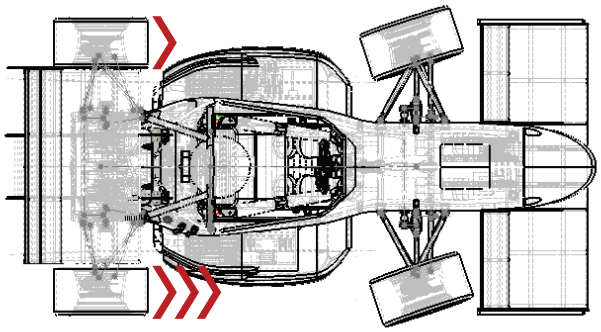
TÉCNICO LISBOA



Torque Vectoring for a Formula Student Prototype

João Pedro Marques Antunes

Thesis to obtain the Master of Science Degree in

Mechanical Engineering

Supervisors: Prof. Paulo Jorge Coelho Ramalho Oliveira
Prof. Carlos Baptista Cardeira

Examination Committee

Chairperson: Prof. Paulo Rui Alves Fernandes
Supervisor: Prof. Paulo Jorge Coelho Ramalho Oliveira
Member of the Committee: Prof. João Fernando Cardoso Silva Sequeira

June 2017

Acknowledgments

The work developed in this thesis, would not be possible without the help of my colleges from Projecto FST. With whom I spent great part of my life in the last years. Spent sleepless nights, discussed, learned, travelled, cheered, celebrated and shared the same passion for engineering and motorsports. I would like to thank specially João Paulo and Beatriz Astride for making it possible to test the controllers in the team's car, the FST06e. And Diogo Arreda for its help with the implementation of the C code in the car. Their help was invaluable.

To my thesis supervisor Professor Paulo Oliveira and co-supervisor Professor Carlos Cardeira, whose insight, knowledge and advice were equally important on pursuing this ambitious but highly enlightening challenge.

To my friends, for their support and everlasting friendship.

To my mother, my father, my brother and all my family, for their inestimable support, and always believing in me, not only during this work, but throughout the entire degree.

Resumo

Torque Vectoring (TV) tem como objectivo substituir a necessidade de um diferencial mecânico, ao mesmo tempo melhorando a resposta e manobrabilidade de um carro.

Este tese aboarda o projecto de um sistema de *torque vectoring* num protótipo *Formula Student* com tracção traseira. Propõe-se um controlador do tipo PI para o anel de controlo de rumo (*yaw rate*), com o objectivo de obter uma distribuição de torque igual para ambas as rodas. Um controlador LQR também é estudado, para o controlo de *yaw rate* e velocidade lateral.

Para avaliar os controladores, foi desenvolvido um modelo não linear de 7 graus de liberdade seguido de um modelo linear de 2 graus de liberdade. Os modelos foram ambos validados utilizando dados reais do protótipo. Estes controladores são depois discretizados e implementados no protótipo. Foram realizados testes para a avaliação do aumento de desempenho atingido pela adição do controlador.

Palavras-chave: Torque Vectoring, Formula Student, controlador PI, controlador LQR, dinâmica de veiculo

Abstract

Torque Vectoring (TV) has the objective to substitute the need of a mechanical differential, while also improving the handling and response of the Formula Student vehicle. This thesis addresses the design of a torque vectoring system in a rear wheel driven Formula Student prototype. The proposed solution resorts to a PI controller for yaw rate tracking with an evenly distributed torque to each wheel. Also an LQR scheme is discussed, for tracking the yaw rate and the lateral velocity. To assess and design, first a 7 degree of freedom (DOF) non linear model is constructed, followed by a linear 2 DOF model, both models are validated with real data. The linear model, is used to design and simulate the proposed controllers. When the controller is within the desired parameters it is tested in the non linear model. Tests with the vehicle are performed to verify the contribution of the controller to the overall performance of the vehicle.

Keywords: Torque Vectoring, Formula Student, PI controller, LQR controller, vehicle model

Contents

List of Tables

List of Figures

Nomenclature

α	Tyre slip angle
δ	Steering angle
δ_l	Steering angle of left wheel
δ_r	Steering angle of right wheel
$\dot{\omega}$	Angular acceleration
$\dot{\psi}_{des}$	Desired yaw rate
μ_l	Longitudinal road surface adhesion coefficient
μ_r	Resultant road surface adhesion coefficient
μ_s	Lateral road surface adhesion coefficient
ω	Angular velocity
ψ	Yaw moment
a_x	Longitudinal acceleration of the vehicle
a_y	Lateral acceleration of the vehicle
$C_{1,2,3}$	Tire constant
$C_{y,f}$	Cornering stiffness of the front axle
$C_{y,r}$	Cornering stiffness of the rear axle
CG	Center of gravity
F_x	Longitudinal force acting on the vehicle
F_y	Lateral force acting on the vehicle
F_z	Vertical force
F_{drag}	Drag force
F_{roll}	Wheel's rolling resistance

g	Gravity
G_r	Gear ratio
h_{CG}	Center of gravity height
I_w	Moment of inertia of the wheel around the turning axis
I_{zz}	Moment of inertia around the vertical vehicle axis
l	Wheelbase of vehicle
l_f	Distance between the CG and the front axle
l_r	Distance between the CG and the rear axle
m	Mass of the vehicle
M_z	Yaw moment around the vertical axis of the vehicle
R_w	Effective tyre radius
s_l	Lateral wheel slip
s_l	Longitudinal wheel slip
s_r	slip ratio
T	Torque
t_f	Distance between front wheels
t_r	Distance between rear wheels
v_x	Longitudinal velocity of the vehicle
v_y	Lateral velocity of the vehicle
w_x^F	Longitudinal force in the wheel acting on the vehicle
w_y^F	Lateral force in the wheel acting on the vehicle
w_x^v	Longitudinal velocity in the wheel coordinate system
w_y^v	Lateral velocity in the wheel coordinate system
fl	Front left wheel
fr	Front right wheel
rl	Rear left wheel
rr	Rear right wheel
x, y, z	Cartesian components
I,B,W	Coordinate system components

Glossary

ABS	Anti-lock Braking System
ADC	Analogue to Digital Converter
CAN	Communications Area Network
ESP	Electronic Stability Program
FST06e	Sixth car of the formula student team
FS	Formula Student
K&C	Kinematic and Compliance test
LQR	Linear Quadratic Regulator
PID	Proportional Integral Derivative controller
TTC	Tire Testing Consortium
TV	Torque Vectoring

Chapter 1

Introduction

Each year that goes by sees an increase in sales of personal use of electric vehicles (battery EVs, plug-in hybrids and regular hybrids), all of which rely on electric motors as a basis or an aid to propulsion. With this increase, manufacturers are starting to explore new ways of implementing electric motors. They are favoring the use of 2 or 4 motors instead of just a single motor. Using more motors is advantageous: it gives a lower center of gravity (CG), and are lighter when compared to a combustion engine. It also opens the opportunity of vehicle stability control systems directly at the motors, like Electronic Stability Program (ESP) and Anti-lock Braking System (ABS) which gives the vehicle better stability and manoeuvrability. This thesis addresses another type of vehicle control solution called Torque Vectoring (TV), which is especially suited for electric vehicles. By controlling the amount of torque distributed to each driven wheel. The system has the potential to improve both the stability and response of the vehicle without compromising safety and drivability.

It is known that the response of a vehicle to a steering input is relatively slow and imprecise. Moreover some deformation of the tires occur before generating enough lateral force to steer the vehicle. The principle of torque vectoring is to improve the response of the vehicle to a steering input. By controlling the driving and braking torque between the left and right driven wheels a yaw moment (torque) can be created with the goal of controlling the yaw rate of the vehicle.

Torque vectoring has made a big impact on Formula Student events. Since its appearance in 2011, electric cars have consistently been in the top position every year. 2016 FS competition once again showed that a four wheel drive electric vehicle is the winning concept.

1.1 Motivation

Formula Student is an engineering competition between university teams from around the globe. Students have to design and build their own race car prototype to compete in a series of static and dynamics events, evaluated by judges from both professional motorsport and automotive industry. The static events are where the team defends its design, manufacturing, cost and business plan, while in the dynamic events, the performance of the car is evaluated in a series of tests: skid-pad, 0-75m acceleration, auto-cross, endurance and efficiency.

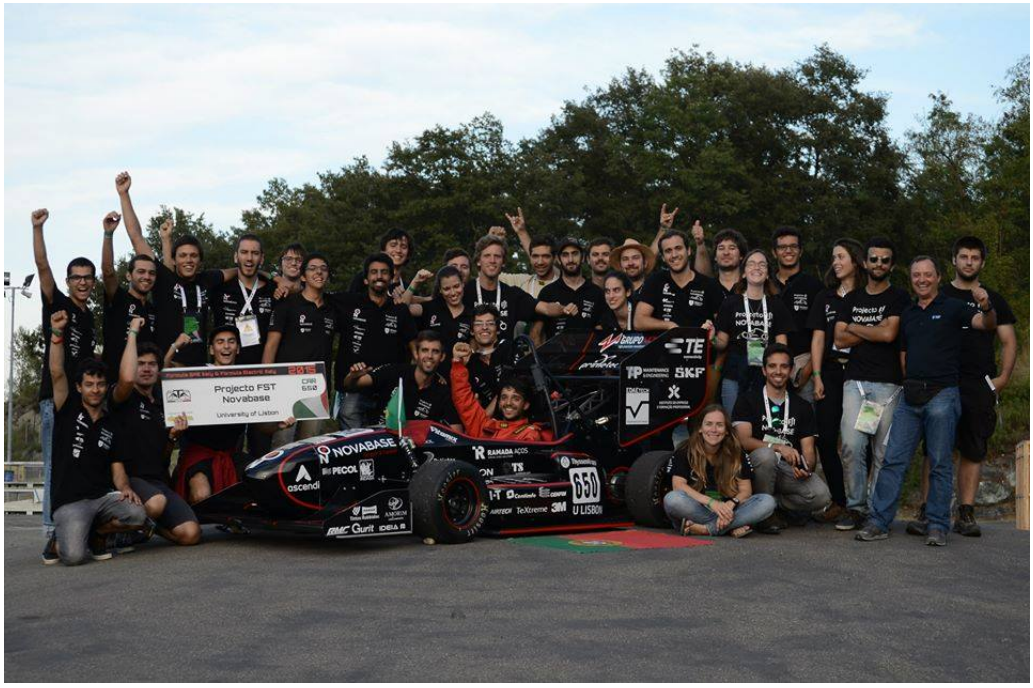


Figure 1.1: FST06e team at Formula Student Italy 2015

Projecto FST Novabase (FST) is a student team from Instituto Superior Técnico (IST). The team designs and builds electric cars. Currently the 4th electric car, the FST07e prototype, is under development.

To be possible for the team to compete successfully against the top teams, it is necessary that the next prototype (FST07e) be a four wheel drive electric vehicle (4WD-EV). The problem is that from the time that the car is built until the time of competition, the test window is very short. To address this problem, the team uses the previous prototype, the FST06e (a two wheel rear driven car) as the test car for the development and implementation of the innovations, before implementing the system for the FST07e.

1.2 Projecto FST Novabase: FST06e

The FST06e was designed and built during 2014-2015, by a team of 40 enthusiastic people. It is the 3rd electric car of the project. It is powered by two *Siemens* permanent magnet synchronous motors, each one with an RPM range from 0 to 8000, and produces a maximum torque of 107 Nm. With a planetary gear set fixed with a gear ratio of 4.1:1, torque at the wheel is increased to a total of 876 Nm.

Figure 1.2: FST06e during autocross at formula student Czech Republic

1.3 Work Contributions

The following contributions are outlined:

- A self-developed vehicle model with 7 degrees of freedom (DOFs) was built for the design of the torque vectoring system and simulation. The complete model, which is described in Chapter 2, includes the horizontal dynamics, for vehicle position and orientation, a simple tire model, load transfer, steering Ackerman, longitudinal and lateral forces. The model works for any Formula Student vehicle, currently loaded with the values of the FST06e.
- A 2 DOFs model contemplating the vehicle lateral motion. This model is modified to contemplate the additional yaw moment provided by the different torque at each wheel, also added the relation between the torque at the wheel and the motor for the FST06e
- The proposed PID control is a look-up table with a set of gains for a range of velocities of the vehicle. The controller tracks the yaw rate and based on the difference between the desired and current yaw rate a torque difference between the left and right wheel is applied.

- A LQR control system for the yaw rate tracking and lateral velocity was modelled and integrated in the vehicle model.
- A linear model in discrete time in Simulink is developed. This simulation allows the input of real data from the vehicle to test the controllers.
- A combination of a 3 point median and low pass filter is designed for filtering the yaw rate values from the inertial measurement unit.

1.3.1 Contributions to the Team

The team develops its own software and hardware. Being the first time implementing a control strategy the team no previous software was available. For that reason a set of tools were developed to aid in the design, validation and implementation of the controllers:

- A set of constant radius circle tests are performed to validate both models and sensors used in the vehicle. The velocity and steering angle are the inputs, the outputs are the longitudinal, lateral acceleration and the yaw rate.
- A Matlab script is developed to process the data values logged from the vehicle. This script separates the data from each sensor, re-sampling and presenting to the user in the correct time instant, in order to correlate all the data with respect to time.
- A Matlab script was created which allowed CAN communication between the FST06e and Matlab environment. This permitted an open loop real time communication for simulating real data from the vehicle. This permitted the testing of the developed C code in the development board. Variables from the car (steering angle, velocity and yaw rate) where fed to the simulation, and observed it affected the controller and the calculation of the reference. Also it allowed the testing of the fail safe system and filters.
- Implementation of the C code was also developed, with the development board made by the team. A workflow is proposed. Divided in 3 main parts, receiving data, algorithm, sending data. The code is design thinking in further applications, where it is only necessary to change the algorithm part.
- Based on the experienced gathered from testing the controller on the vehicle, a set of fail safe systems were implemented in the vehicle. These fail safes helped to increase the safety of the team when testing the controllers.

1.4 Objectives

A comprehensive quantitative test of the control system developed in the thesis can only be performed with access to the FST06e, by a dedicated team with a minimum number of six people over several

days. With the team completing their regular tasks for the upcoming car, it would be difficult for them to find the time to prepare and maintain the previous car, with which the tests would be performed. Having this in mind the objective of this thesis is to:

- Develop the team's knowledge on the design and implementation of control systems, by providing a complete workflow from design to implementation to validation of the controller.
- Increase the lateral performance of the prototype, measured in terms of lap times.

1.5 Thesis Outline

Chapter 2 begins with the introduction of the models used through this thesis. It starts by defining the coordinate system of the vehicle, followed by the basic physical relations and equations of the vehicle movement, especially the lateral dynamics. The model is then linearized and the necessary equations are added to take into account the dynamics related to the torque. The chapter concludes with simulations that compare both controllers using the linear and nonlinear models. A simple test track is mounted and important parameters of the car are logged and then compared with the simulation for the same inputs.

Chapter 3 discusses the proposed controllers, starting with an analysis on the available sensors in the FST06e. Based on the sensors, the computation of the reference yaw rate is proposed. An analysis of the system response is then carried out. Furthermore, the controller is implemented in Matlab simulink. Same approach is used in the design of the LQR controller. The chapter then concludes with a comparison between both controllers in the linear simulation and in the non linear simulation.

In chapter 4, the linear model is discretized to allow the development of the proposed controllers in discrete time. Simulation with real data (sampled data and noise) is performed.

Chapter 5 describes the controller implementation in C code on the FST06e. The chapter starts with and introduction on the communications protocol used in the vehicle, from which a simulink model that allows the communication between the vehicle and the controller board is developed, allowing the test of the developed code in hardware. It was possible to test the code, filter data and validate the fail safe systems.

The thesis concludes with chapter 6 where the proposed controllers are tested in the real prototype and the performance of the vehicle is assessed with and without the controllers.

Chapter 2

Vehicle Dynamics

Vehicle dynamics is the area devoted to the development of models that describe the behaviour of a vehicle for any given set of inputs and disturbances. Modelling this type of system is a very complex and challenging task. A lot of different approaches and models can be used depending on the needs of the user. A complex multi-body system (with 20+ degrees of freedom), or a simple two degree of freedom model (with 2 degrees of freedom) [1, p.6].

The vehicle model used to study a torque vectoring control system will typically have seven degrees of freedom. The lateral and longitudinal velocities of the vehicle (v_x and v_y respectively) and the yaw rate $\dot{\psi}$ are three degrees of freedom related to the vehicle body. The wheel velocities of the four wheels, the front left wheel (w_{fl}), front right wheel (w_{fr}), rear left wheel (w_{rl}) and rear right wheel (w_{rr}) constitute the other four degrees of freedom [2].

The model is divided in two parts. In the case where more information is available (parametric values and/or behaviour), a more accurate and complex model is used, otherwise a more simplified approach is applied (ex: tire model). This is intended to reduce complexity and provide a better understanding of the dynamics. Although this model is only valid in some range of velocities, it is easier to start working with a simpler and understandable model than a complex one.

The basic principles required to model a 7 degree of freedom model will be detailed in this chapter. After a brief description of the coordinate system used in section 2.1, section 2.2 offers an overview of a vehicle kinematics, followed by geometrical relations between the wheels and the vehicle velocity. Section 2.3 details the relation between the wheels and steering wheel used in the FST06e. After, in section 2.4 it is discussed how the longitudinal and lateral forces are generated at the wheels and how they are passed to the vehicle. Then, all relations from the previous sections are put together in section 2.5, in a balance of forces and moments. In section 2.7, the linearization of the non linear model is presented. This chapter then concludes in section 2.8 with a validation of the presented models based on real data from the FST06e.

2.1 Vehicle Model

To obtain the vehicle dynamics, the vehicle is simplified to a mass m at the center of gravity (CG) and a moment of inertia I_{zz}. To describe the model, 3 cartesian coordinate frames are necessary:

The inertial coordinate frame, denoted by the upper-script I, is a fixed frame with the origin placed at any point and the orientation that is most convenient to the user. This frame is useful when analyzing the trajectory of the vehicle, which is the path taken from an initial time to a final time seen by an observer in a static point.

The body fixed coordinate system, with the upper-script B, has the origin in the center of mass of the vehicle. The x-axis is in the longitudinal direction of the vehicle, the y-axis is towards the left of the vehicle and the z-axis points upwards. The rotations about the x,y,z-axis are called roll, pitch and yaw, respectively.

The wheel fixed coordinate system, denoted by the upper-script W, with the origin at the center of each wheel. The x,y,z-axis have the same direction and orientation as the body fixed system. The vehicle in study has 4 wheels, so there will be a fixed coordinate system attached to each wheel.

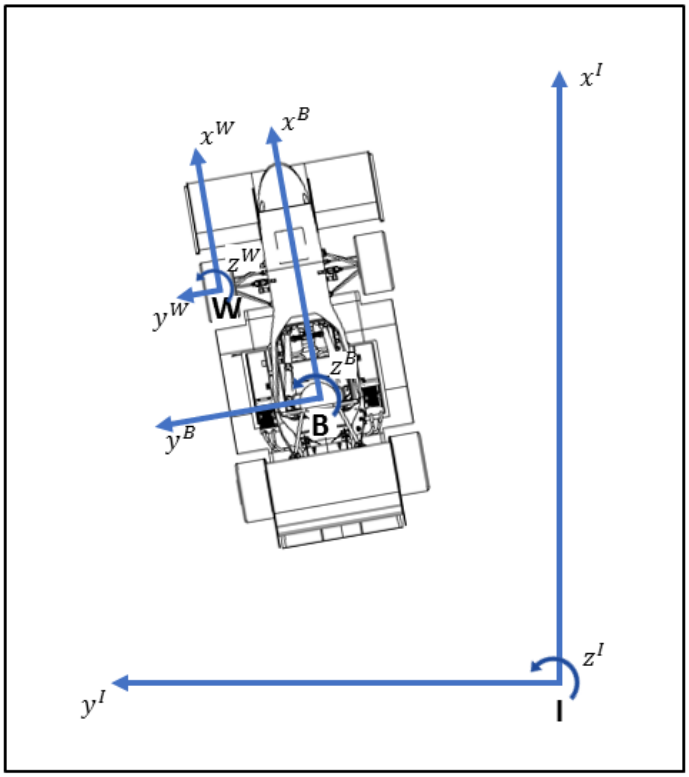


Figure 2.1: Representation of inertial (I), body (B) and wheel (W) coordinate frames

2.2 Vehicle Trajectory

The previous defined coordinate frames will be used throughout this thesis. The transformation from the vehicles' frame to the inertial frame (${}^I R_B$), the Euler angle rotation matrix is used [3, p. 233]. Note that

we are only interested in studying the movement in the xy plane, which means that null roll and pitch are assumed, simplifying the rotation matrix:

$$ {}^{I}R_{B} = \begin{bmatrix} \cos(\psi) & -\sin(\psi) & 0 \\ \sin(\psi) & \cos(\psi) & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{2.1} $$

The path of the vehicle will be obtained by the integration of the vehicle's velocity ${}^{B}v$ with respect to time, and rotated to the inertial frame ${}^{I}v$, by a yaw angle ($\dot{\psi}$).

$$ {}^{I}v = {}^{I}\ R_{B}\ {}^{B}v = \begin{bmatrix} {}^{B}\dot{x}\cos(\psi) - {}^{B}\ \dot{y}\sin(\psi) \\ {}^{B}\dot{x}\sin(\psi) + {}^{B}\ \dot{y}\cos(\psi) \\ 0 \end{bmatrix} \tag{2.2} $$

The position of the car is therefore defined by the previous position (x_0) and the distance covered in the x and y direction between time t_0 and t:

$$ {}^{I}x = {}^{B}\ x_0 + \int_{t_0}^{t} ({}^{B}\dot{x}\cos(\psi) - {}^{B}\ \dot{y}\sin(\psi))dt \tag{2.3} $$

$$ {}^{I}y = {}^{B}\ y_0 + \int_{t_0}^{t} ({}^{B}\dot{x}\sin(\psi) + {}^{B}\ \dot{y}\cos(\psi))dt \tag{2.4} $$

The rotation of the car can be described by the previous rotation summed with the rotation in the same time interval

$$ \psi(t) = \psi_0 + \int_{t_0}^{t} \dot{\psi}dt \tag{2.5} $$

2.3 Steering Kinematics

In order for a vehicle to corner, the geometric center of the vehicle's path of curvature must be located on an extension of the line of the vehicle's rear axle. If it isn't the case, that would mean that the rear tires must slip, because of the front track width (t_f) the front wheels follows a different curvature radius [4]. Figure 2.2 illustrates this case.

2.3.1 Ackerman's Geometry

To counter this effect, the majority of vehicles, including the FST06e, uses the Ackerman geometry. The driver can only input one steering angle δ. This angle is then transformed into two steering angles, one for each wheel, δ_l for the left wheel and δ_r for the right wheel [3, Chapter 7].

