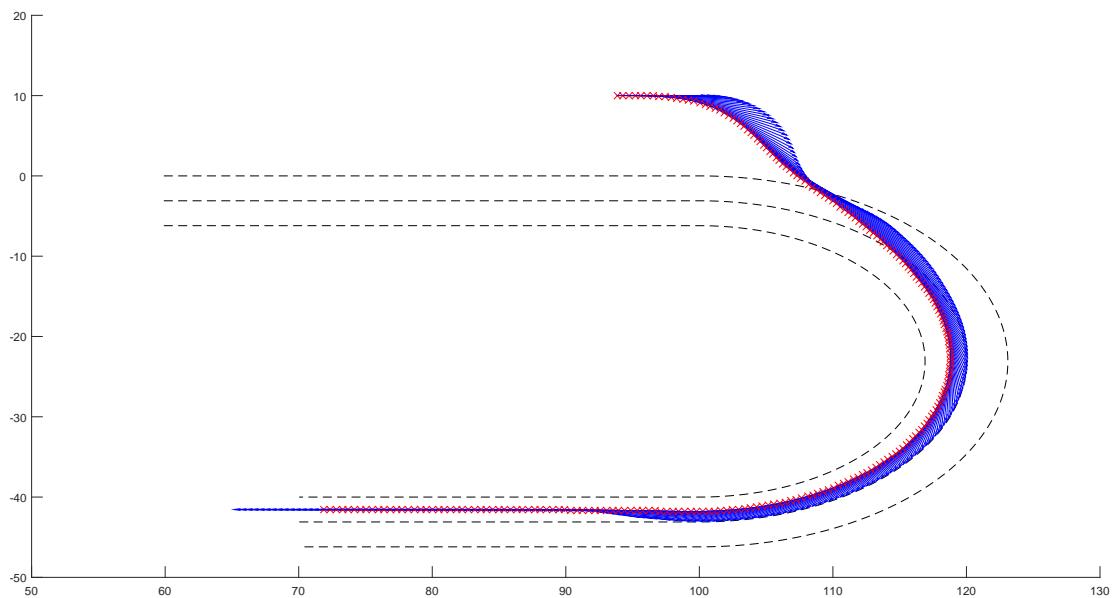


Figure 5.23: Path following and initial error correction - x in red represents the position of the vehicle and the blue arrow its direction.

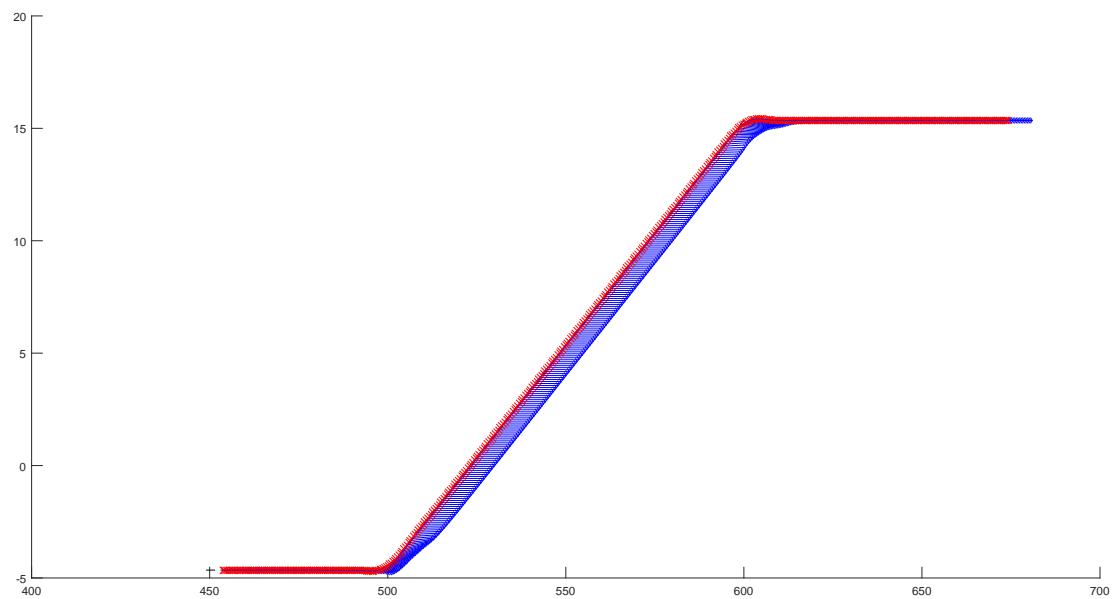


Figure 5.24: Path following by using crabbing - x in red represents the position of the vehicle and the blue arrow its direction.