Term	Symbol	Value	Units
Left wheel steering angle	δ_l	-	$[rad]$
Right wheel steering angle	δ_r	-	$[rad]$
steering angle	δ	[-3.3,3.3]	$[rad]$
Front track	t_f	0.62	$[m]$
Wheelbase	l	1.29	$[m]$

Table 2.1: Table with the symbols and values for the steering geometry from the FST06e

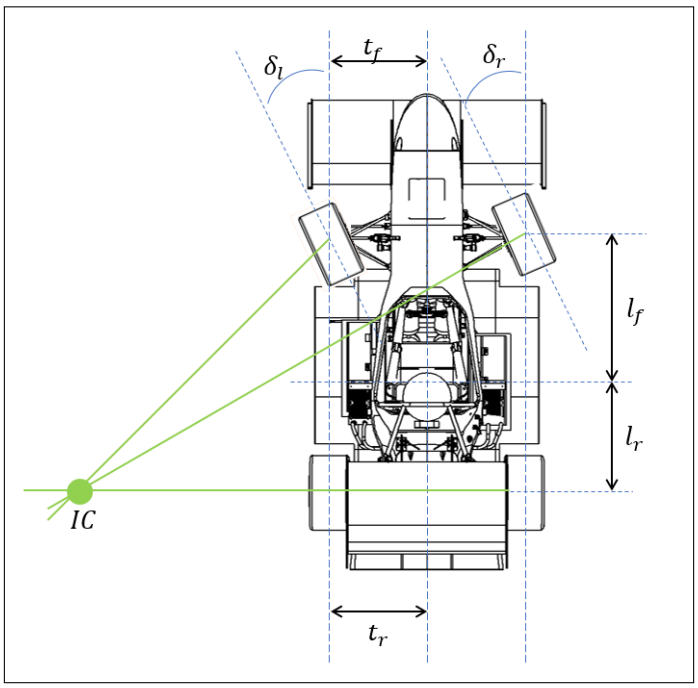


Figure 2.2: Instantaneous center of rotation (IC), with pro-ackerman geometry

$$\delta_l = \tan^{-1}\left(\frac{l}{lcot(\delta) - \frac{t_f}{2}}\right) \tag{2.6}$$

$$\delta_r = \tan^{-1}\left(\frac{l}{lcot(\delta) + \frac{t_f}{2}}\right) \tag{2.7}$$

Another, advantage of this type of system, is that given a steering angle δ, based on the length l and front track t_f it is possible to know how much each wheel has turned, and calculate their respective turning radius.

$$R = \sqrt{\left(\frac{t_f}{2}\right)^2 + l^2 cot^2 \delta} \tag{2.8}$$

2.4 Wheel Velocity Vectors and Slip Angles

The velocities of the wheels ground contact point are determined by a transformation of the velocity of the car to the wheels. Assuming that the velocity of the vehicle can be described as a superposition of pure translatory motion with magnitude and direction v_{CG}, and a purely rotational motion with yaw rate $\dot{\psi}$ around the center of gravity [5, p.304-313].

2.4.1 Velocity Vectors

A division into longitudinal and lateral vehicle directions gives the velocity in each wheel v^B, where the first term is the longitudinal velocity and the second term the lateral velocity for each wheel.

$$\begin{aligned}
v_{fl}^B &= \sqrt{\left(v_x - \dot{\psi} R_{fl} \sin \epsilon_{fl}\right)^2 + \left(v_y + \dot{\psi} R_{fl} \cos \epsilon_{fl}\right)^2} \\
v_{fr}^B &= \sqrt{\left(v_x + \dot{\psi} R_{fl} \sin \epsilon_{fl}\right)^2 + \left(v_y + \dot{\psi} R_{fl} \cos \epsilon_{fl}\right)^2} \\
v_{rl}^B &= \sqrt{\left(v_x - \dot{\psi} R_{fl} \sin \epsilon_{fl}\right)^2 + \left(v_y - \dot{\psi} R_{fl} \cos \epsilon_{fl}\right)^2} \\
v_{rr}^B &= \sqrt{\left(v_x + \dot{\psi} R_{fl} \sin \epsilon_{fl}\right)^2 + \left(v_y - \dot{\psi} R_{fl} \cos \epsilon_{fl}\right)^2}
\end{aligned} \tag{2.9}$$

The distance from the center of mass to each of the wheels is given by:

$$\begin{aligned}
R_{fl} &= \sqrt{t_f^2 + l_f^2} \qquad & R_{fr} &= \sqrt{t_f^2 + l_f^2} \\
R_{rl} &= \sqrt{t_r^2 + l_r^2} \qquad & R_{rr} &= \sqrt{t_r^2 + l_r^2}
\end{aligned} \tag{2.10}$$

The angles between the wheel-ground contact point and the undercarriage axis are:

$$\begin{aligned}
\epsilon_{fl} &= tan^{-1}\left(\frac{t_f}{l_f}\right) \qquad & \epsilon_{fr} &= tan^{-1}\left(\frac{l_f}{t_f}\right) \\
\epsilon_{rl} &= tan^{-1}\left(\frac{l_r}{t_r}\right) \qquad & \epsilon_{rr} &= tan^{-1}\left(\frac{t_r}{l_r}\right)
\end{aligned} \tag{2.11}$$

2.4.2 Side slip angle

The instant that a tire develops a slip angle α and in order to develop cornering force any tire must develop a slip angle [4]. The side slip is defined as the angular displacement between the plane of rotation of the wheel and the path that the tire will follow on the road. Using trigonometric relations the side slip angle of the vehicle can be calculated as [5, p.315]:

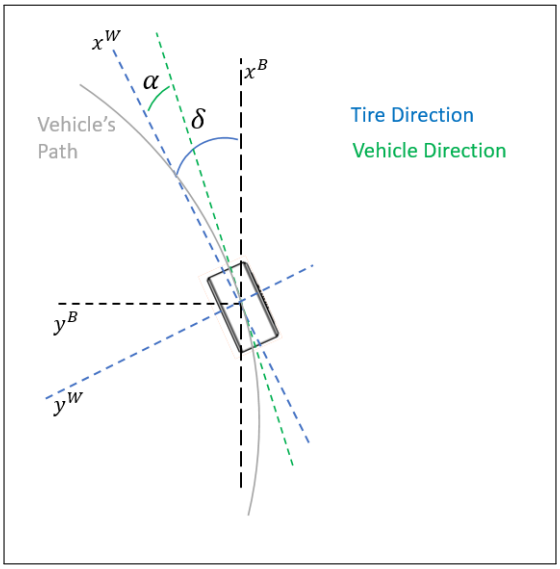


Figure 2.3: Side slip angle developed by the vehicle in a left corner. The trajectory of the car is changed

$$
\begin{aligned}
\alpha_{fl} &= \delta_l - \tan^{-1}\left(\frac{v_y + \dot{\psi} R_{fl}\cos\epsilon_{fl}}{v_x - \dot{\psi} R_{fl}\sin\epsilon_{fl}}\right) \\
\alpha_{fr} &= \delta_r - \tan^{-1}\left(\frac{v_y + \dot{\psi} R_{fr}\sin\epsilon_{fr}}{v_x + \dot{\psi} R_{fr}\cos\epsilon_{fr}}\right) \\
\alpha_{rl} &= -\tan^{-1}\left(\frac{v_y - \dot{\psi} R_{rl}\sin\epsilon_{rl}}{v_x - \dot{\psi} R_{rl}\cos\epsilon_{rl}}\right) \\
\alpha_{rr} &= -\tan^{-1}\left(\frac{v_y - \dot{\psi} R_{rr}\sin\epsilon_{rr}}{v_x + \dot{\psi} R_{rr}\sin\epsilon_{rr}}\right)
\end{aligned}
\tag{2.12}
$$

Note that the rear side slip angle α_{rl}, α_{rr} does not have the steering component, because this model is only front wheel steered.

2.4.3 Slip Ratio

The ratio between the vehicle and it's wheels is called slip ratio. The longitudinal slip s_l is defined in the direction of the wheel velocity $v_{i,j}$. Lateral slip s_s is defined perpendicular to the wheel velocity. The wheel slip is always between -1 and 1, this is obtained by dividing for the respective larger speed, hence $v_{i,j}$ for braking and $w_{i,j}R_w\cos(\alpha_{ij})$ for driving [5, 1, p.315,p.305].

Term	Symbol	Value	Units
Longitudinal slip	s_l	-	-
Lateral slip	s_s	-	-
Side slip angle	α	-	$[rad]$
Wheel velocity	ω	-	$[rads^{-1}]$
Radius of wheel	R_w	0.228	$[m]$
Linear velocity of wheel	v	-	$[ms^{-1}]$

Table 2.2: Table with the terms and values for calculating the slip ratios

The longitudinal and lateral slip, which are summarized in table 2.3, are divided into braking and accelerating. Braking slip occurs when the car is decelerating, so the velocity of the wheels will be smaller than the velocity of the vehicle. When the driver is accelerating the velocity of the wheels will be bigger than the velocity of the vehicle.

Longitudinal slip		Lateral slip	
Braking	Acceleration	Braking	Acceleration
$s_{l,fl} = \frac{w_{fl}R_{w,fl}\cos\alpha_{fl} - v_{fl}}{v_{fl}}$	$s_{l,fl} = \frac{w_{fl}R_{w,fl}\cos\alpha_{fl} - v_{fl}}{w_{fl}R_{w,fl}\cos\alpha_{fl}}$	$s_{s,fl} = \frac{w_{fl}R_{w,fl}\sin\alpha_{fl}}{v_{fl}}$	$s_{s,fl} = \tan(\alpha_{fl})$
$s_{l,fr} = \frac{w_{fr}R_{w,fr}\cos\alpha_{fr} - v_{fr}}{v_{fr}}$	$s_{l,fr} = \frac{w_{fr}R_{w,fr}\cos\alpha_{fr} - v_{fr}}{w_{fr}R_{w,fr}\cos\alpha_{fr}}$	$s_{s,fr} = \frac{w_{fr}R_{w,fr}\sin\alpha_{fr}}{v_{fr}}$	$s_{s,fr} = \tan(\alpha_{fr})$
$s_{l,rl} = \frac{w_{fl}R_{w,rl}\cos\alpha_{rl} - v_{rl}}{v_{rl}}$	$s_{l,rl} = \frac{w_{rl}R_{w,rl}\cos\alpha_{rl} - v_{rl}}{w_{rl}R_{w,rl}\cos(\alpha_{rl})}$	$s_{s,rl} = \frac{w_{rl}R_{w,rl}\sin\alpha_{rl}}{v_{rl}}$	$s_{s,rl} = \tan(\alpha_{rl})$
$s_{l,rr} = \frac{w_{rr}R_{w,rr}\cos\alpha_{rr} - v_{rr}}{v_{rr}}$	$s_{l,rr} = \frac{w_{rr}R_{w,rr}\cos\alpha_{rr} - v_{rr}}{w_{rr}R_{w,rr}\cos(\alpha_{rr})}$	$s_{s,rr} = \frac{v_{R,rr}R_{w,rr}\sin\alpha_{rr}}{v_{rr}}$	$ss_{s,rr} = \tan(\alpha_{rr})$

Table 2.3: Longitudinal and lateral slips for braking and acceleration

The resultant slip ratio is the geometrical sum of the longitudinal and lateral slip.

$$\begin{aligned} s_{r,fl} &= \sqrt{s_{l,fl}^2 + s_{s,fl}^2} & s_{r,fr} &= \sqrt{s_{l,fr}^2 + s_{s,fr}^2} \\ s_{r,rl} &= \sqrt{s_{l,rl}^2 + s_{s,rl}^2} & s_{r,rr} &= \sqrt{s_{l,rr}^2 + s_{s,rr}^2} \end{aligned} \tag{2.13}$$

2.4.4 Friction Coefficient

The friction coefficient can be calculated using empirical models, it was chosen to use one of the most common, the Burkhardt Model. This model was chosen, because no processed tire data was available for the tires equipped on the FST06e, and deriving a more complex model with the tire data would be too time consuming. Instead it was opted for the Burkhardt model, which takes into account the influence of the velocity, but there is no dependency in the tire vertical load F_z [5, 3, p.319,p.315].

$$\mu_r(s_r) = C_1(1 - e^{-C_2 s_r}) - C_3 s_r \tag{2.14}$$

Where depending on the type of surface the constant (C_1, C_2, C_3) will vary.

	C_1	C_2	C_3
Asphalt, dry	1.2801	23.99	0.52
Asphalt, wet	0.857	33.822	0.347
Concrete, dry	1.1973	25.168	0.5373
Concrete, wet	0.4004	33.7080	0.1204
Snow	0.1946	94.129	0.0646
Ice	0.05	306.39	0

Table 2.4: Tire-Road Constants [5, p.322]

From the Burkhardt equation it is possible to obtain the resultant friction and from that, calculate the longitudinal and lateral friction in each wheel based on the slip ratios.

Longitudinal friction	Lateral friction
$\mu_{l,fl} = \mu_{r,fl}\frac{s_{l,fl}}{s_{r,fl}}$	$\mu_{s,fl} = \mu_{r,fl}\frac{s_{s,fl}}{s_{r,fl}}$
$\mu_{l,fr} = \mu_{r,fr}\frac{s_{l,fr}}{s_{r,fr}}$	$\mu_{s,rl} = \mu_{r,rl}\frac{s_{s,rl}}{s_{r,rl}}$
$\mu_{l,rl} = \mu_{r,rl}\frac{s_{l,rl}}{s_{r,rl}}$	$\mu_{s,rl} = \mu_{r,rl}\frac{s_{s,rl}}{s_{r,rl}}$
$\mu_{l,rr} = \mu_{r,rr}\frac{s_{l,rr}}{s_{r,rr}}$	$\mu_{s,rr} = \mu_{r,rr}\frac{s_{s,rr}}{s_{r,rr}}$

Table 2.5: Longitudinal and lateral friction for each wheel

2.5 Dynamic Model

This section covers the vehicles dynamic model. The equations of motion of the vehicle will describe the longitudinal and lateral motion, on the vehicle coordinate frame. These equations are derived thought the equilibrium forces in the x and y directions and a moment balance around the z axis.

2.5.1 Equation's of Motion

The resultant of external forces and moments that the vehicle receives from the ground and environment, constitutes the vehicle force system.

The longitudinal force, F_x is the resultant force acting along the x-axis. If $F_x > 0$ then the car is accelerating and if $F_x < 0$ then the vehicle is braking.
F_y represents the total lateral force. If the force is positive, it points to the left, from the drivers view point. The yaw moment M_z, is an upward moment about the z-axis. If the resultant moment is positive the car will turn left, if it's negative it will turn to the right [1, 5, p.258,p.340].

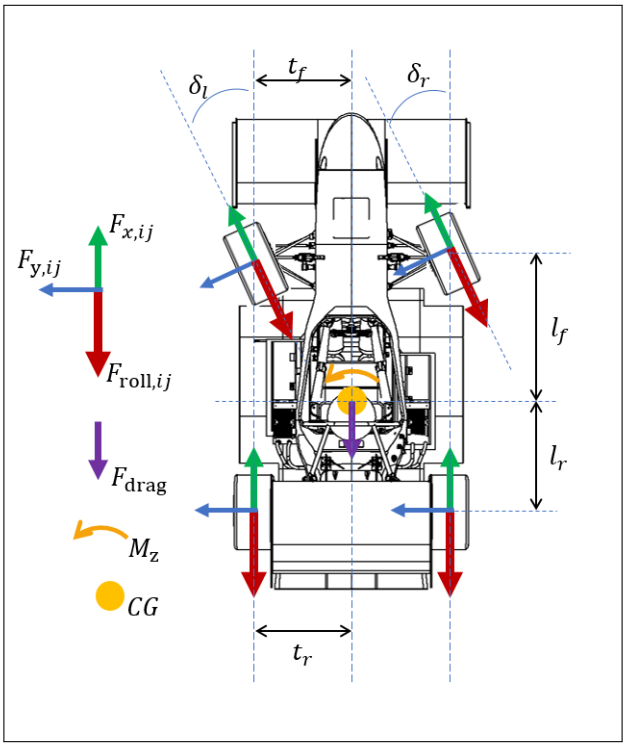


Figure 2.4: Vehicle's Free Body Diagram. $F_{x,ij}$ - Longitudinal force. $F_{y,ij}$ - Lateral force. $F_{roll,ij}$ - Rolling resistance. F_{drag} - Aerodynamic drag. M_z - Yaw moment. CG - Center of gravity

Term	Symbol	Value	Units
Inertia around z axis	I_{zz}	120	$[Kgm^2]$
Longitudinal friction force on the wheel	$F_{x,ij}$	-	$[N]$
Lateral friction force on the wheel	$F_{y,ij}$	-	$[N]$
Wheel's rolling resistance	$F_{roll,ij}$	-	$[N]$
Drag force	F_{drag}	1.29	$[N]$

Table 2.6: Table with terms and symbols for the free body diagram

$$\begin{aligned}
&\sum F_x = ma_x = \\
&= F_{x,fl} + F_{x,fr} + F_{x,rl} + F_{x,rr} - (F_{xroll,fl} + F_{xroll,fr} + F_{xroll,rl} + F_{xroll,rr}) - F_{drag} \\
&\sum F_y = ma_y = \\
&= (F_{y,fl} + F_{y,fr} + F_{y,rl} + F_{y,rr}) - (F_{yroll,fl} + F_{yroll,fr}) \\
&\sum M_z = I_{zz}\ddot{\psi} = \\
&= t_r(-F_{x,fl} + F_{x,fr} - F_{x,rl} + F_{x,rr}) + l_f(F_{y,fr} + F_{y,fl}) - l_r(F_{y,rr} + F_{y,rl})
\end{aligned} \tag{2.15}$$

2.5.2 Free Body Diagram of a Wheel

For the FST06e, which is a rear wheel driven vehicle, the wheel dynamics can be described by a simple torque balance at the wheel. Based on the applied torque (T_{rr}, T_{rl}) at each wheel, the angular acceleration $\dot{\omega}$ can be calculated. A damping factor b is added to equation, its value is obtained based on data collected from the car. [5, p.327].

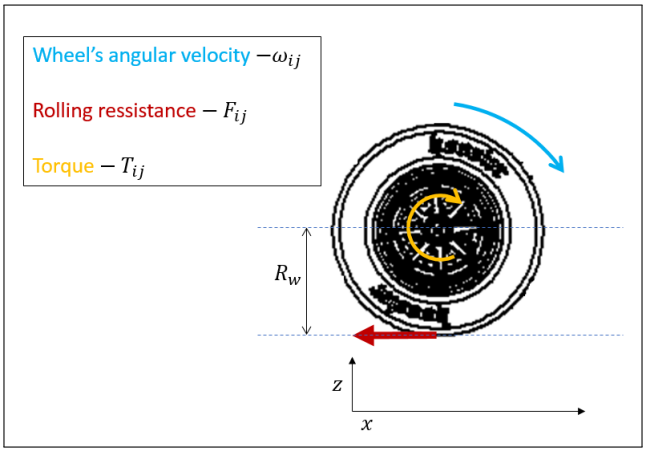


Figure 2.5: Free body diagram of a driven wheel

$$
\begin{aligned}
I_w\dot{\omega}_{fl} &= -R_w F^w_{fl} - b\omega_{fl} \\
I_w\dot{\omega}_{fr} &= -R_w F^w_{fr} - b\omega_{fr} \\
I_w\dot{\omega}_{rl} &= T_{rl} - R_w F^w_{rl} - b\omega_{rl} \\
I_w\dot{\omega}_{rr} &= T_{rr} - R_w F^w_{rr} - b\omega_{rr}
\end{aligned}
\tag{2.16}
$$

Term	Symbol	Value	Units
inertia of the wheel	I_w	2.5	$[Kgm^2]$
Radius of wheel	R_w	0.228	$[m]$
Wheel rolling resistance	F^w_{ij}	-	$[N]$
Torque	T_{ij}	-	$[Nm]$
Wheel's angular velocity	ω	-	$[rad]$
Wheel's angular velocity	$\dot{\omega}$	-	$[rads^{-1}]$
damping factor of the wheel	b	0.1	-

Table 2.7: Table with values and symbols for the free body diagram of the wheel

2.5.3 Rolling Resistance

At low speed the main force resisting the vehicle motion is the rolling resistance of the tires, which is caused by the deflection of the tire [6, p. 110]

$$
\begin{aligned}
F_{xroll,fl} &= c_{roll} F_{z,fl} sign(w_{fl}) \cos(\delta_l) \\
F_{xroll,fr} &= c_{roll} F_{z,fr} sign(w_{fr}) \cos(\delta_r) \\
F_{xroll,rl} &= c_{roll} F_{z,rl} sign(w_{rl})) \\
F_{xroll,rr} &= c_{roll} F_{z,rr} sign(w_{rr})
\end{aligned}
\tag{2.17}
$$

$$
\begin{aligned}
F_{yroll,fl} &= c_{roll} F_{z,fl} sign(w_{fl}) \sin(\delta_l) \\
F_{yroll,fr} &= c_{roll} F_{z,fr} sign(w_{fr}) \sin(\delta_r)
\end{aligned}
\tag{2.18}
$$

Note that because the car only has steering at the front wheels, there is no lateral force at the rear wheels.

Typical values for the rolling resistance can be found in [3, p. 121]

Road condition	c_{roll}
Concrete	0.008-0.1
Asphalt	0.01-0.0125

Table 2.8: Values for rolling resistance

2.5.4 Wind Resistance

When a vehicle is in motion it experiences a retarding force, which is called drag. Its action is always parallel to and opposite direction from the direction of motion. The aerodynamics values used are from tests made by the aerodynamics team. [6, p. 97].

Term	Symbol	Value	Units
Air density	ρ	1.223	$[Kgm^{-3}]$
Frontal Area	A	1.35	m
Aerodynamic drag coefficient	C_D	0.89	-

Table 2.9: Table with values and symbols of aerodynamic drag force

$$F_{Drag} = \frac{1}{2}\rho v^2 C_D A \tag{2.19}$$

2.5.5 Vertical Force and Weight Transfer

When the car accelerates, the chassis will accelerate in the opposite direction, as the weight at the front will shift to the back. Assuming that the weight transfer from the roll and pitch are independent, the vertical load F_z in each wheel can be determined separately based on the longitudinal and lateral accelerations [5, p.389].

Term	Symbol	Value	Units
Front wheelbase	l_f	2.5	$[m]$
Wheelbase	l	0.228	$[m]$
Height of center of Gravity(CG)	H_{CG}	0.813	$[m]$
Rear wheelbase	l_r	0.717	$[m]$
Front track	t_f	1.24	$[m]$
Rear track	t_r	1.24	$[m]$
Mass	m	356	$[Kg]$
Vehicle longitudinal acceleration	a_x	-	$[ms^{-2}]$
Vehicle lateral acceleration	a_y	-	$[ms^{-2}]$
Gravity	g	9.8	$[ms^{-2}]$

Table 2.10: Table with symbols and values for the mass transfer variables

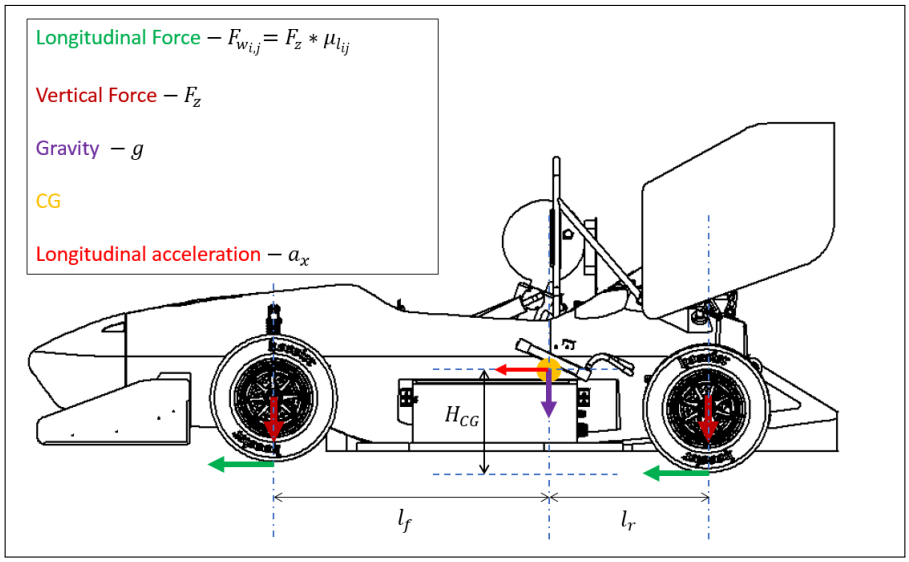


Figure 2.6: Vehicle's weight transfer with generation of longitudinal force

$$
\begin{aligned}
F_{z,fl} &= m\Big(\frac{l_f}{l}g - \frac{h_{CG}}{l}a_x\Big)\Big(\frac{1}{2} - \frac{h_{CG}}{t_f g}a_y\Big) \\
F_{z,fr} &= m\Big(\frac{l_f}{l}g - \frac{h_{CG}}{l}a_x\Big)\Big(\frac{1}{2} + \frac{h_{CG}}{t_f g}a_y\Big) \\
F_{z,rl} &= m\Big(\frac{l_r}{l}g + \frac{h_{CG}}{l}a_x\Big)\Big(\frac{1}{2} - \frac{h_{CG}}{t_r g}a_y\Big) \\
F_{z,rr} &= m\Big(\frac{l_r}{l}g + \frac{h_{CG}}{l}a_x\Big)\Big(\frac{1}{2} + \frac{h_{CG}}{t_r g}a_y\Big)
\end{aligned}
\tag{2.20}
$$

Once the vertical load is calculated, multiplying with the longitudinal and lateral friction coefficient from table 2.5, the longitudinal and lateral forces at each wheel can be calculated.

$$
\begin{aligned}
F_{wl,fl} &= \mu_{l,fl}F_{z,fl} \qquad & F_{ws,fl} &= \mu_{s,fl}F_{z,fl} \\
F_{wl,fr} &= \mu_{l,fr}F_{z,fr} \qquad & F_{ws,fr} &= \mu_{s,fr}F_{z,fr} \\
F_{wl,rl} &= \mu_{l,rl}F_{z,rl} \qquad & F_{ws,rl} &= \mu_{s,rl}F_{z,rl} \\
F_{wl,rr} &= \mu_{l,rr}F_{z,rr} \qquad & F_{ws,rr} &= \mu_{s,rr}F_{z,rr}
\end{aligned}
\tag{2.21}
$$

The previous equation (2.21) expressed in the wheel's coordinate frame, and it is required to be expressed in the fixed wheel's frame.

$$
\begin{aligned}
F^w_{x,fl} &= F_{wl,fl}\cos\alpha_{fl} + F_{ws,fl}\sin\alpha_{fl} \\
F^w_{x,fr} &= F_{wl,fr}\cos\alpha_{fr} + F_{ws,fr}\sin\alpha_{fr} \\
F^w_{x,rl} &= F_{wl,rl}\cos\alpha_{fl} + F_{ws,rl}\sin\alpha_{rl} \\
F^w_{x,rr} &= F_{wl,rr}\cos\alpha_{rr} + F_{ws,rr}\sin\alpha_{rr}
\end{aligned}
\tag{2.22}
$$

$$
\begin{aligned}
F^w_{y,fl} &= F_{ws,fl}\cos\alpha_{fl} - F_{wl,fl}\sin\alpha_{fl} \\
F^w_{y,fr} &= F_{ws,fr}\cos\alpha_{fr} - F_{wl,fr}\sin\alpha_{fr} \\
F^w_{y,rl} &= F_{ws,rl}\cos\alpha_{rl} - F_{wl,rl}\sin\alpha_{rl} \\
F^w_{y,rr} &= F_{ws,rr}\cos\alpha_{rr} - F_{wl,rr}\sin\alpha_{rr}
\end{aligned}
\tag{2.23}
$$

The computed longitudinal and lateral forces in the wheels frame, so they will be rotated to the bodys' frame to calculate the acceleration of the car.

$$
\begin{aligned}
F^B_{x,fl} &= F_{x,fl} = F^w_{x,fl}\cos\delta_l - F^w_{y,fl}\sin\delta_l \\
F^B_{x,fr} &= F_{x,fr} = F^w_{x,fr}\cos\delta_r - F^w_{y,fr}\sin\delta_r \\
F^B_{x,rl} &= F_{x,rl} = F^w_{x,rl} \\
F^B_{x,rr} &= F_{x,rr} = F^w_{x,rr}
\end{aligned}
\tag{2.24}
$$

$$
\begin{aligned}
F^B_{y,fl} &= F_{y,fl} = F^w_{y,fl}\cos\delta_l + F^w_{x,fl}\sin\delta_l \\
F^B_{y,fr} &= F_{y,fr} = F^w_{y,fr}\cos\delta_r + F^w_{x,fr}\sin\delta_r \\
F^B_{y,rl} &= F_{y,rl} = F^w_{y,rl} \\
F^B_{y,rr} &= F_{y,rr} = F^w_{y,rr}
\end{aligned}
\tag{2.25}
$$

2.5.6 Constraints

Until this point a model would already be possible however this model would be perfect, it would be able to perform every corner with maximum grip, which in reality is not true.
When cornering, due to weight transfer (covered in section 2.5.5), the outer wheels will be more loaded than the inner wheels. Thus more traction is available at the outer wheels, while the inner wheels will be less loaded thus having less traction available. To calculate the maximum lateral and longitudinal forces that can be transmitted to each wheel, the g-g diagram will be used. This diagram states the envelope of longitudinal and lateral forces that will be transmitted to the road [5, p. 322-324].

$$\begin{aligned}
\sqrt{F_{wl,fl}^2 + F_{ws,fl}^2} &\leq \mu_{r,fl} F_{z,fl} \\
\sqrt{F_{wl,fr}^2 + F_{ws,fr}^2} &\leq \mu_{r,fr} F_{z,fr} \\
\sqrt{F_{wl,rl}^2 + F_{ws,rl}^2} &\leq \mu_{r,rl} F_{z,rl} \\
\sqrt{F_{wl,rr}^2 + F_{ws,rr}^2} &\leq \mu_{r,rr} F_{z,rr}
\end{aligned} \tag{2.26}$$

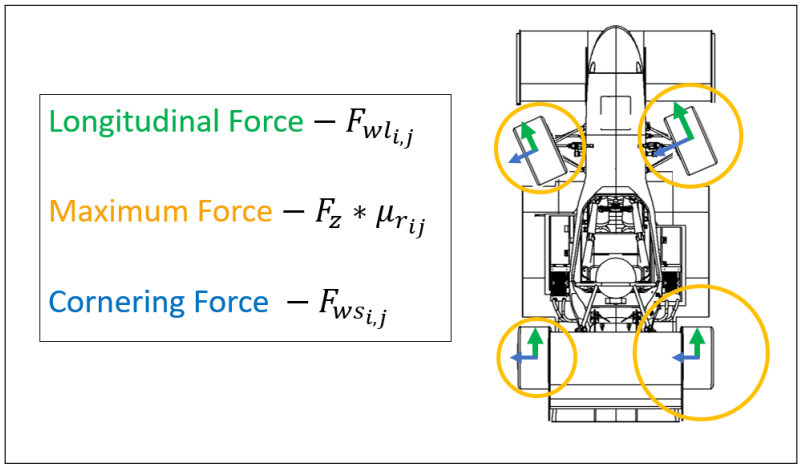


Figure 2.7: Maximum force circle in a case of a left corner. The rear right wheel has a bigger circle, which mean that more traction is available and thus more torque can be added

To calculate the maximum longitudinal grip in equation 2.26 the resultant friction coefficient (μ_r) will be change to the maximum possible friction coefficient (μ_{max}), calculate by the maximum value of equation 2.14.

$$\begin{aligned}
Fmax_{wl,fl} &= \sqrt{(\mu_{max,fl} F_{z,ij})^2 + F_{ws,fl}^2} \\
Fmax_{wl,fr} &= \sqrt{(\mu_{max,fr} F_{z,ij})^2 + F_{ws,fr}^2} \\
Fmax_{wl,rl} &= \sqrt{(\mu_{max,rl} F_{z,ij})^2 + F_{ws,rl}^2} \\
Fmax_{wl,rr} &= \sqrt{(\mu_{max,rr} F_{z,ij})^2 + F_{ws,rr}^2}
\end{aligned} \tag{2.27}$$

Then a rotation to the wheels' coordinate system is made.

$$\begin{aligned} F^w max_{x,fl} &= Fmax_{wl,fl} \cos\alpha_{fl} + F_{ws,fl} \sin\alpha_{fl} \\ F^w max_{x,fr} &= Fmax_{wl,fr} \cos\alpha_{fr} + F_{ws,fr} \sin\alpha_{fr} \\ F^w max_{x,rl} &= Fmax_{wl,rl} \cos\alpha_{rl} + F_{ws,rl} \sin\alpha_{rl} \\ F^w max_{x,rr} &= Fmax_{wl,rr} \cos\alpha_{rr} + F_{ws,rr} \sin\alpha_{rr} \end{aligned} \tag{2.28}$$

The same operation is done for the lateral force. Note also that the maximum lateral force depends on the left over of the longitudinal force.

$$\begin{aligned} F^w max_{y,fl} &= F_{ws,fl} \cos\alpha_{fl} - Fmax_{wl,fl} \sin\alpha_{fl} \\ F^w max_{y,fr} &= F_{ws,fr} \cos\alpha_{fr} - Fmax_{wl,fr} \sin\alpha_{fr} \\ F^w max_{y,rl} &= F_{ws,rl} \cos\alpha_{rl} - Fmax_{wl,rl} \sin\alpha_{rl} \\ F^w max_{y,rr} &= F_{ws,rr} \cos\alpha_{rr} - Fmax_{wl,rr} \sin\alpha_{rr} \end{aligned} \tag{2.29}$$

Equation 2.28 and 2.29 is in the wheels' coordinate system. To sum the forces it is more practical to be in the vehicle body frame.

$$\begin{aligned} F^B max_{x,fl} &= F^w max_{l,fl} \cos\delta_{fl} - F^w max_{y,fl} \sin\delta_{fl} \\ F^B max_{x,fr} &= F^w max_{l,fr} \cos\delta_{fr} - F^w max_{y,fr} \sin\delta_{fr} \\ F^B max_{x,rl} &= F^w max_{l,rl} - F^w max_{y,rl} \\ F^B max_{x,rr} &= F^w max_{l,rr} - F^w max_{y,rr} \end{aligned} \tag{2.30}$$

2.6 Simulink

The relations introduced in the previous sections of this chapter are implemented in *simulink*. The fixed step option is used and with a maximum step size of 0.001s. The model starts by receiving the steering value and the torque at each wheel. From the torque received it calculates the slip ratio (ratio between the velocity of the wheel and the velocity of the vehicle), which is then put in a tire model, from which a longitudinal and lateral friction is computed. Multiplying the vertical load by the friction coefficient gives us the forces at the wheels. It is then necessary to rotate to the vehicle body frame so that the sum of all forces can be used to calculate the accelerations and yaw moment. These values are then used to calculate the vertical load, drag coefficients and side slip.

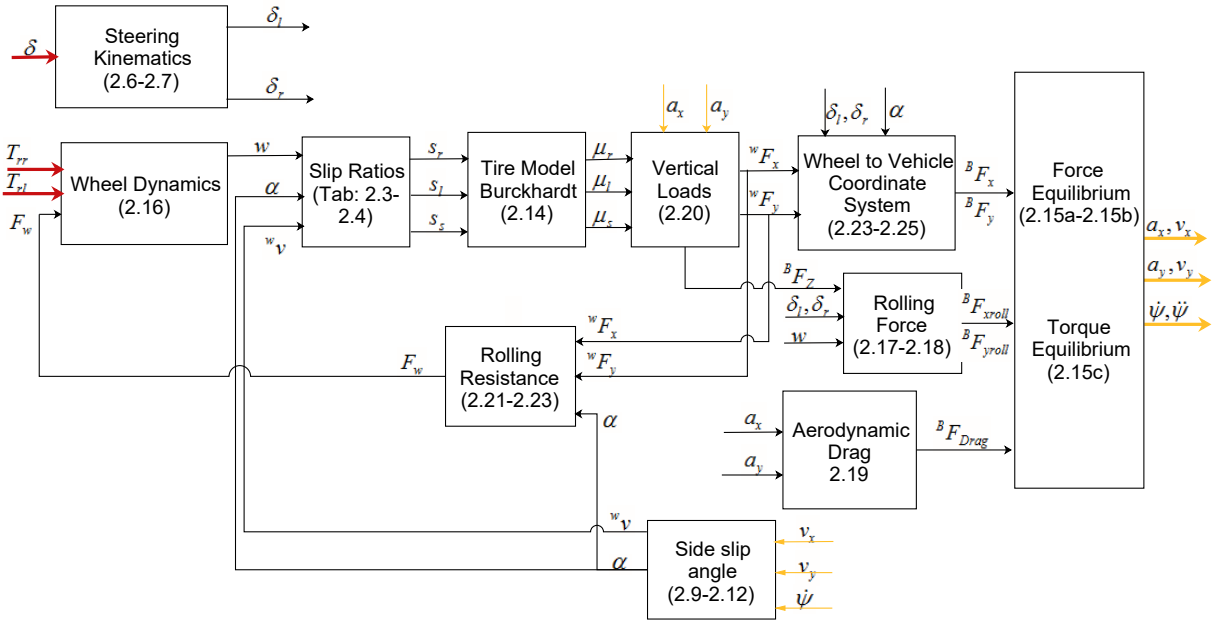


Figure 2.8: Simulink Schematic of Non Linear Model. Inputs: δ - Steering angle, T_{rr} - rear right torque, T_{rl} - rear left torque. Outputs: a_x - longitudinal acceleration,a_y - lateral acceleration, $\ddot{\psi}$ - yaw rate

2.7 Linear Model

The dynamic model described up until now is complex. When studying lateral dynamics, one of the most common models used is the linear bicycle model. This model merges both wheels of each axle into a single wheel. The model is linearized around a fixed velocity, and assumes that the left and right wheels generate the same lateral forces. In summary the following assumptions are used [1, 7, 8, 9, p.223, p.3, p.2]:

1. Velocity of the vehicle's center of gravity is considered constant along the longitude of it's trajectory.
2. All lifting, rolling and pitching motion will be neglected.
3. The mass of the vehicle is assumed to be at the center of gravity
4. Front and rear tires will be represented as one single tire, one each axle.
5. Aligning torque resulting from the side slip angle will be neglected.
6. The wheel-load distribution between front and rear axles is assumed to be constant.
7. The longitudinal forces on the tires, resulting from the assumption of a constant longitudinal velocity, will be neglected.

Before introducing the linearized equations the following terminology will be used in the linear model.

Term	Symbol	Value	Units
Yaw rate	$\dot{\psi}$	-	$[rads^{-1}]$
Longitudinal velocity	v_{x0}	[0,40]	$[ms^{-1}]$
Cornering stiffness at rear wheel	$C_{y,r}$	21429	$[Nrad^{-1}]$
Cornering stiffness at front wheel	$C_{y,f}$	15714	$[Nrad^{-1}]$
Inertia moment	I_{zz}	120	$[Kgm^2]$
Mass	m	356	$[Kg]$
Front wheelbase	l_f	0.873	$[m]$
Rear wheelbase	l_r	0.717	$[m]$
Steering angle	δ	[-3.3,3.3]	$[rad]$
Yaw moment	M_z	-	$[Nm]$
Lateral velocity	v_y	-	$[ms^{-1}]$

Table 2.11: Table with linear model values

2.7.1 Linear Model equations

To achieve the linear bicycle, the previous non linear model is linearised around a constant, longitudinal velocity v_{xo}. This simplification has the input of the steering angle, and will output the lateral force (v_y) and thus a yaw moment ($\dot{\psi}$) [3, p.636].

$$\dot{v_y} = -\frac{C_{y,f} + C_{y,r}}{m v_{x0}} v_y + \left(\frac{-l_f C_{y,f} + l_r C_{y,r}}{m v_{x0}} - v_{x0}\right)\dot{\psi} + \frac{C_{y,f}}{m v_{x0}}\delta \tag{2.31}$$

$$\ddot{\psi} = \left(\frac{-l_f C_{y,f} + l_r C_{y,r}}{I_{zz} v_{x0}}\right) v_y - \left(\frac{l_f^2 C_{y,f} + l_r^2 C_{y,r}}{I_{zz} v_{x0}}\right)\dot{\psi} + \frac{l_f C_{y,f}}{I_{zz}}\delta \tag{2.32}$$

The linear model written in state space will be:

$$\dot{x} = Ax + Bu_1$$

$$\dot{x} = \begin{bmatrix} \dot{v_y} \\ \ddot{\psi} \end{bmatrix}; \qquad x = \begin{bmatrix} v_y \\ \dot{\psi} \end{bmatrix}; \qquad u_1 = \delta$$

$$\begin{bmatrix} \dot{v_y} \\ \ddot{\psi} \end{bmatrix} = \begin{bmatrix} -\frac{C_{y,f} + C_{y,r}}{m v_{x0}} & \frac{-l_f C_{y,f} + l_r C_{y,r}}{m v_{x0}} - v_{x0} \\ \frac{-l_f C_{y,f} + l_r C_{y,r}}{I_{zz} v_{x0}} & -\frac{l_f^2 C_{y,f} + l_r^2 C_{y,r}}{I_{zz} v_{x0}} \end{bmatrix} \begin{bmatrix} v_y \\ \dot{\psi} \end{bmatrix} + \begin{bmatrix} \frac{C_{y,f}}{m v_{x0}} \\ \frac{l_f C_{y,f}}{I_{zz}} \end{bmatrix} \delta \tag{2.33}$$

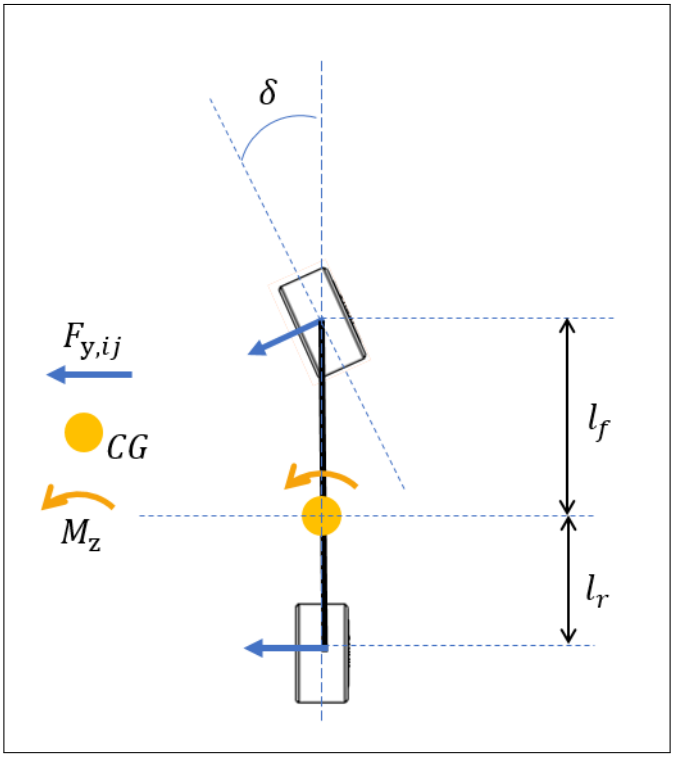


Figure 2.9: Vehicle's linear model free body diagram. δ - Steering angle, CG - center of gravity, M_z - Yaw moment, $F_{y,ij}$ - Lateral force

2.7.2 Linear Model equations with additional yaw moment

The way the model is presented, the equations only generate lateral force by the steering angle input. The goal with the torque vectoring is to generate yaw moment based on controlling the torque (longitudinal force) at the driven wheels. For this it will be necessary to introduce a new term M_z that will represent the additional yaw moment generated by the torque distribution.

$$\dot{x} = Ax + Bu_1 + Eu_2$$

$$\dot{x} = \begin{bmatrix} \dot{v_y} \\ \ddot{\psi} \end{bmatrix}; \qquad u_1 = M_z; \qquad u_2 = \delta$$

$$\begin{bmatrix} \dot{v_y} \\ \ddot{\psi} \end{bmatrix} = \begin{bmatrix} -\frac{C_{y,f}+C_{y,r}}{mv_{x0}} & \frac{-l_fC_{y,f}+l_rC_{y,r}}{mv_{x0}} - v_{x0} \\ \frac{-l_fC_{y,f}+l_rC_{y,r}}{I_{zz}v_{x0}} & -\frac{l_f^2C_{y,f}+l_r^2C_{y,r}}{I_{zz}v_{x0}} \end{bmatrix} \begin{bmatrix} v_y \\ \dot{\psi} \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{1}{I_{zz}} \end{bmatrix} M_z + \begin{bmatrix} \frac{C_{y,f}}{mv_{x0}} \\ \frac{l_fC_{y,f}}{I_{zz}} \end{bmatrix} \delta \tag{2.34}$$

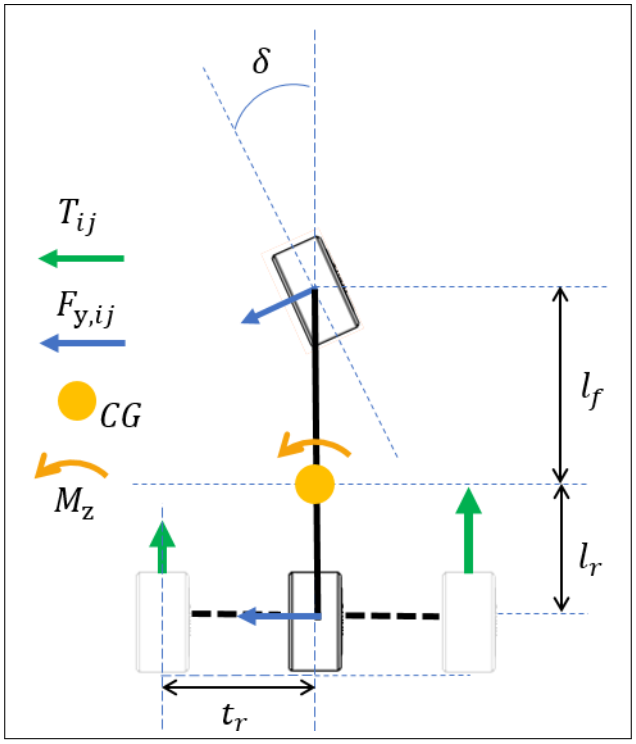


Figure 2.10: Vehicle's linear model with extension of the rear wheels in order to generate yaw moment based on the longitudinal force

The added momentum results from the difference between the left wheel torque T_{rl} and the right wheel torque T_{rr}. This difference multiplied by the half track of the car constitutes the additional yaw momentum in the model. If the right wheel as more torque than the left wheel, the car will have a positive yaw momentum, thus turning to the left. If the opposite happens the car while have a negative momentum and will turn right.

$$M_z = \Delta T * t_f = (T_{rr} - Trl)t_f \tag{2.35}$$

The torque at the wheel is not the same as the torque at the motor. Between the motor and the wheel there is a planetary gear set. This gear set amplifies the torque from the motor by a gear ratio, i.e $T_{wheel} = G_r * T_{motor}$. Then the torque at the wheel has to be divided by the wheel radius R_w to obtain the force at the ground.

Term	Symbol	Value	Units
Gear ratio	G_r	4.4	-
Half the track of the car	t_r	0.65	m
Radius of the wheel	R_w	0.265	m

Table 2.12: Table with values and terms for the torque conversion

Putting together all this information and with equation 2.38 the yaw moment from the difference in torque is given by:

$$\Delta T = \frac{R_w}{2t_r G_r} M_z = \frac{0.265}{2 * 0.65 * 4.4} M_z = 0.05 M_z \tag{2.36}$$

Thus rewriting the state space equation (2.34) expressed as function of the delta torque instead of the yaw moment.

$$\begin{bmatrix} \dot{v_y} \\ \ddot{\psi} \end{bmatrix} = \begin{bmatrix} -\frac{C_{y,f} + C_{y,r}}{m v_{x0}} & \frac{-l_f C_{y,f} + l_r C_{y,r}}{m v_{x0}} - v_{x0} \\ \frac{-l_f C_{y,f} + l_r C_{y,r}}{I_{zz} v_{x0}} & -\frac{l_f^2 C_{y,f} + l_r^2 C_{y,r}}{I_{zz} v_{x0}} \end{bmatrix} \begin{bmatrix} v_y \\ \dot{\psi} \end{bmatrix} + \begin{bmatrix} \frac{C_{y,f}}{m v_{x0}} \\ \frac{l_f C_{y,f}}{I_{zz}} \end{bmatrix} \delta + \begin{bmatrix} 0 \\ \frac{1}{0.05 * I_{zz}} \end{bmatrix} \Delta T \tag{2.37}$$

2.8 Model Calibration

Once the model is completed it is required to validate it with as much information, from the real system as possible. Some vehicle properties can be directly measured, such as mass, inertia and center of gravity. Quantities, like damping ratio and cornering stiffness cannot be measured directly. Thus these values have to be tuned based on measured data. The data used to validate both models is obtained from skidpads.

Skidpad is a circular track with a defined radius, in which the driver maintains a constant steering angle and velocity. The driver follows the radius circle while accelerating the vehicle moderately. After some time, the maximum velocity for the defined radius is reached, and it is not possible to follow the circle with a higher velocity.

The main advantage is that the radius of the circle is known, as well as the vehicle's velocity. This allows for the use of basic relations to verify if the data measured is correct. The recorded variables are compared with the corresponding outputs from the model in order to determine if it is sufficiently accurate.