Chapter 6

Conclusions

The main objective of this thesis was to make the RCV fully autonomous, by designing longitudinal and lateral controller, and, also, to participate in the GCDC 2016.

As mentioned before the participation was not possible, but since the minimum requirements to participate were to have longitudinal control, it is believed that if not for battery problems, the RCV would have been able to participate in all the scenarios and perform them. By looking at the different results presented, where both the distance and velocity errors were always small, it is possible to conclude that, from a controller point of view, the vehicle would be ready to compete.

The lateral controller, although not implemented in the RCV, was simulated and the results were very promising and show a clear use of the crabbing motion, which is one of the great advantages of the RCV over a normal vehicle and was the main challenge to be tackled by this controller.

The longitudinal controller, performed as expected, meeting all the requirements and being able to deal with all the scenarios proposed in the competition.

The main challenge of the thesis was the RCV, because, although it was physically available everyday, it was not available to be used or tested on. There was always something that needed some fixing or that needed to be checked, which, once again, made it impossible to finish everything proposed in the beginning of the thesis. About half of the time spent on this thesis, was spent trying to fix or figure out some problem with hardware or some other part of the system, where the main problem was the battery failing and burning its controller.

The hardware was very easy to use, given the direct connection to MATLAB, but it had major problems, for example, different rates on different ports cause the CPU to overload and crash without giving any further error messages. Also, the speedgoat CPU cannot handle a very high frequency of the controllers which would again cause a CPU overload. Visualization wise, it is a very limiting platform and even the logging of data is very weak, i.e., the Speedgoat does not allow to record everything one might want.

To conclude, the controllers were able to perform the tasks for which they were build while fulfilling all the specifications and constraints implemented.

Chapter 7

Future Work

The next step is to implement the lateral controller on the RCV, test it and tune it for the real vehicle, to be able to make the RCV fully autonomous.

The longitudinal controller can also be further tuned and changed. A major improvement to the controller, that would possibly make all the steady state errors disappear, even with a low battery level, is to add, to the error calculation, a P or PI controller, where this will further compensate for errors in the model used to estimate the future states of the vehicle. The same goes for the lateral controller if it proves necessary. To improve even further the longitudinal controller capabilities, one could also add a operation mode that enables trajectory tracking, to do this it is possible to use the CACC mode by changing the vehicle to be followed for points to be tracked, where the distance reference would always be 0, so the ego vehicle is on top of the point to track, and the velocity reference would be the velocity needed to go from one point to the next.

Another possible change is to upgrade the hardware, from the actuators to the Speedgoat and MABX. New hardware would not only allow the controllers to run faster and smoother, but also, it would make the whole system more stable and easier to control.

Finally, the software could be upgraded. One of the major problems in this thesis was the integration of the C-code with simulink, which would disappear if something like the robot operating system (ROS) [24] is used. The only problem with integrating everything in ROS is that, considering the RCV is a research platform where students working on it are constantly changing, it would make it harder for any student to just look at what is happening and build over it. Compared to Simulink, where it is really simple to understand how the system is connected and what inputs and outputs are needed, ROS has a more complex system that one needs to first learn and understand and, only after, it is easy to do work on top and integrate new things.