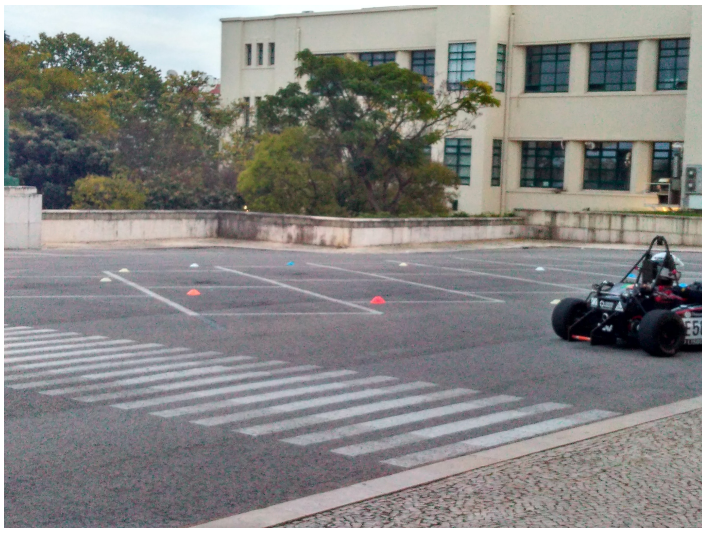


Figure 2.11: Test track marked by the cones to form a circular path

2.8.1 Test Track

The constant radius turn (skidpad) is defined by ISO 4138 [10]. According to the norm, the test should be performed with a minimum circle radius of 30 m. For the FST06e, the available test track is at the IST campus in front of the main build in the parking lot (see figure 2.11), has the maximum circle radius of 7.5m. Another test track was at the International Kartodromo in Palmela, where the maximum radius is 15m. Therefore, the constant radius test had to be modified.

In the Formula Student competition, one of the disciplines is to perform a constant radius turn in a track with a radius of 8.75 m. To be closer to the competition conditions, the tests are modified and performed with a range of 5-9m radius.

In total, three data sets are available. These tests were done in different days and in two different places. ”Test 1” and ”Test 2” were both done in front of the central building at IST. The radius of the circle was the same, but the driver varied the velocity. ”Test 3” is a test with a bigger radius done in Palmela. Table 2.13 summarizes the tests.

	Test 1	Test 2	Test 3	Unit
Radius of circle	5.62	5.62	9.02	$[m]$
Global velocity	7	8.5	9.3	$[ms^{-1}]$
Steering angle	110	100	71	$[deg]$

Table 2.13: Table of the three tests with the recorded values: Steering angle, global velocity and radius of circle

During the tests some data points are being monitored. The inputs are the global velocity and the steering angle, the output values are longitudinal, lateral acceleration and yaw rate. Although more data from the car can be logged, these are sufficient to conclude if the models are accurate enough [5, p. 345].

Model Variables		
Input	Steering angle	δ
	Torque/Velocity	T, v_{CG}
Output	Longitudinal acceleration	a_x
	Lateral acceleration	a_y
	Yaw rate	$\dot{\psi}$

Table 2.14: Measured variables

In the following sections 2.8.2, 2.8.3 the data collected from ”Test 1” is used. The same procedure is made for all 3 tests but for illustration purposes only a graphical comparison between the data and model validation from ”Test 1” is shown. The remaining tests are summarized in tables 2.16, 2.17.

2.8.2 Inputs

The data presented illustrates that the vehicle made 4 runs. Two to the left and two to the right. Negative values of steering angle and yaw rate represent the car cornering to the right (in a clockwise way) while positive values represents the car cornering left (counter clockwise).

Steering Angle

Figure 2.12 shows the angle of the steering wheel. At 96s, the driver goes in a counter clockwise skidpad during 30s, turning the steering wheel 125deg. At 125s he exits the skidpad and does the same maneuver but this time to the right(140-180s), turning 145 deg. The vehicle then repeats the test. Notice that the car is not symmetrical in terms of turning right and left, due to the current suspension setup.

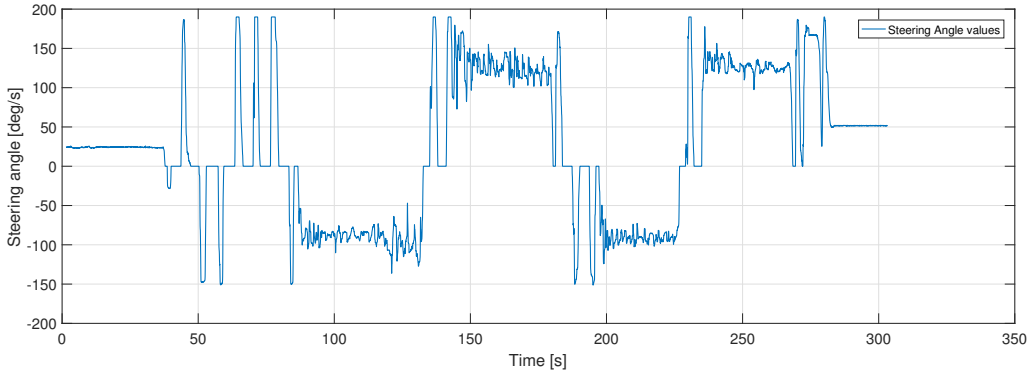


Figure 2.12: Steering angle data of car from Test1

Velocity

Figure 2.13 shows the velocity of the vehicle. It can be seen that during the skidpad test the driver maintains a constant velocity of 7 m/s. The variations of velocity (110-120s), occur when the car exits the circle and goes around the track to prepare for the next lap.

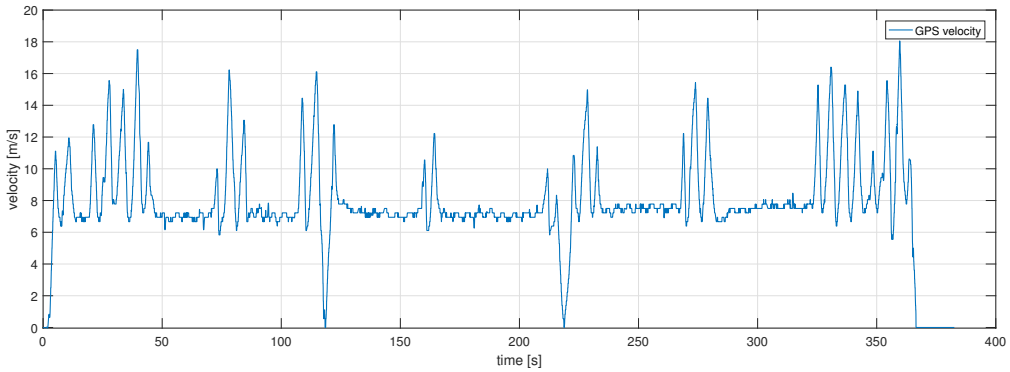


Figure 2.13: Longitudinal velocity data from Test1

Yaw Rate

The yaw rate of the car is measured through the rate gyro. Once again, for negative yaw rate values the car is cornering to the right and for positive value it is cornering to the left. The yaw rate for this particular track is 72 deg/s. Note that the yaw rate is symmetrical for both right and left cornering.

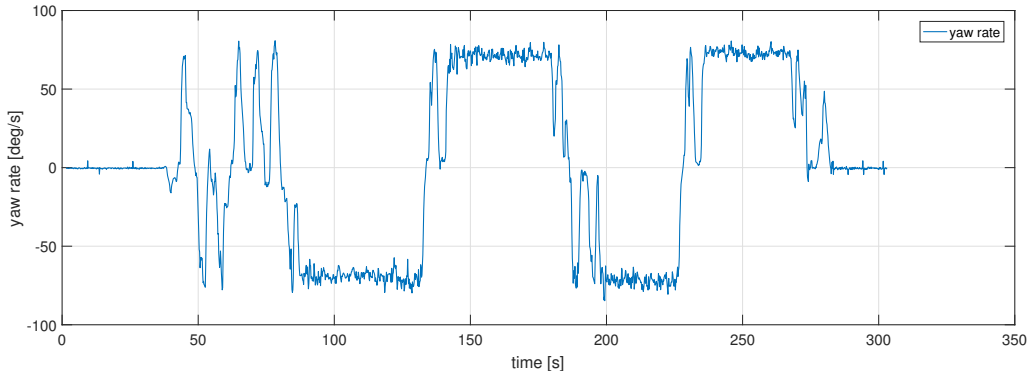


Figure 2.14: Yaw rate data from Test1

2.8.3 Outputs

The next step is to give the same inputs to the model and compare the simulation results with the real data. The model has the velocity and steering as inputs, and it will be compared against the accelerations and yaw rate.

Yaw Rate

The first step was to verify if the data provided from the sensors are accurate. Based on the gathered data, presented in table 2.13, knowing the radius of the track as well as the vehicles' speed, the yaw rate is calculated. Which is then compared to the measured yaw rate. In a steady state cornering the yaw rate can be computed by the following expression:

$$\dot{\psi} = \frac{v_{CG}}{Radius} \tag{2.38}$$

$$\dot{\psi} = \frac{9.16}{8.3875} \approx 1.09 rad/s \approx 62 deg/s$$

	Yaw Rate (Real Data) [deg/s]	Theoretical Yaw Rate [deg/s]	Difference [%]
Test 1	58.6	63	6.98
Test 2	72	86.65	20
Test 3	72.3	71.37	1.29

Table 2.15: Yaw Rate comparison between theoretical and read data

Based on the values presented in table 2.15, it can be inferred that the sensors used in the car are working and giving correct values. ”Test 2”, the big difference presented is that for the car to be able to maintain that velocity and steering angle, the trajectory of the vehicle (around the track) had to be bigger.

Figure 2.15 shows the comparison between the yaw rate from the vehicle and the yaw rate from both linear and non linear simulations. It can be seen that the simulated values are very close to the real data, except at (156-157s and 171-173s), where the suffered from understeer.

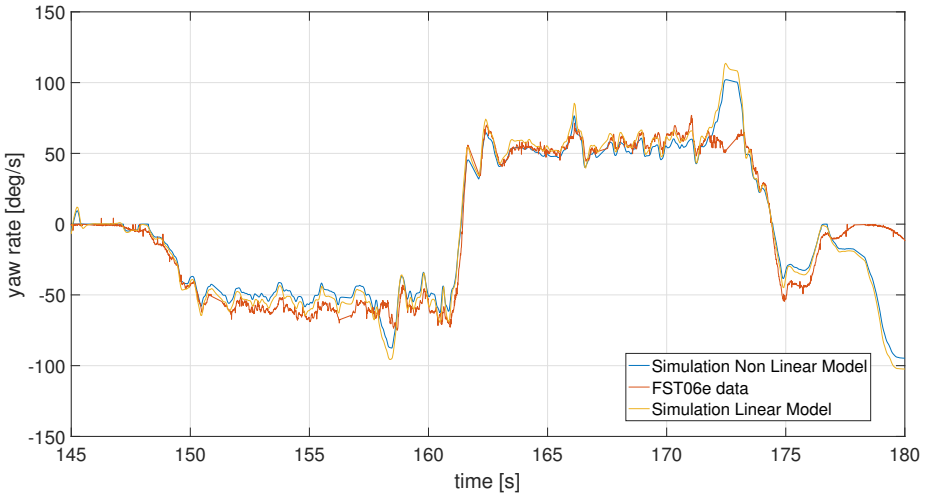


Figure 2.15: Comparison between real data from the fst06e and both linear and non linear simulation during a skidpad to the left and right from Test1

Longitudinal Acceleration

The spikes presented in the images are from the braking and acceleration from the driver.

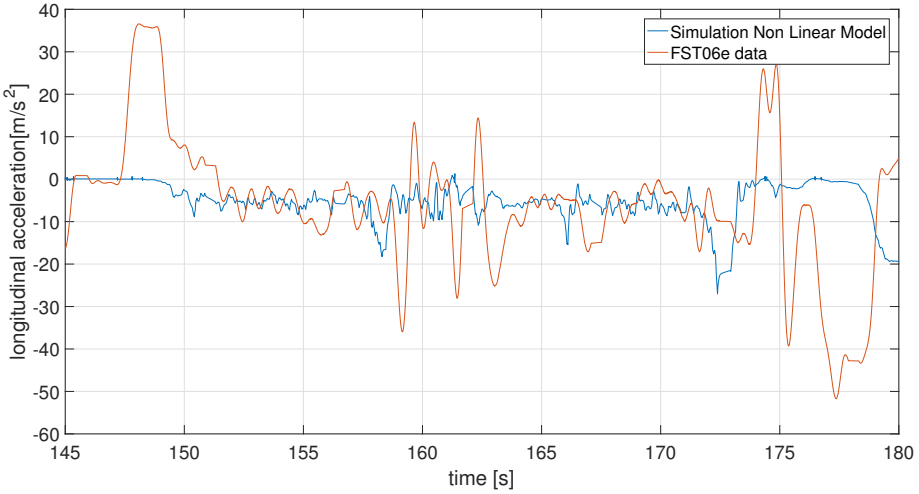


Figure 2.16: Correlation between longitudinal acceleration data from the real vehicle and non linear model during a skidpad to the left and right from Test1

Lateral Acceleration

From (150-160s), when the car is cornering to the right, the model is not adjusted while when the car is cornering to the left from (160-175s) the model is adjusted. This is because of the steering asymmetry of the car, resultant of the suspension setup.

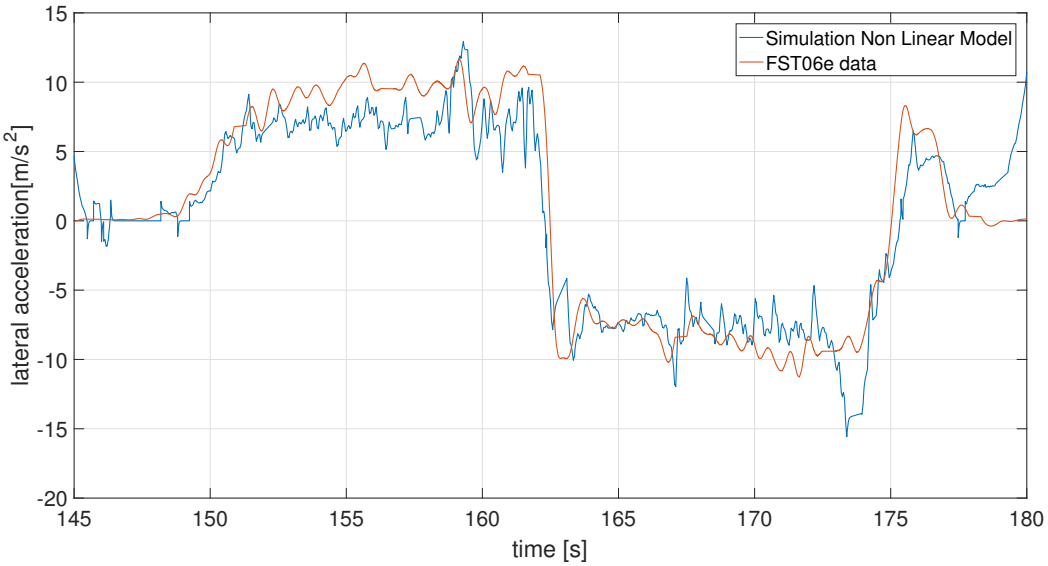


Figure 2.17: Correlation between lateral acceleration data from the real vehicle and non linear model during a skidpad to the left and right from Test1

The same analysis is done to ”Test 2” and ”Test 3”. The results are summarized in table 2.16 and 2.17.

	Real Data			Non Linear Simulation		
	a_x [m/sˆ2]	a_y [m/sˆ2]	$\dot{\psi}$ [deg/s]	a_x [m/sˆ2]	a_y [m/sˆ2]	$\dot{\psi}$ [deg/s]
Test 1	1.14	8.67	72.3	1.035	8.2	70.8
Test 2	0.9293	9.694	72	0.839	8.8	69
Test 3	0.67	8.9	59.63	0.6	8	60.2

Table 2.16: Comparison of the longitudinal, lateral acceleration and yaw rate between the real vehicle and the model

	Difference		
	a_x [%]	a_y [%]	$\dot{\psi}$ [%]
Test 1	8.6	4.6	2
Test 2	9.71	9.2	4.17
Test 3	9.51	10	2.2

Table 2.17: Difference between the vehicle and simulation for the longitudinal, lateral acceleration and yaw rate expressed as percentages

Chapter 3

Proposed Controller

In this chapter the proposed torque vectoring control system design will be discussed. It starts by an introduction to the available sensors onboard the FST06e. Based on the availability, sampling time and quality of the sensors, the strategy for the calculation of the reference value is presented, followed by the choice and tunning of the controllers. The chapter concludes with a comparison between the PI and LQR controllers in both the linear and non linear models.

A variety of control laws can be used for torque vectoring. The most basic method is to distribute the left and right torque, proportional to the amount of steering input $\Delta T = f(\delta)$.
The proportional integral derivative (PID) controller is the classic control structure and the most commonly used in practical applications. It is a straightforward method to implement and tune [11, 12].
Sliding mode control is a non linear control design methodology used by several researchers to achieve the objectives of tracking the yaw rate and slip angle [11, 13, 14].
Predictive control estimates the future states of the vehicle in order to find the best control input [15, 16].
Some authors also implement Fuzzy control to create a set of rules for the allocation of the torque [9].

3.1 Available sensors

Developing a torque vectoring system can be tackled resorting to several different approaches. The majority [7, 8, 9, 17] use the yaw rate of the vehicle as the reference for the controller. More advanced solutions [11, 13, 18] use the sideslip angle and/or a combination of yaw rate and sideslip angle. The choice on the strategy will depend on the available sensors. The FST06e is equipped with the following sensors:

- Brake and accelerator pedal positions, based on data from 2 linear potentiometers.
- Front left and right wheel speeds, from 2 wheel encoders
- Steering angle measured by a rotational Potentiometer
- Vehicle acceleration and rotation, based on data from 1 inertial measuring unit (IMU)
- Vehicle position and velocity, thought the use of a GPS
- Motor torque and RPM, provided by the 2 Siemens Inverter

3.1.1 GPS

The FST06e is equipped with one GPS receiver, located in front of the driver providing, accurate measurements of the vehicle velocity. The start up time, it can take up to 30sec for the GPS to acquire signal, and if there are not enough satellites in view the signal can be lost. Signal dropouts are a relevant limitation as our model is dependent on the velocity.

GPS - Skytraq S1216f8	
Update rate	50 Hz
Velocity accuracy	0.1 m/s
Timing accuracy	21 ns
Start up Time	1/28/29 sec (hot/warm/cold start)

Table 3.1: GPS datasheet values

3.1.2 Wheel speed encoder

The wheel speed encoder is installed in a non driven wheel. Thus the true speed of the car at the wheel can be obtained assuming no slippage, then by a matrix rotation the vehicle speed can be computed. The wheel speed encoder is an alternative to the GPS, the only limitation being that for low speeds the wheel encoder has low resolution.

Wheel encoder - leonard and bauer gel 2443	
Speed range	700 min^{-1}

Table 3.2: Wheel encoder datasheet values

3.1.3 IMU

The FST06e is equipped with an Inertial Measurement Unit (IMU). The IMU will be used to measure the yaw rate and the accelerations, to measure the increase or decrease in yaw rate for the different controllers.

IMU - GY-80	
Accelerometer	
Range (dynamic)	2/4/16 g
Noise density	150 μg/$\sqrt{(Hz)}$
Non linearity	$<0.05\%$
Bias Stability	20 μg
Bandwidth	200 Hz
Gyroscope	
Range (dynamic)	250/500/200 deg/s
Noise density	0.03 deg/s/$\sqrt{(Hz)}$
Non linearity	$<0.2\%$
Bandwidth	100/200/400 Hz

Table 3.3: Accelerometer datasheet values

3.1.4 Steering encoder

With a rotary potentiometer it is possible to know how much the steering has turned to the left or to the right.

Steering Encoder	
Range	4096 deg
Resolution	0.1 deg
Accuracy	2 deg

Table 3.4: Steering Encoder datasheet values

3.1.5 Acceleration and brake pedal

Resorting to a pair of linear potentiometers it is possible to know the percentage of throttle and braking requested by the driver in real time. WIth this information, it can be computed the amount of torque provided by the motors.

3.1.6 Motor Controllers

The FST06e has 2 inverters from Siemens, which are responsible for controlling the motors. They take the DC current from the battery pack as an input, and output a three phase, AC current to power each motor. These inverters sends information, like the motor speed, torque, current and power.

From the inverters it is possible to know the value of the torque at each wheel, which can be used to know if the controller is sending the correct torque value to the inverters. Also, it is possible to know the wheel speed by assuming that the wheel does not have slippage occurring.

3.1.7 Sample Times

All sensors have different sampling period. Table 3.5 shows the sample time from each one. Based on the maximum sampling time that each sensor was able to provide. It is advantageous to have more data points available and discard/average them, than having less points.

Sensor	Sampling Time (ms)
IMU	8
Encoders Rear	50
Encoders Front	100
GPS	40
Steering Angle Encoder	40
Pedal	10

Table 3.5: Acquisition time of the sensors form the FST06e

To read all the data a Matlab script was developed. This program receives, converts and resamples all the data so that the user can correlate it, being possible to infer the performance of the car.

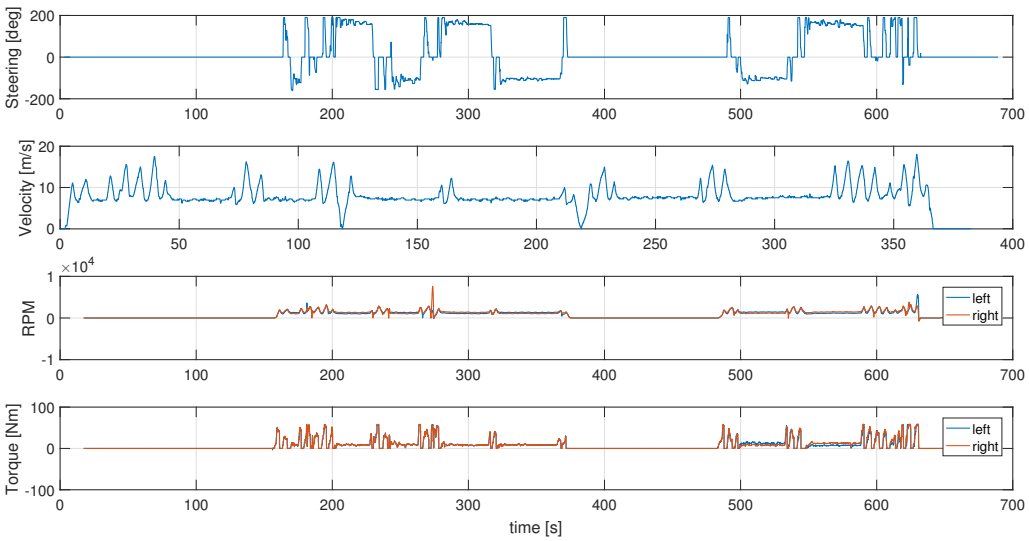


Figure 3.1: Data processed from the matlab script for a skidpad run. The data shown is: steering angle, GPS velocity, motor RPM, motor torque

3.2 Reference Value

Before the introduction of the control algorithm, it is necessary to define a reference signal. Various authors propose the calculation of the reference based on the velocity and steering angle of the car. It is assumed that the vehicle is in a steady state condition [2].