In a more general sense, cooperative driving as just started to see its first light. It will be a major improvement and it will help the everyday life of every single person. But, for this to be a reality, it is important that every vehicle cooperates with each other and with infrastructures. For that to happen, one can not only focus about its own autonomy, one has to focus and think about how the vehicles interact with each other and what they need from each other. So, in that sense, it is now important to focus on: *what is the vehicle going to do next?*. This will allow other vehicles and infrastructure to predict its movement and act accordingly, i.e., if the vehicle sends its future intentions it will improve further the benefits of autonomous vehicles. So, a possible next step of this research is

path and trajectory planning. Which was thought of, to some extent, and it was an input to the lateral controller, but always a very simple approach was taken which was only possible, because the road where the vehicle would be competing was known before hand.

For other vehicles to know the ego vehicle intentions, they have to be transmitted. Although, it was not the focus of this thesis, during the competition first hand experience was obtained on how difficult it is to make every different team to follow a protocol and also to make different hardware and equipment work together. So, another possible improvement of this thesis, would be to look into communication and how to integrate it and make it stable on the RCV.

Bibliography

- [1] Jonas Didoff. Specification of scenarios. URL <http://www.gcdc.net/en/teams/rules-and-technology>.
- [2] Hrishikesh Salunkhe, Bart Nijssen, and Jack Terken. Proposal for gcdc judging criteria and their evaluation. URL <http://www.gcdc.net/en/teams/rules-and-technology>.
- [3] B. van Arem, C. J. G. van Driel, and R. Visser. The impact of cooperative adaptive cruise control on traffic-flow characteristics. *IEEE Transactions on Intelligent Transportation Systems*, 7(4):429–436, 12 2006. ISSN 1524-9050. doi: 10.1109/TITS.2006.884615.
- [4] J. Ploeg, S. Shladover, H. Nijmeijer, and N. van de Wouw. Introduction to the special issue on the 2011 grand cooperative driving challenge. *IEEE Transactions on Intelligent Transportation Systems*, 13(3):989–993, 9 2012. ISSN 1524-9050. doi: 10.1109/TITS.2012.2210636.
- [5] Joel Vander Werf, Steven Shladover, Mark Miller, and Natalia Kourjanskaia. Effects of adaptive cruise control systems on highway traffic flow capacity. *Transportation Research Record: Journal of the Transportation Research Board*, 1800:78–84, 2002. doi: 10.3141/1800-10. URL <http://dx.doi.org/10.3141/1800-10>.
- [6] Darpa. URL <http://archive.darpa.mil/grandchallenge/index.html>.
- [7] Gcdc i-game. URL <http://www.gcdc.net/en/i-game>.
- [8] J. Ploeg, B. Scheepers, E. Van Nunen, N. Van de Wouw, and H. Nijmeijer. Design and experimental evaluation of cooperative adaptive cruise control. In *Intelligent Transportation Systems (ITSC), 2011 14th International IEEE Conference on*, pages 260–265, Oct 2011. doi: 10.1109/ITSC.2011.6082981.
- [9] Achour Amazouz. Optimal control for decentralized platooning, 2013.
- [10] P. Falcone, F. Borrelli, J. Asgari, H. E. Tseng, and D. Hrovat. Predictive active steering control for autonomous vehicle systems. *IEEE Transactions on Control Systems Technology*, 15(3):566–580, 5 2007. ISSN 1063-6536. doi: 10.1109/TCST.2007.894653.
- [11] I. Maurovic, M. Baotic, and I. Petrovic. Explicit model predictive control for trajectory tracking with mobile robots. In *Advanced Intelligent Mechatronics (AIM), 2011 IEEE/ASME International Conference on*, pages 712–717, 7 2011. doi: 10.1109/AIM.2011.6027140.