3.2.1 Desired Yaw Rate

This signal is adapted to the characteristics of the car's behavior. It can be defined by the ratio between front and rear masses and between the front and rear tire cornering stiffness [6, p.202-204].

$$K_u = \frac{l_r m}{C_{y,f}(l_f + l_r)} - \frac{l_f m}{C_{y,r}(l_f + l_r)} \tag{3.1}$$

If the under-steer gradient, K_u is positive ($K_u > 0$): The car is said to have an under-steer behavior (under yaw rate).

$$\frac{l_r m}{C_{y,f}(l_f + l_r)} > \frac{l_f m}{C_{y,r}(l_f + l_r)} \rightarrow K_u > 0 \tag{3.2}$$

If K_u is negative ($K_u < 0$): The car is said to have an over-steer behavior (over yaw rate).

$$\frac{l_r m}{C_{y,f}(l_f + l_r)} < \frac{l_f m}{C_{y,r}(l_f + l_r)} \rightarrow K_u < 0 \tag{3.3}$$

And if $K_u = 0$, it means that the car is neutral - steer (ideal yaw rate):

$$\frac{l_r m}{C_{y,f}(l_f + l_r)} = \frac{l_f m}{C_{y,r}(l_f + l_r)} \rightarrow K_u = 0 \tag{3.4}$$

A neutral-steer vehicle has the smallest possible turning radius for a given velocity, which corresponds to optimal performance. Therefore, it would be assumed that a neutral-steer vehicle should be chosen as reference. However, this approach would put the car on the verge of over-steer instability, so a slightly under-steered vehicle is taken as reference. This reference is much closer to neutral-steered than the actual car, so the controller can improve the yaw rate. The ideal or desired yaw rate can be defined by the velocity and the radius of the curve:

$$\dot{\psi}_{desired} = \frac{v_{CG}}{R} \tag{3.5}$$

Given the velocity and steering angle of the car and with the known steer gradient and wheelbase it is possible to know the turning radius.

$$\frac{1}{R} = \frac{\delta}{(l_r + l_f) + K_u v_{CG}^2} \tag{3.6}$$

Combining both expressions 3.5 and 3.6, the under-steer gradient and the circular road radius gives the desired yaw reference.

$$\dot{\psi}_{desired} = \frac{v_{CG}}{(l_r + l_f) + K_u v_{GC}^2}\delta \tag{3.7}$$

The desired yaw rate is a function of the velocity, steering and characteristic of the car. The under-steer gradient K_u can be tuned in for each driver preference. The smaller the under-steer gradient the bigger the difference between the desired and actual yaw rate, more will the car have neutral steer characteristics, and more difficult to drive.

To determine the values of the under-steer gradient it is necessary to know the cornering stiffness of both the front and rear tires. The best approach is to test the tires in a tire testing facility, derive a tire model and from it retrieve the cornering stiffness.

For Formula Student teams, data for the tires is available from the TTC (Tire Test Consortium). The data is only available as raw data and therefore a lot of pre-processing and model fitting is required. It is possible to get a simple graph that correlates the vertical load with the cornering stiffness directly from the raw data.

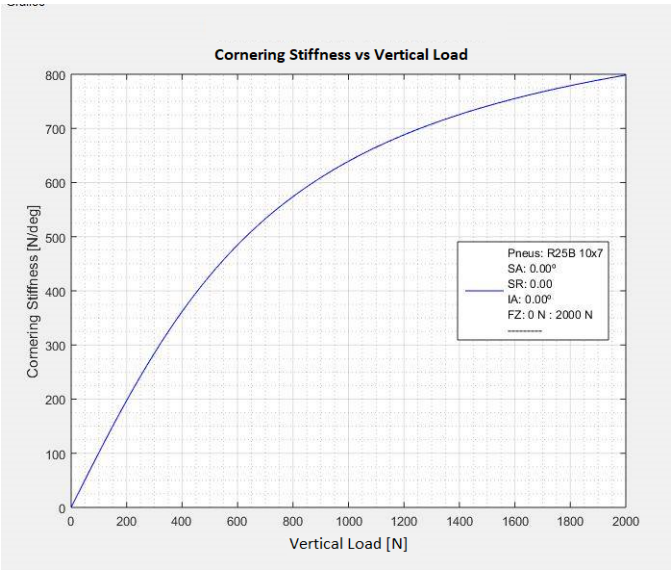


Figure 3.2: Correlation between vertical load with the cornering stiffness

As a first approximation, the vertical load is simply the weight of the car distributed on each wheel. Because the car has a weight distribution of 40% front and 60% rear the vertical load will be respectively 697N in and 1046N. From (figure 3.2), the cornering stiffness will be:

$$C_{y,f} = 525N/deg \qquad C_{y,r} = 633N/deg$$

Because the cornering stiffness is dependent of the vertical load and on the acceleration, a series of tests was done with different velocities in order to have an idea if the theoretical value of the cornering stiffness is close to the real value.

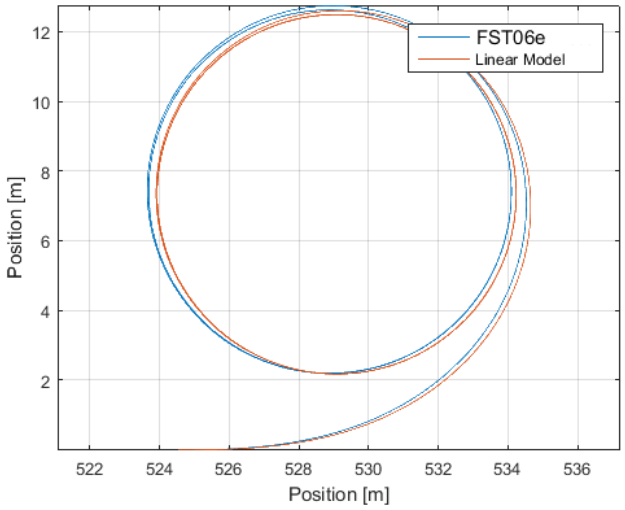


Figure 3.3: Comparison between real vehicle data and linear model for a steady state cornering

$$C_{y,f} = 546N/deg \qquad C_{y,r} = 745N/deg$$

3.2.2 Maximum Yaw Value

With all the various possible implementations and control strategies, some limitations are valid for all torque vectoring controllers. These limitations are related to the physical properties of the vehicle like the maximum yaw moment, the maximum tire adhesion, microprocessor computing time, etc. Depending on the entry speed of the car, it will be able or not to achieve the desired yaw. If entering in a corner too fast the road may be unable to provide the necessary tire forces, and the car just goes forward, thus under-steering. The solution is to bound by its limiting factor the tire-road coefficient [2, p. 233].

$$v_{CG}\dot{\psi} + a_x\beta + \frac{v_{CG}\dot{\beta}}{\sqrt{1+\tan\beta^2}} \leq \mu g \tag{3.8}$$

In equation (3.8), if considering that the car has small heading angle, the equation can be further simplified and reduced to:

$$\dot{\psi}_{max} = \sigma\frac{\mu g}{v_{CG}} \tag{3.9}$$

where σ represents a tunability factor to take into account changes in the friction coefficient from different types of pavement. The yaw rate reference will be used as long as it does not pass the maximum possible yaw rate.

$$X(m,n) = \begin{cases} \dot{\psi}_{des}, & |\dot{\psi}_{des}| \leq \dot{\psi}_{max}| \\ \pm\dot{\psi}_{max}, & \text{otherwise} \end{cases} \tag{3.10}$$

3.3 PI Controller

3.3.1 Requirements/Limitations

If the controller is designed to be very fast, it could produce a yaw moment so big that the car could not be able to achieve by lack of tire adhesion and end up oversteering. If the controller is designed to be much slower, the driver won't feel any change in the car. Also, if the frequency of calculation is too high, the microprocessor may not be able to process everything in real time.

The goal of the controller is to be somewhat between the two states described, fast enough to ensure good tracking of the yaw rate, while not exceeding the physical limitations of the vehicle and driver. Based on the available data, and with the goal to reduce the under-steering of the vehicle, the controller should have an overshoot less than 10%, and settling time of 0.2s.

Also, the maximum torque that the controller can request has to be known in order to verify if the control effort simulated is feasible. Table 3.6 summarizes the maximum torque the motor can provide, and the maximum torque that the wheel will have.

Term	Symbol	Units
Max motor speed	8000	RPM
Max motor torque	107	Nm
Max vehicle torque	438.7	Nm

Table 3.6: Maximum RPM and Torque from the motors

3.3.2 Step response

Retrieving the linear model from section 2.7.2 an analysis of the vehicle behaviour will be performed. Figure 3.4 shows the response from the yaw moment to the yaw rate. For lower velocities, the magnitude of the response is lower but with a shorter settling time.

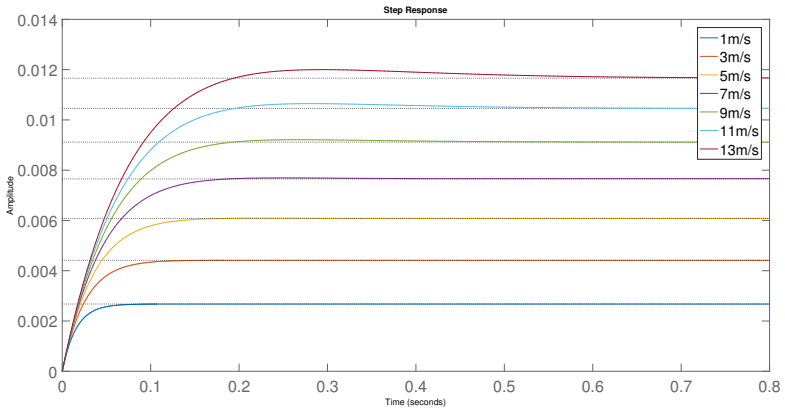


Figure 3.4: Step Response M_z to $\dot{\psi}$ for different velocities

3.3.3 Root Loci

The location of poles and zeros characterize the behavior of the system. Poles close to the left imaginary axis represent a slower system while poles far from the plane define a rapid system. Figure 3.5 shows the poles of the linearized single track model around various longitudinal velocities. These points vary from 1 m/s to 13 m/s. For low velocities, two poles are located far in the left semi plane. At higher velocities the poles move closer to the imaginary axis and end up as complex conjugate poles.

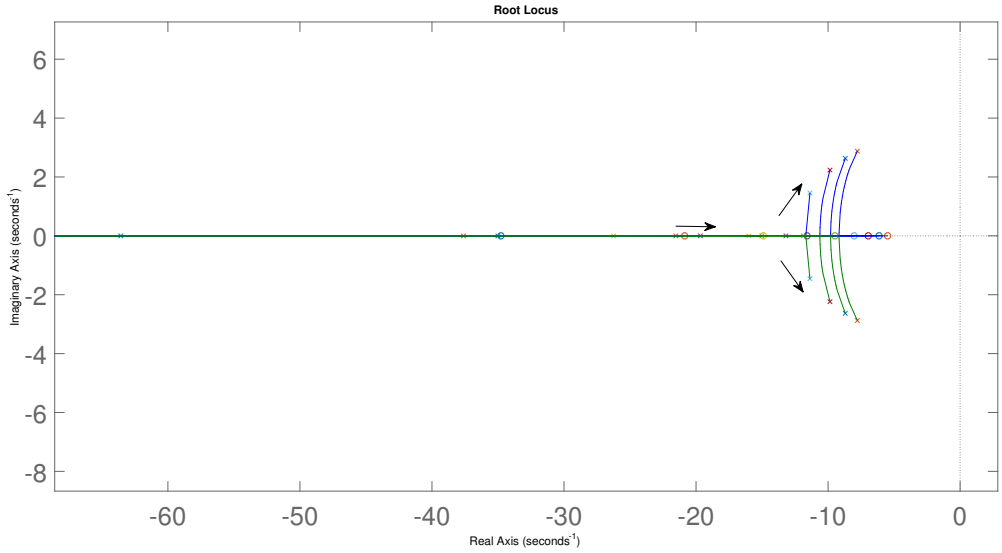


Figure 3.5: Root loci for various operating velocities [1-13m/s]

3.3.4 Frequency Domain

Figure 3.6 shows the frequency behaviour of the system for varying longitudinal velocities. The absolute magnitudes of the signals are not significant because the system is not normalized. For example, a yaw moment change of 1 Nm has nearly no effect. The main factor is the impact of the longitudinal velocity on the values. With increasing velocity the value of the magnitude also rises. This implies that depending on the velocity the car will react differently to the same inputs. The frequency responses for a constant longitudinal velocity and varying steering angles are not shown here because these differences are relatively low, compared to the longitudinal velocity variation.

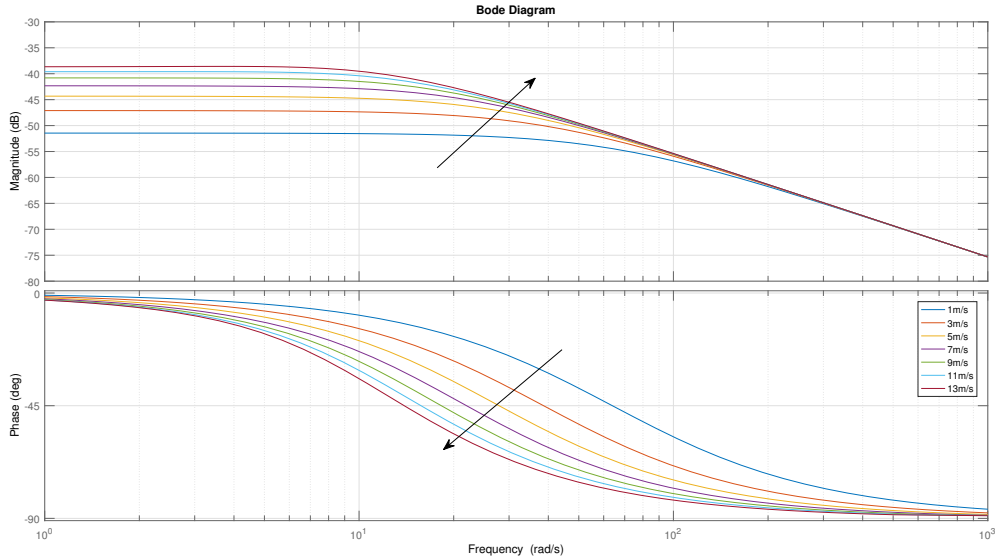
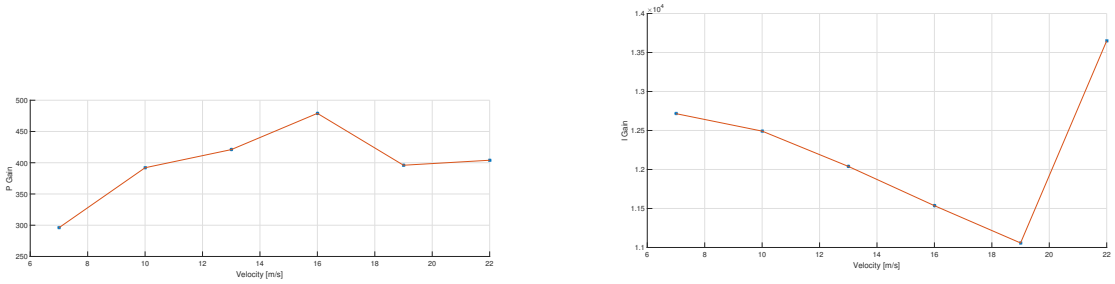


Figure 3.6: Bode plot for various operating velocities [1-13m/s]

Taking into consideration that the linear model is dependent on the velocity, the design of one controller will not be sufficient to ensure the control for every point. So a gain scheduled controller is implemented. Six different setting points are chosen. The main disadvantage of having few points is that when the gains change the driver could sense an unexpected yaw rate change. If this occurs more setting points are necessary and thus the same procedure is repeated.

Vx (m/s)	P	I
7	296.29	12716.7
10	392.23	12492.5
13	421.72	12040
16	479.87	11536.07
19	396.22	11058.5
22	404.79	13650

Table 3.7: PI values for different velocities values



(a) P gains (b) I gains

Figure 3.7: P and I gains of the controller

As the velocity increases higher proportional gains (P) are obtained, this makes sense with the dynamic of the car. The faster the car goes the more difficult it will be for the car to change it's direction.

3.3.5 Anti-Windup

The anti-windup of the integrator plays an important role in the control system. Many situations, like wheel spinning or drifting, may arise where the controller output will not be able to reach the desired yaw moment. It is necessary to bind the integral term so that it will not explode to infinity. This saturation will occur from two places. The first was already addressed in section (3.2.2), where we concluded that depending on the road and weather conditions the car could or could not achieve the desired yaw reference. That problem was already solved in section (3.2.2) where we bounded the yaw value. Another problem arises when the algorithm computes a value that it is possible to achieve the desired yaw, but the motors can't provide the requested torque at the wheel, for instance when the wheel is spinning. The integral part would then increase to infinity. To prevent this situation a condition integration was added.

$$\begin{aligned} &if DesiredTorque > MaxTorque \\ &DesiredTorque = MaxTorque \\ &e(t) = e(t) \\ &end \end{aligned} \tag{3.11}$$

When the motor is giving the maximum possible torque but still the car is no able to get to the desired yaw the integral term will not continue to accumulate error.

3.3.6 Simulation

Using *Simulink* a simulation environment is created to test the designed PI look-up table in the linear model. Based on the current velocity of the vehicle a set of P and I gains are chosen from the table. These gains are feed to the PI controller block, which then sends the amount of torque to the model. The linear model is from eq: 2.34, because the model is velocity dependent, it is constantly updating based on the velocity, also the model has the steering angle as an input. This allows to test the controller to the extra yaw moment provided by the steering angle.

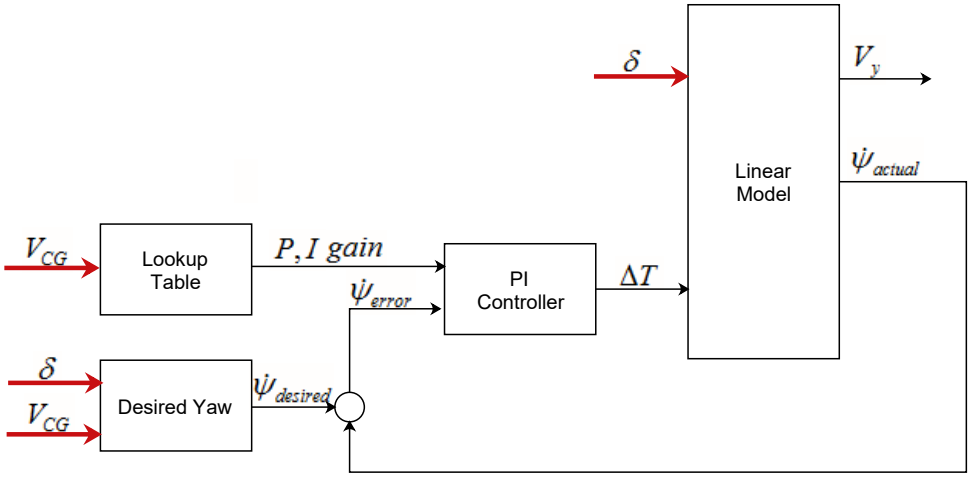


Figure 3.8: Schematic of simulink linear model

Figure 3.9 shows a comparison between the response of the vehicle with and without the TV controller for a constant velocity of 9m/s and a steering angle of 70 deg.

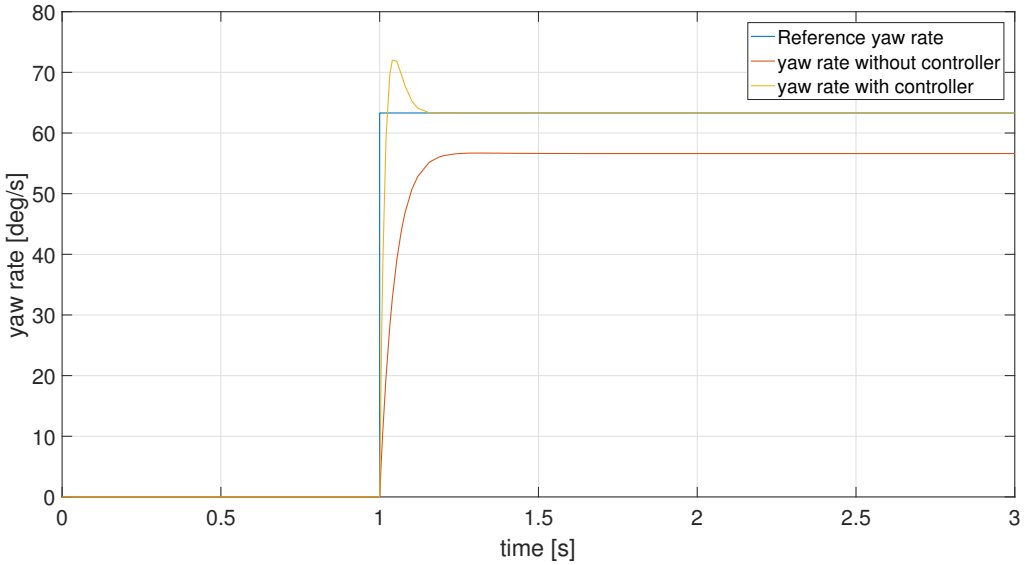


Figure 3.9: Step response of yaw rate $\dot{\psi}$ with and without controller

Figure 3.10 compares the response of the controller on the linear and non linear model for the same velocity and steering angle as in figure 3.9.

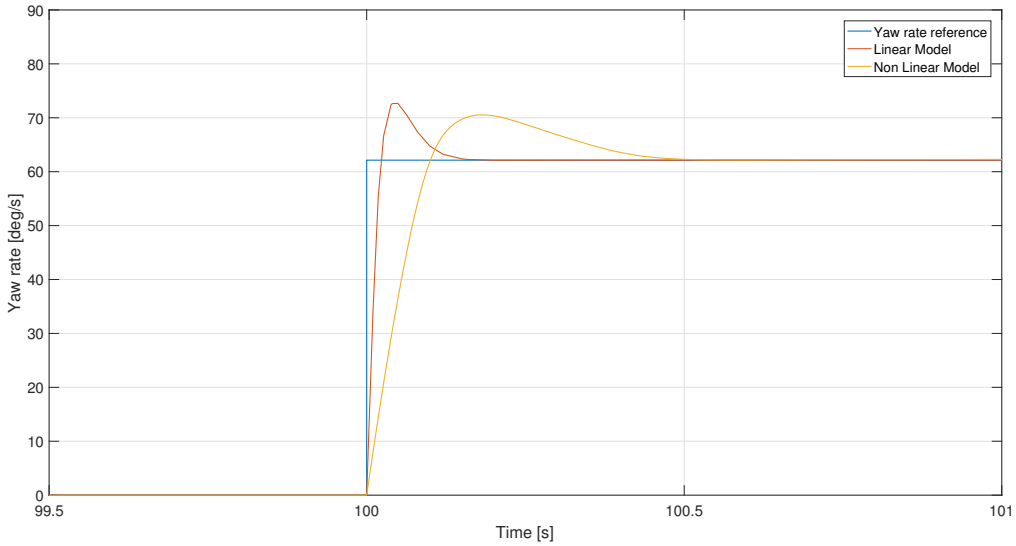


Figure 3.10: Comparison between yaw rate step response from the linear and non linear model, for a given velocity and steering angle

3.4 LQR

Linear quadratic regulator (LQR) is an optimal control strategy for linear systems. In the design of this type of controller an optimal gain K is calculated based on a performance index J.

The advantage of the PI controller is its simplicity in implementation and understanding of what is happening in terms of allocated torque, but this simplicity also has some drawbacks. If the tire-road friction is wrong or the calculated reference yaw rate is excessive for the current state of the vehicle, the vehicle behavior may become unstable.

As an example, if the track is wet the tire-road friction will typically be half [5, p.322]. This means that the torque should also be half, if no change on the controller is made the amount of torque will be too much. Another situation is when the car is sliding, the car is rotating but the yaw rate will stay constant, the controller will not sense the alteration.

To further improve the torque control, a LQR controller is presented. The controller will also use both models previously discussed in chapter 2. Lateral velocity v_y and yaw rate $\dot{\psi}$ are considered state variables and ΔT the control input. The difference is that now it will also be monitored the lateral velocity.

$$\Delta T = K_r \dot{\psi} + K_v v_y \tag{3.12}$$

As mentioned in section 3.1, the yaw rate is measurable by the IMU. The same cannot be said for the lateral velocity, because there is no direct way of measuring it. An online estimation of this state variable is desired. The easiest way of obtaining the velocity is by direct integration of the lateral acceleration. The problem is that by integration, due to noise, non null bias and missing data points, the values of the velocity will drift over time. The best approach is using a Kalman Filter to obtain the velocity.

To calculate the lateral velocity a simplification can be made. If we only test in a skidpad, the car will be in a trajectory with a constant velocity and a known radius, thus under steady state condition. From this it is possible to calculate the lateral velocity of the car. First calculating the longitudinal velocity.

$$v_x = \sqrt{\frac{a_x}{R}} \tag{3.13}$$

Then using the global velocity from the vehicle.

$$v_y = \sqrt{v_{CG}^2 - v_x^2} \tag{3.14}$$

An estimation of the lateral velocity can be obtained.

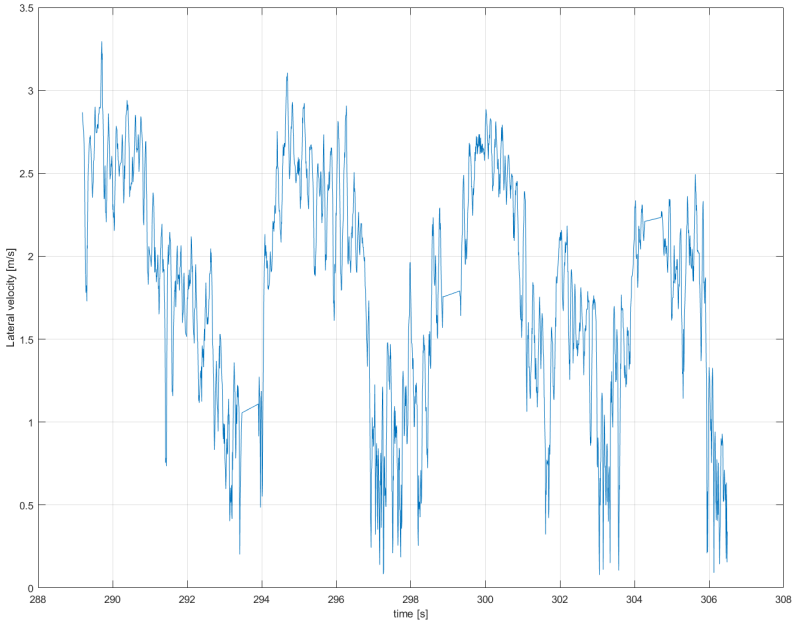


Figure 3.11: Lateral velocity estimation of a skidpad

3.4.1 Optimal controller design

The performance index to be used on the LQR controller design may be written in the following way:

$$J(u) = \frac{1}{2}\int_{t_0}^{t_f}[(X_d - X)^T Q(X_d - X) + u^T R u]dt \tag{3.15}$$

where: u = M_z, R = weight factor of the control effort, Q = Penalty Matrix for the states (v_y, $\dot{\psi}$)

$$J = \int_{t_0}^{t_f}\left[\frac{1}{2}(\dot{\psi} - \dot{\psi}_{des})^2 + \frac{1}{2}w\Delta T^2\right]dt \tag{3.16}$$

where $\dot{\psi}_{des}$ is the desired yaw rate of the vehicle, from section (3.2.1). Minimizing this will lead to a vehicle with very close to neutral steer behaviour. Not forgetting that the control effort ΔT must be constrained both due to the maximum torque possible and the tires limit.

Using the linear model, from which the gains will be calculate, it is necessary to make a small change to the state space matrix. A new entry was added because the system doesn't have an integrator. So the signal is feeded back to the input.

$$\begin{bmatrix}\dot{v_y}\\ \ddot{\psi}\\ \dot{\psi}\end{bmatrix} = \begin{bmatrix} -\frac{C_{y,f}+C_{y,r}}{mv_{x0}} & \frac{-l_fC_{y,f}+l_rC_{y,r}}{mv_{x0}} - v_{x0} & 0\\ \frac{-l_fC_{y,f}+l_rC_{y,r}}{I_{zz}v_{x0}} & -\frac{l_f^2C_{y,f}+l_r^2C_{y,r}}{I_{zz}v_{x0}} & 0\\ 0 & 0 & -1\end{bmatrix}\begin{bmatrix}v_y\\ \dot{\psi}\\ \psi\end{bmatrix} + \begin{bmatrix}0\\ \frac{1}{0.05*I_{zz}}\\ 0\end{bmatrix}\Delta T \tag{3.17}$$

The K gains are calculated by solving the Riccati equation, choosing appropriated values of Q and R. The controller is tuned by varying both values, R and Q.

$$A^T P + PA - PBR^{-1}B^T P + Q = 0 \tag{3.18}$$

Solving for our case will be:

$$\begin{aligned} &2a_{11}p_{11} + 2a_{21}p_{12} - R^{-1}\frac{p_{12}^2}{I_{zz}} = Q_{11} \\ &(a_{11} + a_{22})p_{12} + a_{21}k_{22} + a_{12}k_{11} - R^{-1}\frac{p_{12}p_{22}}{I_{zz}} = Q_{12} \\ &2a_{12}p_{12} + 2a_{22}p_{22} - R^{-1}\frac{p_{22}^2}{I_{zz}^2} = Q_{22} \end{aligned} \tag{3.19}$$

This system can be solved and the optimal feedback gain matrix K will be:

$$K = R^{-1}B^T P = R^{-1}\begin{bmatrix} 0 & \frac{1}{I_{zz}} \end{bmatrix}\begin{bmatrix} p_{11} & p_{12} \\ p_{12} & p_{22} \end{bmatrix} = R^{-1}\begin{bmatrix} \frac{p_{12}}{I_{zz}} & \frac{p_{22}}{I_{zz}} \end{bmatrix} \tag{3.20}$$

Resorting to *matlab* to solve equation 3.19 (with the use of the command "*lqry(sys,Q,R)*"), The matrix is Q = $\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 10^6 \end{bmatrix}$

Starting with an R value of R = 10^{-6}. The gains are K = $\begin{bmatrix} 3 & 473 & -31623 \end{bmatrix}$

Where the last value is the integrator gain.

The response of the controller is within the desired objective.

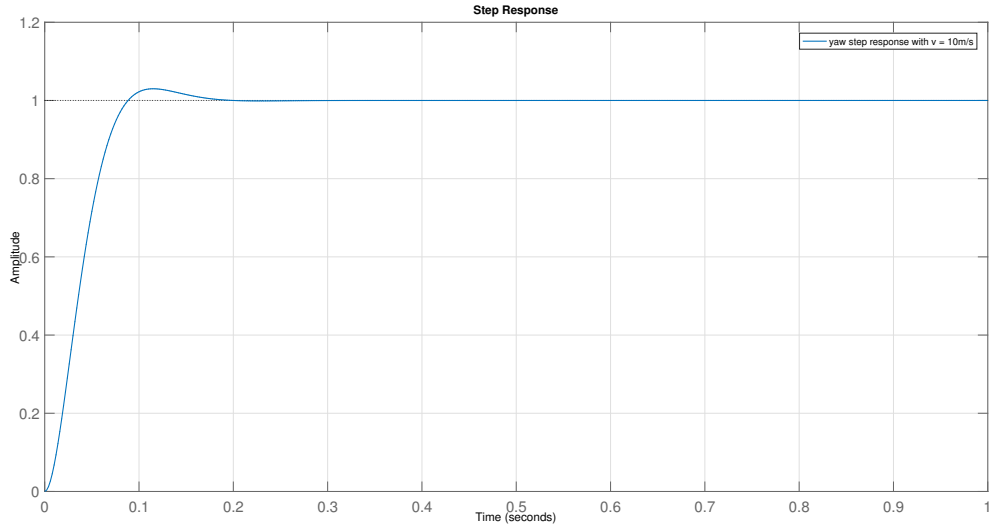


Figure 3.12: Step response of yaw rate ($\dot{\psi}$) for the LQR controller

3.4.2 Gains

As mentioned, this model is velocity dependent, therefore it is necessary to verify if the gains need to be calculated for every velocity v_0 as is the case for the PI controller. Figure 3.13 shows the gains calculated for a range of various velocities [2-20 m/s].

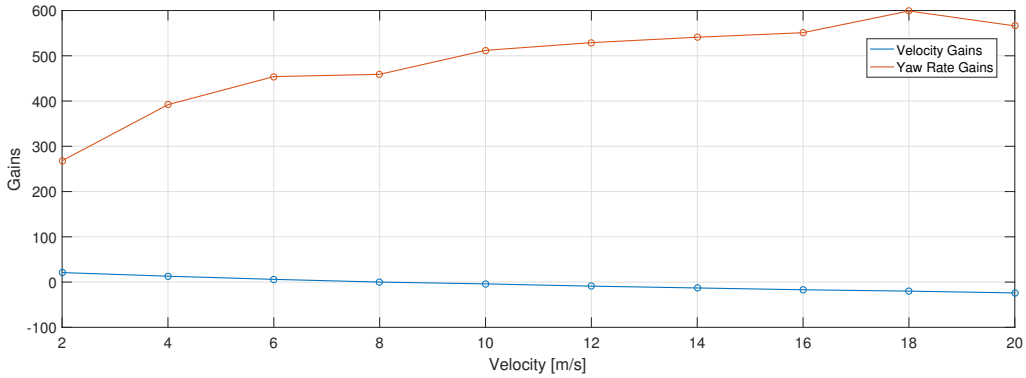


Figure 3.13: Lateral velocity and yaw rate gains for a rage of velocities between 2-20m/s

For working operations (velocity $>$5) the variations in the gains are not very significant, primarily in the lateral velocity gain.

To analyse how the change in gains affects the response of the controller, a comparison between the step response of the system, with gains calculated for each velocity (fig: 3.14) and maintaining the same gains (fig: 3.15) is made.

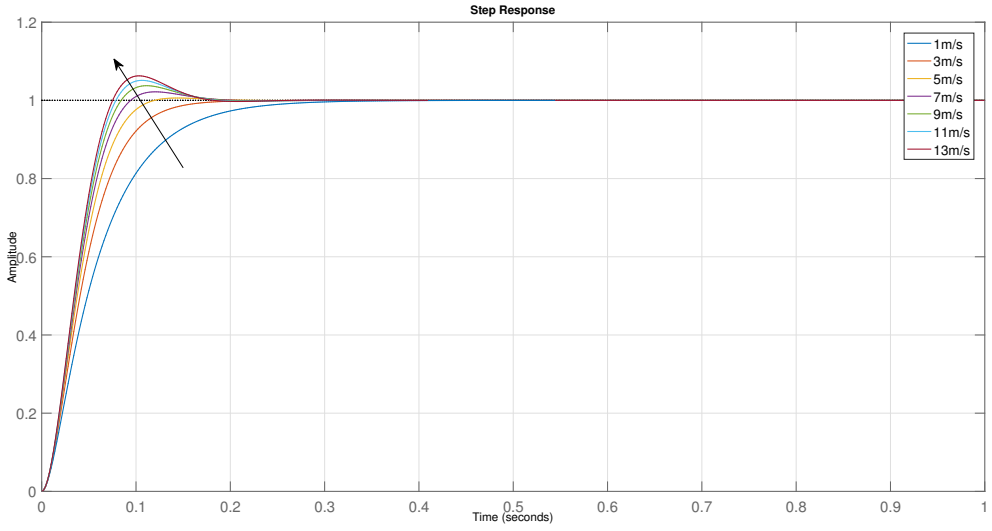


Figure 3.14: Comparison of LQR step response for same gains for different velocities

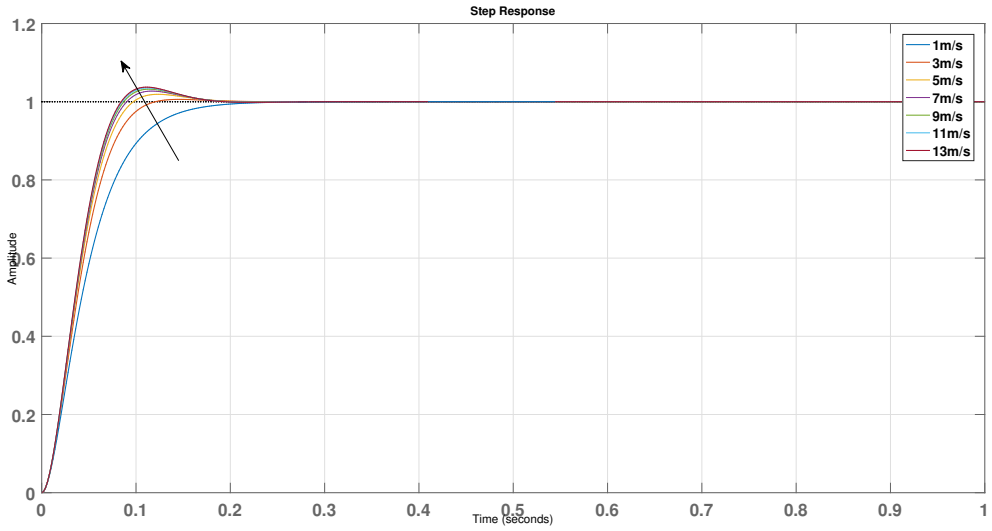


Figure 3.15: Comparison of LQR step response calculating LQR gains for each velocity

Subtle differences are noticeable, for high velocities the system with the same gains has a higher overshoot and for low velocities a low response time when compared with the system that has the gains calculated for each velocity. But the difference are small enough that it won't be necessary to have a table of gains for each velocity, this will further simplify the implementation.

3.4.3 Lateral Velocity

Looking at the gains from lateral velocity and yaw rate, in figure 3.16 and 3.17 it can be noticed that the contribution of the gains are very different. The yaw rate is much bigger than the velocity gains, for this an analysis of the gains is conducted.

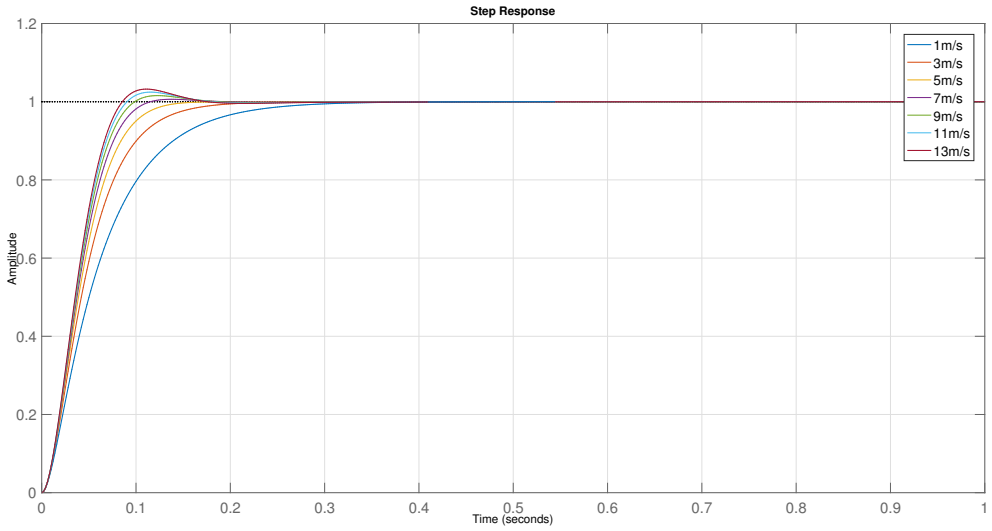


Figure 3.16: Yaw rate step response with the lateral velocity gain

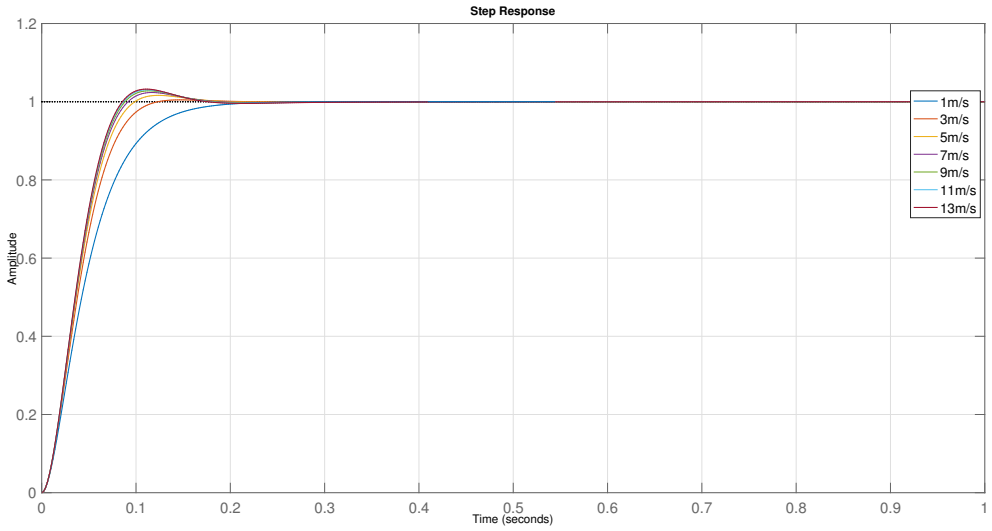


Figure 3.17: Yaw rate step response of model without the lateral velocity gain

It can be seen that for small velocities the lateral velocity gain is not very important when compared to the yaw rate gain. For higher velocities it is more important because sliding effects occur. For the current application it is not important, because we are testing for low speeds.

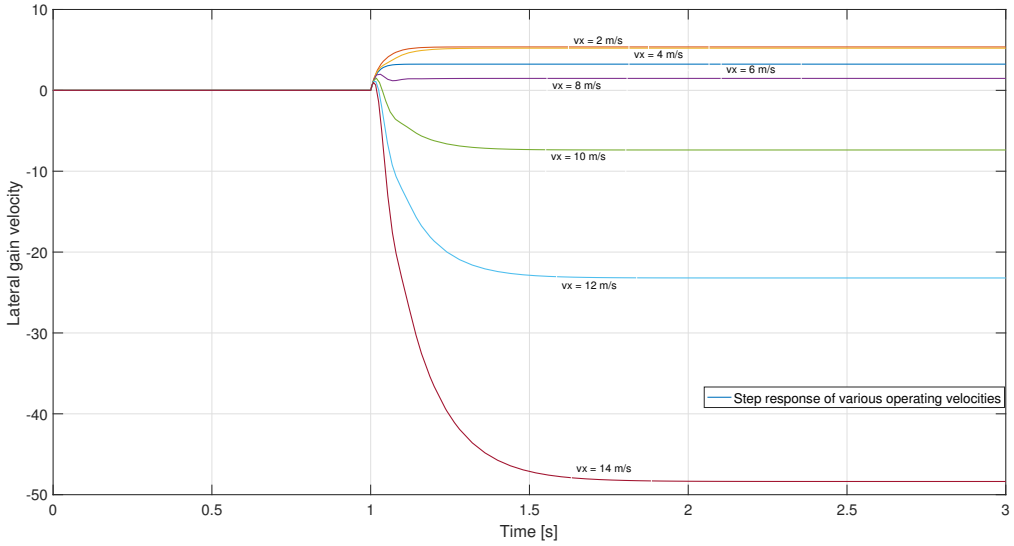


Figure 3.18: Lateral velocity step response gain for various velocities

3.5 Comparison

To conclude this chapter a comparison between the two proposed controllers are presented. First for the linear model, followed by the non linear. The test is based on the test 3 form table 2.15. The tests will be for a range of velocities, but for a fixed steering input of 80 deg as from test 3.

3.5.1 Linear Model

In the linear model, both inputs (steering angle and velocity) are used. The steering angle is a fixed input, while the velocity varies for a range of velocities. The LQR controller presented is with the fixed gains while the PI gains are the ones shown in the lookup table 3.7.

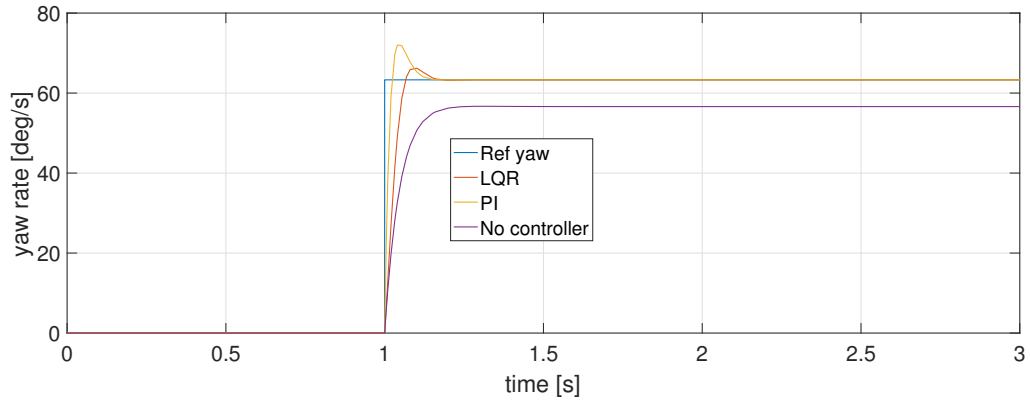


Figure 3.19: Comparison between LQR and PI and no control

Figure 3.20 compares the response of both controllers for a range of velocities (1-11m/s) while maintaining the same steering input of 80deg. The PI controller gains are the lookup table 3.7, and the LQR is with just a gain. As the speed increases also the overshoot of the system will increase.

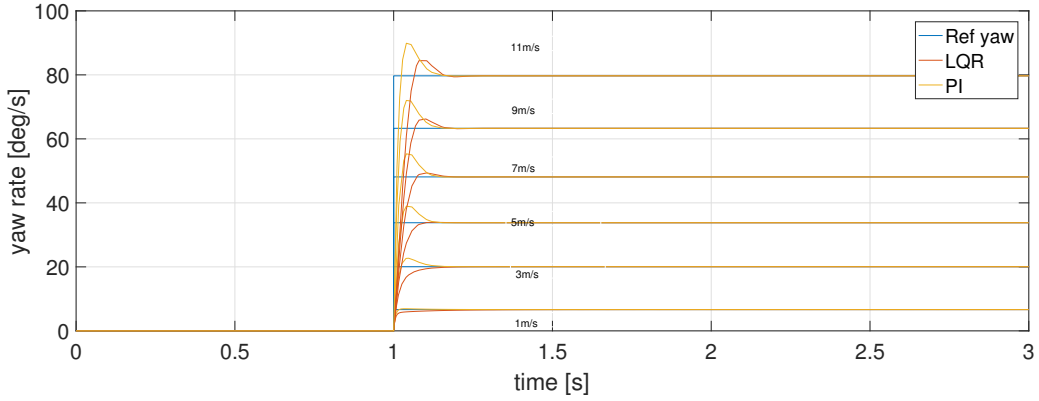


Figure 3.20: Comparison between LQR and PI for a range of velocities (1-11m/s) and a fixed steering input of 80 deg

3.5.2 Non Linear Model

After the design responses of the controller in the linear model, the controller is assessed in the non linear model and their response compared. What can be seen is that the LQR controller achives the same goal has the PI controller but in less time.

Figure 3.21 shows the response of the system. For the same case the LQR controller will have less overshoot and less settling time making a more effective controller.

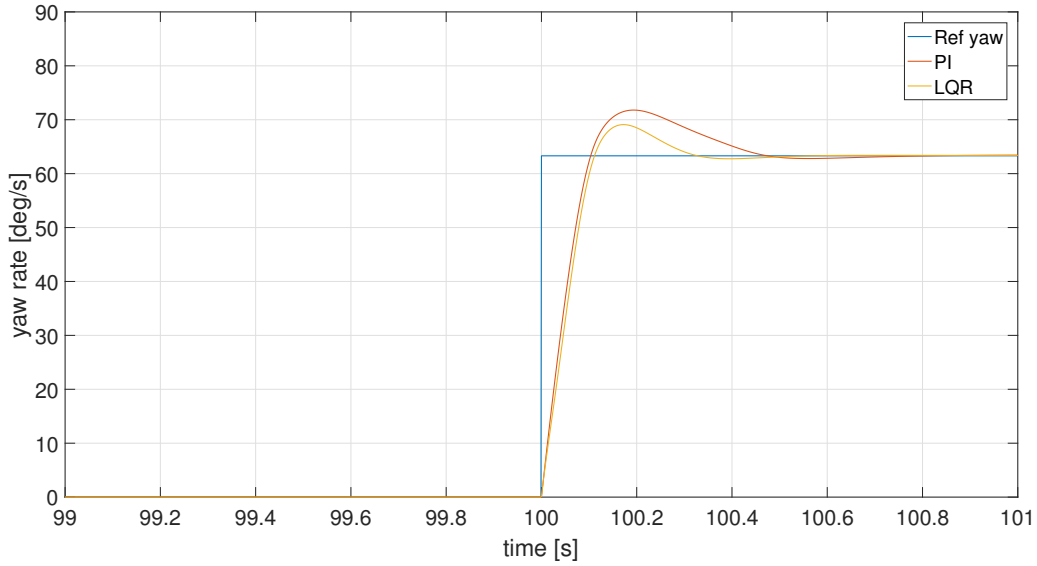


Figure 3.21: Yaw rate response model

Figure 3.22 shows the torque output from both controllers. Like in figure 3.21 the same output has the same characteristics.