- [12] Dongbing Gu and Huosheng Hu. A model predictive controller for robots to follow a virtual leader. *Robotica*, 27:905–913, 10 2009. ISSN 1469-8668. doi: 10.1017/S0263574708005316. URL http://journals.cambridge.org/article_S0263574708005316.
- [13] F. Bu, H. S. Tan, and J. Huang. Design and field testing of a cooperative adaptive cruise control system. In *American Control Conference (ACC), 2010*, pages 4616–4621, 6 2010. doi: 10.1109/ACC.2010.5531155.
- [14] V. Turri, A. Carvalho, H. E. Tseng, K. H. Johansson, and F. Borrelli. Linear model predictive control for lane keeping and obstacle avoidance on low curvature roads. In *Intelligent Transportation Systems - (ITSC), 2013 16th International IEEE Conference on*, pages 378–383, 10 2013. doi: 10.1109/ITSC.2013.6728261.
- [15] F Kühne, J Gomes, and W Fetter. Mobile robot trajectory tracking using model predictive control. In *II IEEE latin-american robotics symposium*, 2005.
- [16] T. Keviczky, P. Falcone, F. Borrelli, J. Asgari, and D. Hrovat. Predictive control approach to autonomous vehicle steering. In *American Control Conference, 2006*, pages 6 pp.–, June 2006. doi: 10.1109/ACC.2006.1657458.
- [17] Petter Tomner. Design and implementation of control and actuation for an over-actuated research vehicle. Master’s thesis, KTH, Vehicle Dynamics, 2015.
- [18] A. Geiger, M. Lauer, F. Moosmann, B. Ranft, H. Rapp, C. Stiller, and J. Ziegler. Team annieway’s entry to the 2011 grand cooperative driving challenge. *IEEE Transactions on Intelligent Transportation Systems*, 13 (3):1008–1017, Sept 2012. ISSN 1524-9050. doi: 10.1109/TITS.2012.2189882.
- [19] Rajesh Rajamani. *Vehicle Dynamics and Control*.
- [20] J.A. Rossiter. *Model-Based Predictive Control: A Practical Approach*. Control Series. CRC Press, 2003. ISBN 9780203503966. URL <https://books.google.se/books?id=owznQTI-NqUC>.
- [21] Michael A. Henson. Nonlinear model predictive control: current status and future directions. *Computers and Chemical Engineering*, 23(2):187 – 202, 1998. ISSN 0098-1354. doi: [http://dx.doi.org/10.1016/S0098-1354\(98\)00260-9](http://dx.doi.org/10.1016/S0098-1354(98)00260-9). URL <http://www.sciencedirect.com/science/article/pii/S0098135498002609>.
- [22] J. Ploeg, N. van de Wouw, and H. Nijmeijer. Lp string stability of cascaded systems: Application to vehicle platooning. *IEEE Transactions on Control Systems Technology*, 22(2):786–793, March 2014. ISSN 1063-6536. doi: 10.1109/TCST.2013.2258346.
- [23] Cvxgen. URL <http://cvxgen.com/docs/index.html>.
- [24] The robot operating system. URL <http://www.ros.org/>.

Appendix A

A and B Matrices for Lateral Control

$$\begin{aligned}
A_{lat_i} = & \begin{bmatrix} 1 & 0 & -\Delta T (\dot{x}_i \sin(\psi_i) + \dot{y}_i \cos(\psi_i)) & 0 \\ 0 & 1 & \Delta T (\dot{x}_i \cos(\psi_i) - \dot{y}_i \sin(\psi_i)) & -\Delta T \sin(\psi_i) \\ 0 & 0 & 1 & \Delta T \cos(\psi_i) \\ 0 & 0 & 0 & \Delta T \end{bmatrix}, \\
& \begin{bmatrix} 0 & 0 & 0 & \Delta T \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & \Delta T \\ 0 & 0 & 0 & 0 \end{bmatrix} \\
& \begin{bmatrix} 1 + \Delta T \frac{\text{Aux}_1 \sin(\delta_{f_i}) + \text{Aux}_2 \sin(\delta_{r_i}) + \dot{x}_i C_d A_F}{M} & \Delta T \left(\frac{\text{Aux}_3 \sin(\delta_{f_i}) + \text{Aux}_4 \sin(\delta_{r_i})}{M} + \dot{\psi}_i \right) & \Delta T \left(\frac{\text{Aux}_5 \sin(\delta_{f_i}) + \text{Aux}_6 \sin(\delta_{r_i})}{M} + \dot{y}_i \right) \\ \Delta T \left(\frac{-\text{Aux}_1 \cos(\delta_{f_i}) - \text{Aux}_2 \cos(\delta_{r_i})}{M} - \dot{\psi}_i \right) & 1 + \Delta T \frac{-\text{Aux}_3 \cos(\delta_{f_i}) - \text{Aux}_4 \cos(\delta_{r_i})}{M} & \Delta T \left(\frac{-\text{Aux}_5 \cos(\delta_{f_i}) - \text{Aux}_6 \cos(\delta_{r_i})}{M} - \dot{x}_i \right) \\ \Delta T \frac{-a \text{Aux}_1 \cos(\delta_{f_i}) + b \text{Aux}_2 \cos(\delta_{r_i})}{I} & \Delta T \frac{-a \text{Aux}_3 \cos(\delta_{f_i}) + b \text{Aux}_4 \cos(\delta_{r_i})}{I} & 1 + \Delta T \frac{-a \text{Aux}_5 \cos(\delta_{f_i}) + b \text{Aux}_6 \cos(\delta_{r_i})}{I} \end{bmatrix} \quad (\text{A.1})
\end{aligned}$$