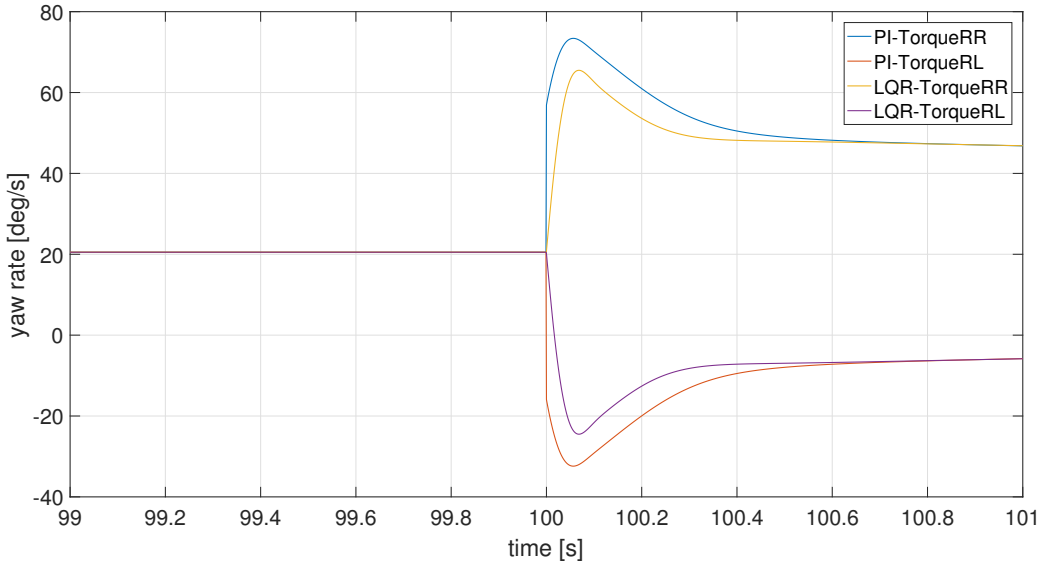


Figure 3.22: Torque request at motor

Chapter 4

Discrete Controller

In the previous chapters, the behavior of the system is described by differential equations. These are in continuous-time, which makes sense given the physics describing the involved phenomena. However, the microcontrollers used to implement the controller in real time operate in the discrete-time domain. To be able to connect both systems, two different approaches are possible.

One can discretize the plant model, and from there design a new discrete controller for that plant, or discretize the control plant from the continuous time to the discrete time.

Due to the simplicity of the linear model in continuous time, it was chosen to also make a discrete model. The main advantage of taking this approach instead of just discretizing the controller in the continuous time (emulation), is that sensor sampling time and noise values can be added to the model in discrete time.

This chapter starts by discretizing the linear model. Sections 4.1.1 and 4.1.2 talks about the controller in discrete time. In section 4.1.3, an analysis of the noise and sampling times is made. The chapter then concludes with an analysis of the controller against real data from the car.

4.1 Discrete Model

With the help of Matlab, given a continuous state space model, it can be easily converted to discrete by appliyng the zero hold method (ZOH), for a sampling time. From the Bode analysis (fig. 3.6) an appropriate sampling time would be between the following values:

$$\frac{2\pi}{20w_b} \leq T_0 \leq \frac{2\pi}{10w_b} = \frac{2\pi}{20*42} \leq T_0 \leq \frac{2\pi}{10*42} = 0.0075s \leq T_0 \leq 0.015s \tag{4.1}$$

Due to the available hardware, the maximum possible sample time was of 0.02s. From equation 2.34, the state space in discrete time will be.

$$\begin{bmatrix} \dot{v_y}(k+1) \\ \dot{r}(k+1) \end{bmatrix} = \begin{bmatrix} -\frac{C_{y,f}+C_{y,r}}{mv_{x0}(k)} & \frac{-l_fC_{y,f}+l_rC_{y,r}}{mv_{x0}(k)} - v_{x0}(k) \\ \frac{-l_fC_{y,f}+l_rC_{y,r}}{I_{zz}v_{x0}(k)} & -\frac{l_f^2C_{y,f}+l_r^2C_{y,r}}{I_{zz}v_{x0}(k)} \end{bmatrix} \begin{bmatrix} v_y(k) \\ r(k) \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{1}{I_{zz}} \end{bmatrix} M_z(k) + \begin{bmatrix} \frac{C_{y,f}}{mv_{x0}}(k) \\ \frac{l_fC_{y,f}}{I_{zz}} \end{bmatrix} \delta(k) \tag{4.2}$$

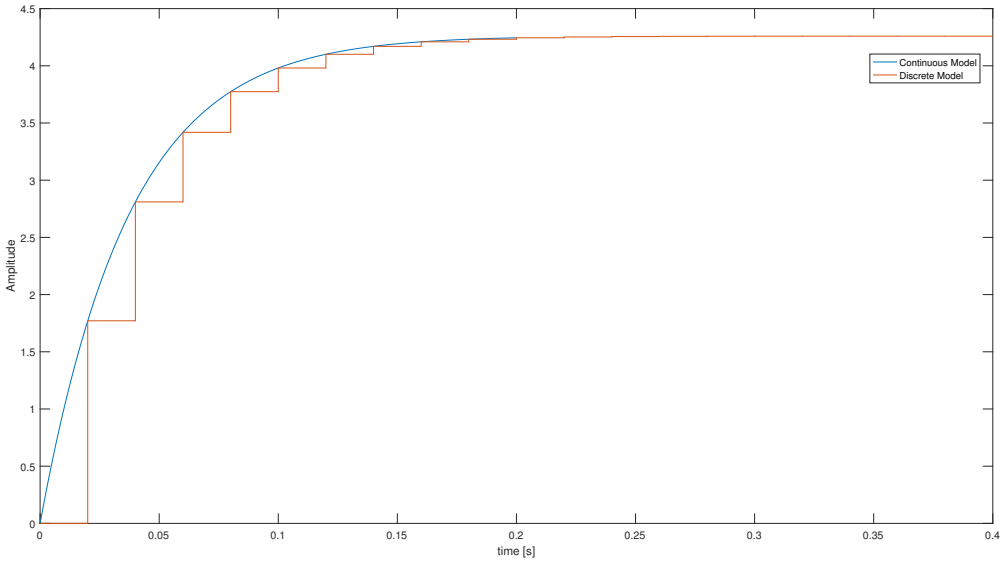


Figure 4.1: Discretization of the linear model with a time sample of 20ms

4.1.1 Discrete PI

For the design of the discrete controller, the same methodology as described in section 3.3 is followed. The final state space equation for the design of the controller is.

$$\begin{bmatrix} \dot{v}_y(k+1) \\ \dot{\psi}(k+1) \end{bmatrix} = \begin{bmatrix} -\frac{C_{y,f}+C_{y,r}}{mv_{x0}(k)} & \frac{-l_f C_{y,f}+l_r C_{y,r}}{mv_{x0}(k)} - v_{x0}(k) \\ \frac{-l_f C_{y,f}+l_r C_{y,r}}{I_{zz}v_{x0}(k)} & -\frac{C_f l_f^2 + C_r l_r^2}{I_{zz}v_{x0}(k)} \end{bmatrix} \begin{bmatrix} v_y(k) \\ \dot{\psi}(k) \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{1}{0.05 * I_{zz}} \end{bmatrix} \Delta T(k) \tag{4.3}$$

The same conditions and requirements of the vehicle apply to this model. Designing the new PI controller, the new look up table will be:

Vx (m/s)	P	I
7	119.52	3855.6
9	127.45	3862
11	139.86	3696.5
13	148.18	3414.2
15	150	3342.6

Table 4.1: Discrete-time values of the PI controller

4.1.2 Discrete LQR

For the LQR controller design in discrete time, given the discrete state space equation in the the previous section, a linear quadratic regulator is used to calculate the discrete gains of the controller. As mentioned before the sample time is also the same, 20ms. The new gains are: $\begin{bmatrix} 10 & 438 & -24900 \end{bmatrix}$. Where the last value is the integrator gain.

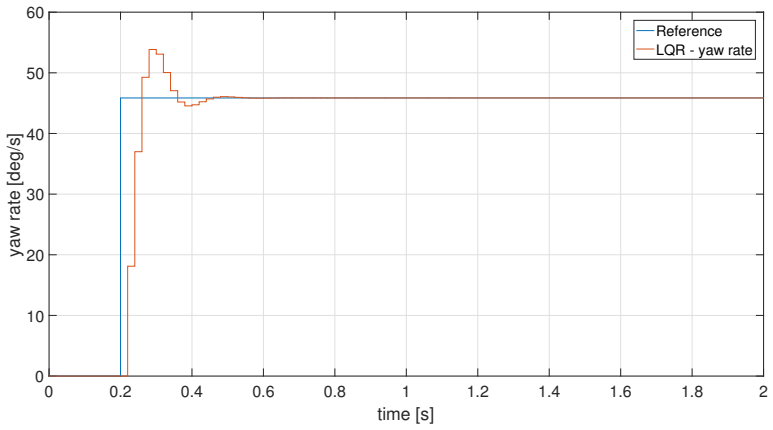


Figure 4.2: Discretized step response for the LQR controller

4.1.3 Noise Values

Followed by the analysis of the sample periods, an investigation to the overall quality of the sensors is also performed. The noise values are mainly from 3 major sources:

- Sensor noise
- Electromagnetic noise
- Vehicle noise

Sensor noise is closely related to the quality of the sensors. Low cost sensors will have a lower resolution and an intensity noise. To evaluate the noise, the sensors are tested with the car stopped. Data is acquired for some time, and then the mean and standard deviation is computed. Table 4.2 summarizes the average and deviation from the sensors taken from the car while stationary.

	Average	Standard Deviation	Units
Yaw Rate	0	0.73	deg/s
Accelerator Pedal	0	4.7	ADC
GPS Velocity	0	0	m/s
Steering Enconder	0	0.34	deg

Table 4.2: Average and standard deviation of the noise values from sensors used in the controller

Because the FST06e is a full electric car couple with 2 big electric motors, when the car is moving the rotation of the motors creates a magnetic field strong enough to alter the sensor values. To avoid this noise it is necessary to shield the sensors. In section 2.8, based on the hand calculations, it is possible to infer that the electromagnetic noise didn't affect the sensor reading.

Another source of noise is the motion of the car i.e, the irregular surface of the road in contact with the car produces noise captured by the sensors. Fortunately, the vehicle operates with a frequency between 1-10 Hz, which are low values that can be filtered if required.

With this new variable considered, the model is update. Noise of the steering input (δ_w) and the output yaw rate ($\dot{\psi}_w$) are added to the model, from table 4.2.

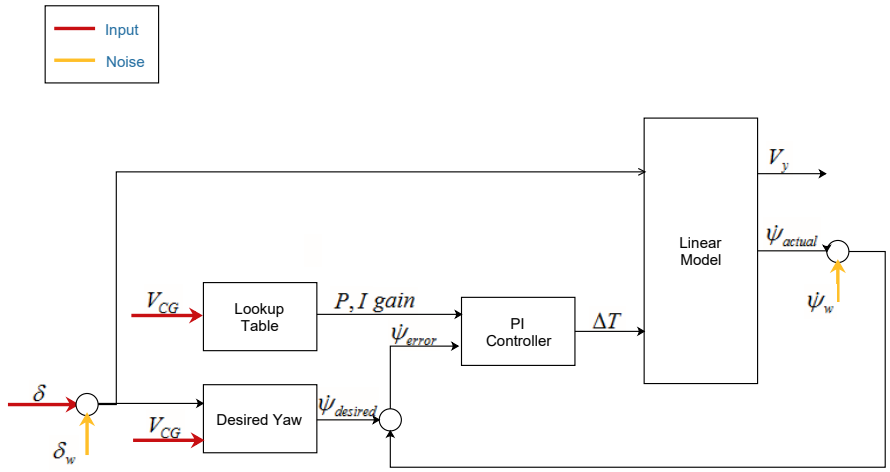


Figure 4.3: Linear model schematic with added noise

4.1.4 Real Data Values

Figure 4.4 shows the step response of the yaw rate for the PI and LQR controller, for a steering angle of 71 deg/s and a velocity 9 m/s (same values from ”Test3”), with added noise for both inputs. Figure 4.5 shows the response of the vehicle with steering angle and velocity data values, and with added yaw rate noise values from table 4.2.

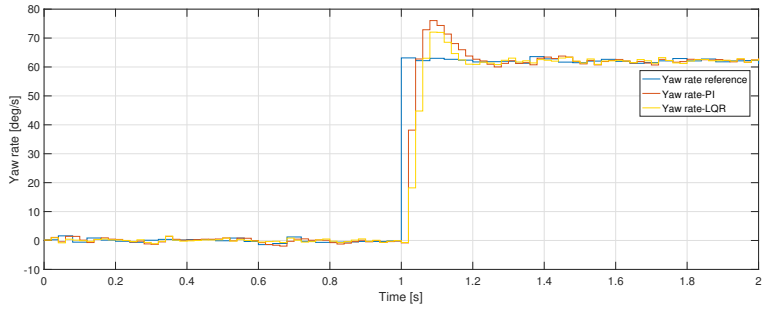


Figure 4.4: Step response of PI and LQR controller with added noise from table 4.2 at input (δ, v) and output $\dot{\psi}$

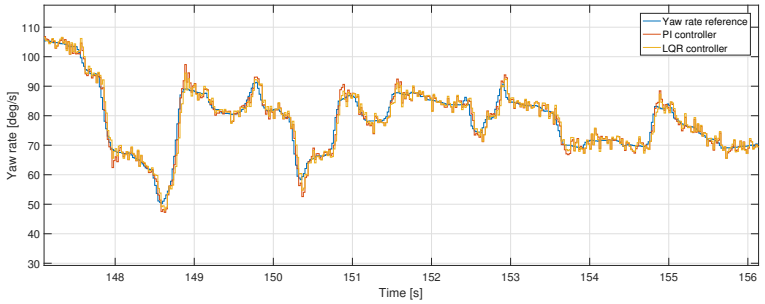


Figure 4.5: Linear simulation with steering and velocity data as inputs, and yaw rate as the output, comparing with and without the PI controller

Chapter 5

Implementation

In chapters 3 and 4, the control system is discussed, as well as the discretization of the model. The next step is the implementation of the controller. To implement it is necessary to convert all the code done in Matlab/Simulink to C code. The implemented code is then compiled into the torque vectoring board. This board is a printed circuit bord (PCB) that has the necessary elements to communicate with the car. Converting Matlab scripts to C code is not straightforward. In Matlab all the models are coherent with SI units, the sensor's output and the actuator's input are both DC voltage (i.e. one cannot request 10Nm from the motors).

The FST06e works in ADC values, which in practice means that what is going to be controlled is the ADC value that comes from the pedal (requested by the driver). This value enters the Torque Vectoring PCB, and the algorithm then decides the value which should go to each motor, sending an ADC value based on the current state. In between the pedal, the PCB and the motors there is a dead zone and a saturation that has to be taken into account. The first one is a lower limit dead zone, which ensures that, although the pedal spring may not always return to the same position, the car will not start as soon as one hits the pedal. The second is a saturation of the maximum value of torque that can go to the motors. Figure 5.1 illustrates the path from the pedal value (driver request) to the motors, with the respective dead zone and saturation.

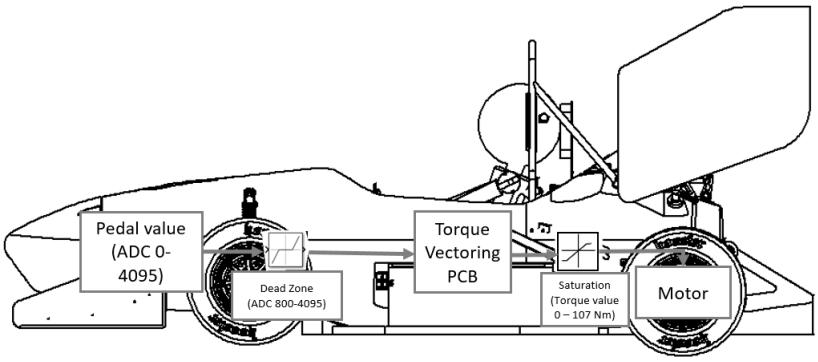


Figure 5.1: Scheme from the pedal value to the motor value

5.1 Communication with the FST06e

The FST06e communication protocol is CAN protocol. A Controller Area Network (CAN) is a specialized internal communications network designed to allow microcontrollers to communicate with each other without the use of a host computer. The main characteristics are:

- CAN is an extremely robust serial communication protocol. Any CAN node on the CAN bus can detect errors on the messages, force it to be destroyed and resend it.
- Message based, not address based. Every node receives the same message, and it is up to the code to act on that data.
- Distributing control across the CAN network that allows peer to peer or master to slave style of communications

The protocol uses the following format:

$$[ID|ID|DLC|Data0|Data0|Data1|Data1|Data2|Data2|Data3|CRC|CRC]$$

- ID - message identifier. (2bytes)
- DLC - Data Length Code, number of bytes of data (0-8) (1byte)
- Data0 - Data to be transmitted (2bytes)
- Data1 - Data to be transmitted (2bytes)
- Data2 - Data to be transmitted (2bytes)
- Data3 - Data to be transmitted (2bytes)
- CRC - Cycle Redundancy Check (2bytes)

The data is of type uint16 (2 messages of uint8). With this format, the possible values are [0, $2^{16} - 1$] = [0, 65355]; Once data is received, the first step is to merge into int16 values. In int16, the possible value ranges from [-32768, +32767]. Because each sensor has a specific resolution and format, it is necessary to multiply or divide accordingly. After all the calculations are done, it is necessary to convert again into uint8 to send back to the CAN line. Figure 5.2 demonstrates the process, and table 5.1 shows the sensor data that is of interest and its conversion factors to SI units.

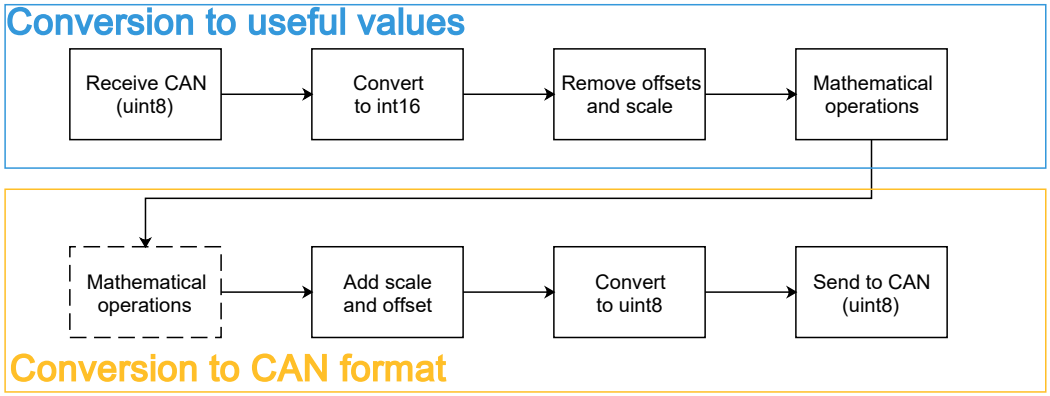


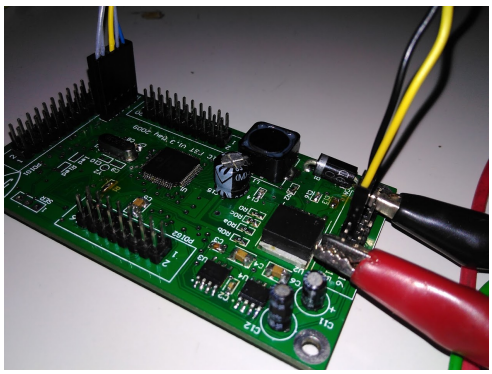
Figure 5.2: Conversion process

Sensor	Aquisition	ID	Data	Conversion Factor	Output
IMU	Acceleration in x direction	900	Data[3]	1/9.8	m/s^2
	Acceleration in y direction	900	Data[2]	1/9.8	m/s^2
	Yaw rate	901	Data[1]	0.0175^2	rad/s^2
GPS	Global velocity	640	Data[3]	0.01	m/s
Torque encoder	Rear left wheel	1155	Data[3]	0.01	Nm
	Rear right wheel	156	Data[2]	105.3/2^14	Nm
Wheel encoder	Front left wheel	913	Data[2]	0.01	RPM
	Front right wheel	913	Data[0]	0.01	RPM
	Rear left wheel	1151	Data[2]	8500/2^14	RPM
	Rear right wheel	1152	Data[2]	8500/2^14	RPM
Steering encoder	Steering angle	150	Data[1]	0.1*0.0175	rad

Table 5.1: Location and conversion factor of the sensors values

5.1.1 Open Loop Communication

The use of this protocol is made possible by using a self develop board with the DSpic6012A microcontroller and two CAN buses. Besides receiving and sending messages, the microcontroller runs the PI controller and the CAN buses send and receive the information. This microcontroller can run up to 112 MHz, and it has a floating point unit, which is particularly useful when performing calculations. The two CAN buses are needed to communicate between the the sensors and the motors.



(a) DSpic6012A development board

(b) Microchip used to program the DSpic6012A

Figure 5.3: a) Development board b) compiler

Once the C code is made, it is necessary to verify if it is sending the correct values. Some small operations like the computation of the yaw reference, sending and receiving messages can be checked directly by sending a known input and checking if the output is correct. For the case of the PID controller and more complex control algorithms it is not possible to just send a known value, it is necessary to constantly update the states based on the inputs given by the controllers. To address this problem a real time simulation is developed.

Using Simulink real time package it is possible to communicate in real time with the development board. This functionality is quite useful when writing code. It can be verified if the development board is sending and receiving data in the correct format, and if the computations are correct. This reduces the total amount of time that is needed to test in the car. Besides the use of the real time simulation, it was also necessary to setup a hardware workstation. Figure 5.4 shows the schematic.

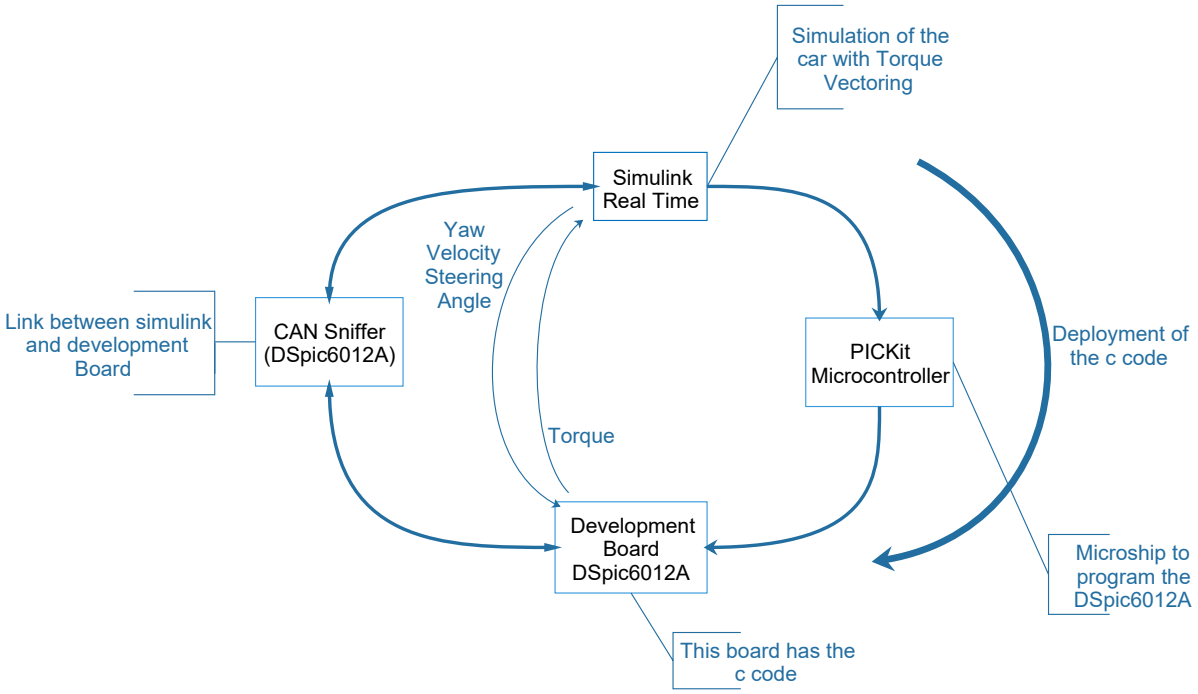


Figure 5.4: Schematic of the workstation

Once the code is done, the development board sends the necessary input data from the real vehicle (pedal input, yaw value and steering) to Simulink. From the received data the PCB calculates the yaw rate, velocity and steering angle, which it sends back to the development board, making it possible to see how the controller acts in a simplified model of the car.