where,

$$\text{Aux}_1 = -2C_f \frac{\dot{y}_i + \dot{\psi}_i a}{(\dot{y}_i + \dot{\psi}_i a)^2 + \dot{x}_i^2}, \quad (\text{A.2a})$$

$$\text{Aux}_2 = -2C_r \frac{\dot{y}_i - \dot{\psi}_i b}{(\dot{y}_i - \dot{\psi}_i b)^2 + \dot{x}_i^2}, \quad (\text{A.2b})$$

$$\text{Aux}_3 = 2C_f \frac{\dot{x}_i}{(\dot{y} + a\dot{\psi}_i)^2 + \dot{x}_i^2}, \quad (\text{A.2c})$$

$$\text{Aux}_4 = 2C_r \frac{\dot{x}_i}{(\dot{y} - b\dot{\psi}_i)^2 + \dot{x}_i^2}, \quad (\text{A.2d})$$

$$\text{Aux}_5 = 2C_f \frac{a\dot{x}_i}{(\dot{y} + a\dot{\psi}_i)^2 + \dot{x}_i^2}, \quad (\text{A.2e})$$

$$\text{Aux}_6 = 2C_r \frac{-b\dot{x}_i}{\left(\dot{y} - b\dot{\psi}_i\right)^2 + \dot{x}_i^2}.$$

$$B_{lat_i} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \quad (A.3)$$

$$\begin{bmatrix} \Delta T \frac{\text{Aux}_7 \cos(\delta_{f_i}) - C_f \sin(\delta_{f_i})}{M} - \frac{2T_i \sin(\delta_{f_i})}{r} & \Delta T \frac{\text{Aux}_8 \cos(\delta_{r_i}) - C_r \sin(\delta_{r_i})}{M} - \frac{2T_i \sin(\delta_{r_i})}{r} \\ \Delta T \frac{\text{Aux}_7 \sin(\delta_{f_i}) + C_f \cos(\delta_{f_i})}{M} + \frac{2T_i \cos(\delta_{f_i})}{r} & \Delta T \frac{\text{Aux}_8 \sin(\delta_{r_i}) + C_r \cos(\delta_{r_i})}{M} + \frac{2T_i \cos(\delta_{r_i})}{r} \\ \Delta T \frac{a \text{Aux}_7 \sin(\delta_{f_i}) + a C_f \cos(\delta_{f_i}) + \frac{2aT_i \cos(\delta_{f_i})}{r}}{I} & \Delta T \frac{-b \text{Aux}_8 \sin(\delta_{r_i}) - b C_r \cos(\delta_{r_i}) - \frac{2bT_i \cos(\delta_{r_i})}{r}}{I} \end{bmatrix}$$

where,

$$\text{Aux}_7 = 2C_f \left(\arctan \left(\frac{\dot{y}_i + a\dot{\psi}_i}{\dot{x}_i} \right) - \delta_{f_i} \right), \quad (A.4a)$$

$$\text{Aux}_8 = 2C_r \left(\arctan \left(\frac{\dot{y}_i - b\dot{\psi}_i}{\dot{x}_i} \right) - \delta_{r_i} \right). \quad (A.4b)$$

TRITA EE 2016:118
ISSN 1653-5146