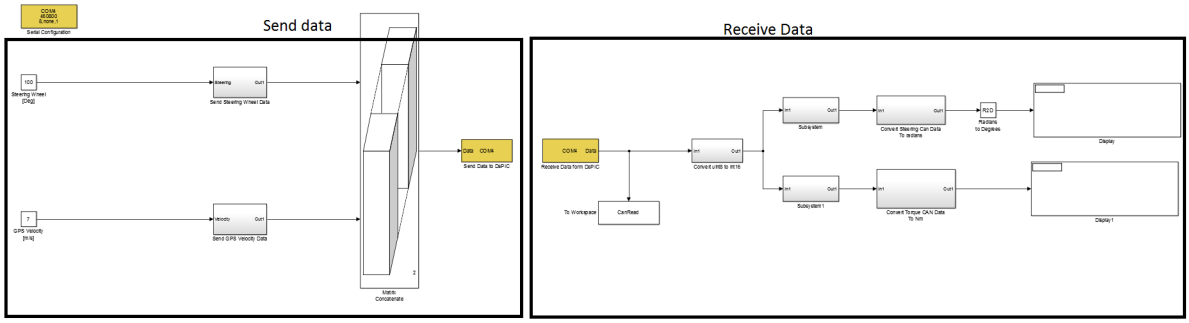


Figure 5.5: Simulink model for receiving and sending data to the PCB board. On the left the model to receive data, on the right the model to receive the data

With the simulation it is possible to confirm if the format of the data sent and received is correct, as well as the computations. This setup helped reduce the amount of testing time necessary with the car. Figure 5.6 shows the data from an open loop test. The data feed is the velocity of the vehicle and steering angle, and the measured values is the desired yaw. With this test it was possible to verify what would be the desired yaw for this test, and if in any occasion there would be an error.

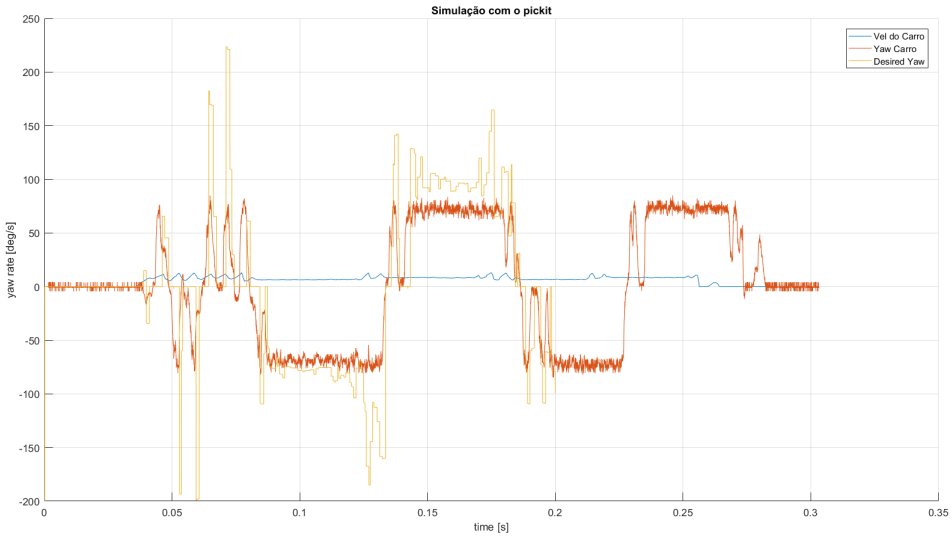


Figure 5.6: Open loop simulation with the microcontroller. Steering and and velocity as inputs and the desired yaw rate as an output

5.2 Code Implementation

5.2.1 Fail Safe System

After the code is made and checked it is also necessary to create a system of fail-safes. These measures are necessary because prototypes can be dangerous, and when testing a new control algorithm for the first time it is necessary to implement fail safe systems. These types of systems not only ensure that in case of sensor failure or communication problems with the microcontroller, the controller does not became unstable, but also that when the car is stopped or pedal value is zero there is no torque applied

- **Denominator 0** - The most basic fail safe is to guarantee that no division has the value 0 in the denominator (for instance the case of the yaw reference). To address this problem an if case is implemented, when the velocity is zero the yaw reference will return zero.
- **Velocity $<$5** - When the car has a speed lower than 5m/s, the output of the torque vectoring is zero. This ensures that when the car is at low speeds, ex: when parking or moving to the track, no torque vectoring is active.
- **Number of messages** - Because the CAN protocol is very complex and has a lot of messages and priorities to handle at once, it may happen that sometimes some messages can't be sent, or be corrupted. Given this possibility, some sensors can still have old values. To protect against this problem, a routine was created to check the total number of messages received. If the total number of messages received is below a certain threshold the Torque Vectoring is deactivated.
- **Maximum allowed torque** - Depending on the test track it is not always possible to have the maximum torque available. Therefore, a parameter that regulates the maximum torque that comes out of the controller is introduced.
- **Pedal torque:** - The last safety measure is to ensure that when the driver lifts his feet from the pedal, the torque vectoring is not active. This safety measure is implemented because a normal Formula Student driver is always pressing the throttle pedal while cornering.

5.2.2 Code

With the communication and fail systems presented, and also the controller, this section assesses the necessary steps in the development and implementation of the C code. The approach can be divided in the following steps:

- Receiving messages

  Conversion from ADC to SI Units

- Calculations

  Fail Safe System

  Reference Yaw

  Controller Torque Output

- Send Message

  Conversion from SI Units to ADC

By dividing the code into independent functions, it is easier to debug and to use for further applications.

Figure 5.7 shows the workflow of the implementation of the controller in C code. In the yellow box, as described in section 5.1, the car sends its values in ADC, then converts to SI units. Every 20ms, it moves to the blue box where it verifies if all the necessary messages are present to calculate the torque. If there are not enough messages, it skips the calculations and the torque given from the pedal to the motors. If the necessary number of messages are present, the code executes the calculations, first computing the reference value followed by the torque. In the green box, it checks the value of pedal, checks the value of the pedal, and returns the new torque value if it is higher than zero or the previous value if it is zero.

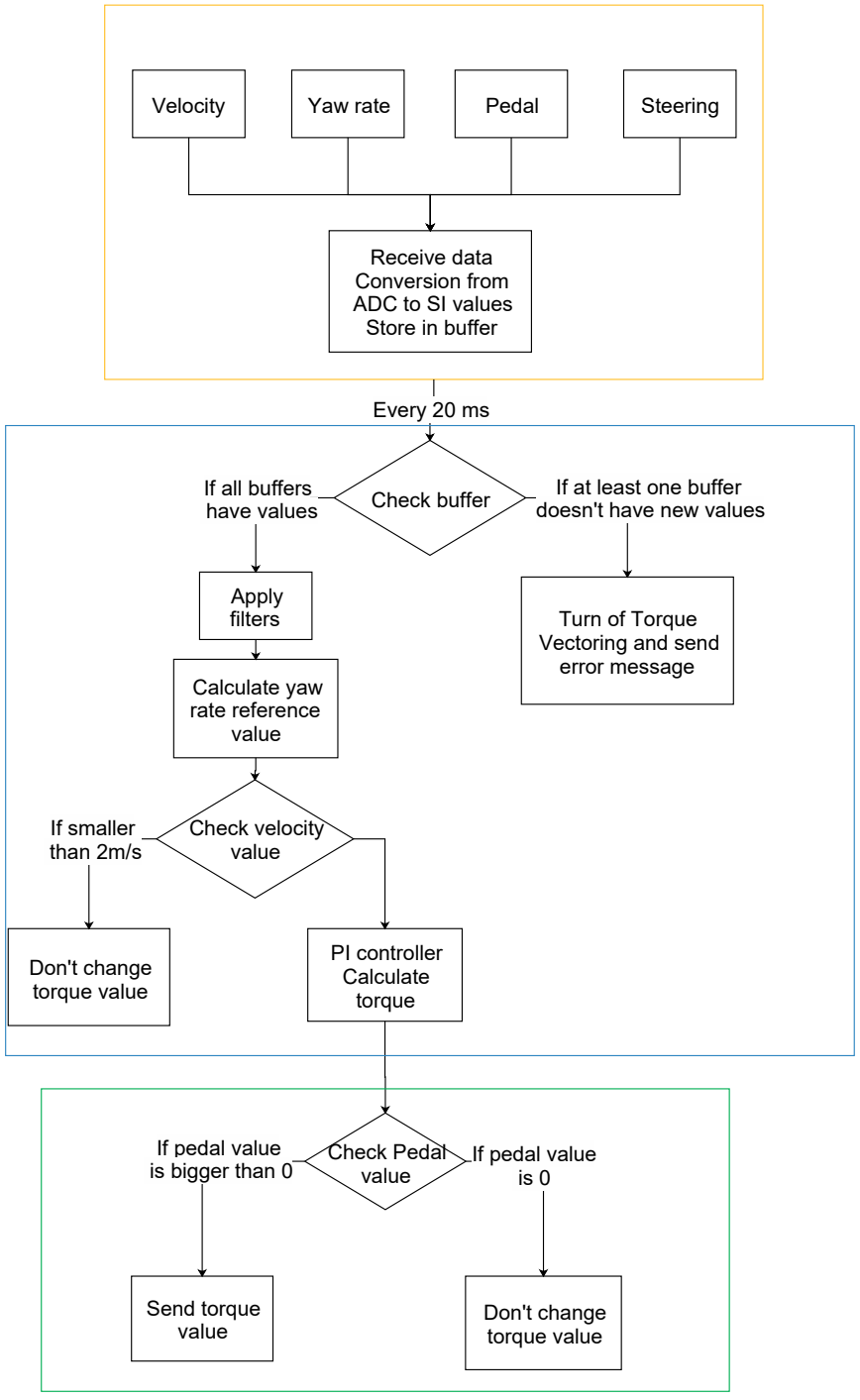


Figure 5.7: Schematic of the implementation of the C code

5.3 Filter Sensors

During the tests with the FST06e, two problems appeared, the first was regarding the GPS. The fact that the GPS loses its signal proved unreliable to use it as the measurement of the car's velocity, often the controller turned off because it lost the GPS signal. To address this problem, the wheel encoder of the front and rear wheels is used. With this it is possible to obtain the vehicle's speed if the GPS signal is not available. In the end it was more practical to use the encoder speed.

Also, the yaw rate measured from the IMU proved to be to noisy when the car is being driven. The noise present, made the controller unstable. To solve this problem a low pass filter was designed to filter the data.

5.3.1 Low Pass Filter

Figure 2.14 shows the yaw rate from the IMU. After some testing it was concluded that it was necessary to use a 3 point median filter and a 1st order low pass filter. The median filter is necessary to remove the outliers, then a low pass filter with a cutoff frequency (f_c) of 3Hz proved to be sufficient to remove the noise (because the car is a mechanical model, the working frequency is very low).

The first order low pass filter is just the current yaw rate $\dot{\psi}(i)$ multiplied by a time constant(α) summed with the previous value $\dot{\psi}$ multiplied by $(1-\alpha)$. The time constant α is a value that is chosen and tuned based on the cut-off frequency (f_c) until the filter is within acceptable values. The time variation depends on the acquisition rate of the IMU, for the FST06e it is 8ms. Figure 5.8 compares the raw data form the IMU and the data after being applied the filter.

$$\dot{\psi}(i) = \alpha\dot{\psi}(i) + (1-\alpha)\dot{\psi}(i-1) \tag{5.1}$$

$$\alpha = \frac{dt}{\frac{1}{2\pi f_c} + dt} \tag{5.2}$$

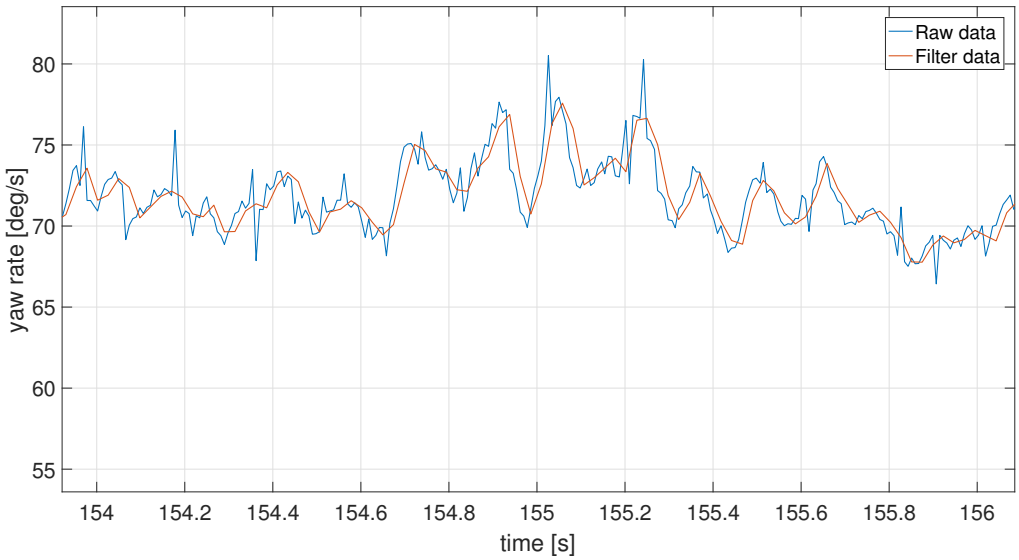


Figure 5.8: Comparison between the raw data and the filtered data for the yaw rate from the IMU

5.3.2 Velocity Measurement

To be able to use the wheel encoder to obtain the vehicle's speed it is necessary to transform the velocity that is measured at the wheel (in the wheels' coordinate system) to the vehicle's coordinate system. This transformation is given by:

$$
\begin{aligned}
{}^{I}v_{FL} &= \left[{}^{B}v_{FL} + \dot{\psi}\left(\frac{b_f}{2} - l_f\beta\right)\right]\cos(\delta_w - \beta) \\
{}^{I}v_{FR} &= \left[{}^{B}v_{FR} - \dot{\psi}\left(\frac{b_f}{2} - l_f\beta\right)\right]\cos(\delta_w - \beta) \\
{}^{I}v_{RL} &= \left[{}^{B}v_{RL} + \dot{\psi}\left(\frac{b_r}{2} - l_r\beta\right)\right]\cos(\beta) \\
{}^{I}v_{RR} &= \left[{}^{B}v_{RR} + \dot{\psi}\left(\frac{b_r}{2} - l_r\beta\right)\right]\cos(\beta)
\end{aligned}
\tag{5.3}
$$

For low velocities it can be assumed that the side slip is very small ($\beta \approx 0$). Also, in the case of the FST06e, the velocity at the wheels coordinate system is in RPM. The conversion from RPM to m/s is made in the following away:

$$
{}^{B}v_{FL|FR} = RPM * \frac{\pi}{30} * Rw \tag{5.4}
$$

and in the case of the rear wheels they still need to be divided by the gear ratio, because the encoder is measuring the speed of the motor.

$$
{}^{B}v_{RL|RR} = \frac{RPM}{G_r} * \frac{\pi}{30} * Rw \tag{5.5}
$$

This way it is possible to not depend on the GPS signal for measuring the velocity of the vehicle. The

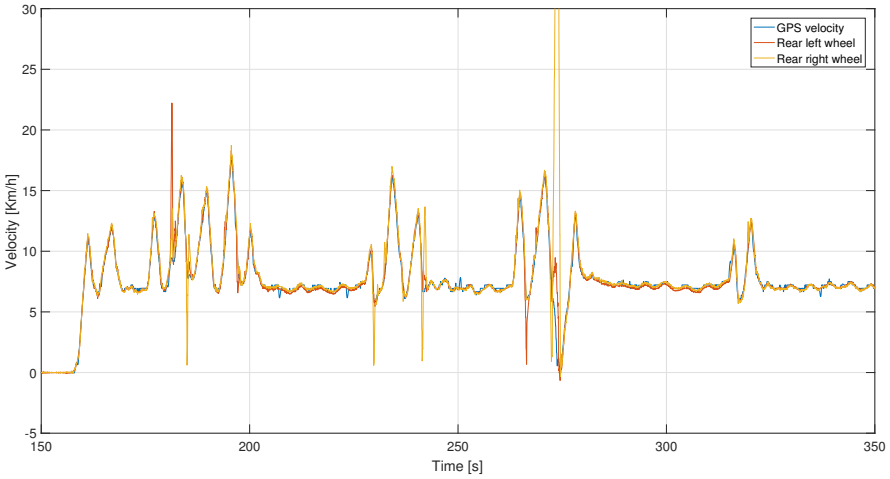


Figure 5.9: Data from GPS and left and right rear wheel, both where converted to km/h

disadvantage of using the wheel velocity is that in case of slippage or lock up the values of the velocity can be wrong.

Chapter 6

Results

In this final chapter the results of the implemented controllers are discussed. As in section 2.8, the car is tested in the same conditions. The driver performs a skidpad with a radius of 5m trying to match the same conditions that are used to validate the model (same speed and steering angle), as in figure 2.11. With the data acquired with the different controllers, a comparison is made, and the gain in vehicle performance evaluated.

First, the data from the car with no torque vectoring is presented. This serves has a baseline to check if the calculations of the reference and output are correct, and performing as expected. For instance, when the controller is not active it is expected that the output saturates, so when this happens we need to see that the integral part of the PI controller will stop integrating.

A series of tests with the torque vectoring were performed, the driver ran some laps in the car and, based on the feedback, tried to push the car even further in order to check if the gains were possible.

Also, the LQR controller is tested, this test also compares with the baseline and PI controller in order to assess which one of the controllers is the best.

6.1 Test Setup

The tests are performed in front of the main build at IST Técnico. In order to be the most cost effective with testing time and the car's battery, a dropdown menu is added to the FST 06e interface which allow the user to select the controller, the gains and the sensors to use. Because the car does not have a direct way to see the data in real time, and having to stop the car to download the data takes too long. All the tests are done at once and the results seen at the end of the session.

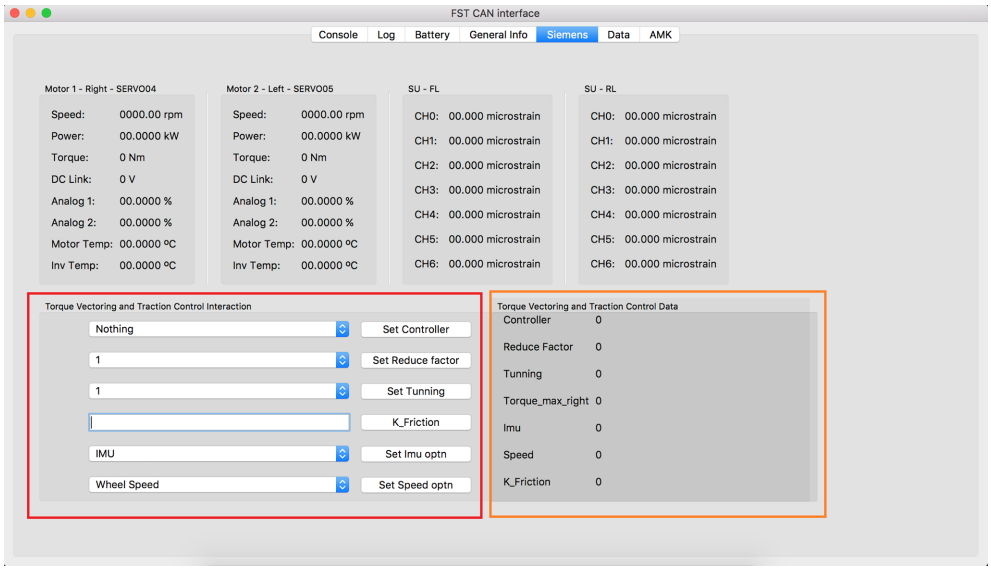


Figure 6.1: Interface of communication between the user and the car. In the red box the user can select which sensors or controller to be active. In the orange box the user can see the current mode activated

6.1.1 No Torque Vectoring

Figure 6.2 shows the variables, desired yaw rate (blue color), current yaw rate (red color), and the speed of the vehicle (yellow color). The driver does 5 laps of approximately 5s each (80s-110s) to the left, cornering in a counter-clockwise way, and from 110s-120s the driver exits and starts cornering in the opposite way in a clockwise way, from (120s-150s).

Recalling equation 3.7 the reference is calculated based on the velocity and the steering angle. If both are constant then the desired yaw will also be constant. Because the velocity is constant, the desired yaw rate variations that appear are the steering corrections made by the driver. It can be observed that in the first laps (80s-100s) the inconsistency of the driver is diminishing until at (100s-110s) the driver is constant. The same happens from (140s-150s).

Focusing now just on the current yaw rate value and desired yaw rate value it can be observed that the current value of the yaw rate is 70 deg/s, and the desired yaw rate is 81deg/s, which means that with torque vectoring there can be a possible gain of 11deg/s. Also, it is important the see that if we look at test1 from table 2.15, the yaw rate is quite similar, which makes sense because the test track is the

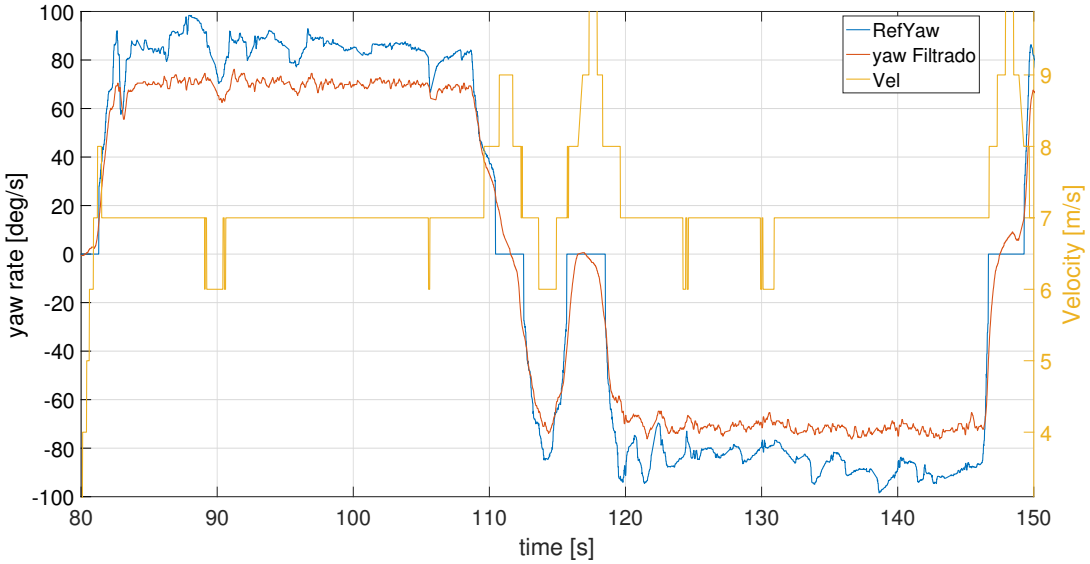


Figure 6.2: Data logged form the vehicle during the test with no torque vectoring. The variables presented are: Yaw rate reference, yaw rate, and global velocity

same.

6.1.2 Torque Vectoring

After the baseline test is performed, the next setup is to make the same trajectory as with the controller. Figure 6.3 shows the same variables as in figure 6.2, but this time the torque vectoring controller is acting on the vehicle. The data presented is from the car cornering to the right, clockwise way.

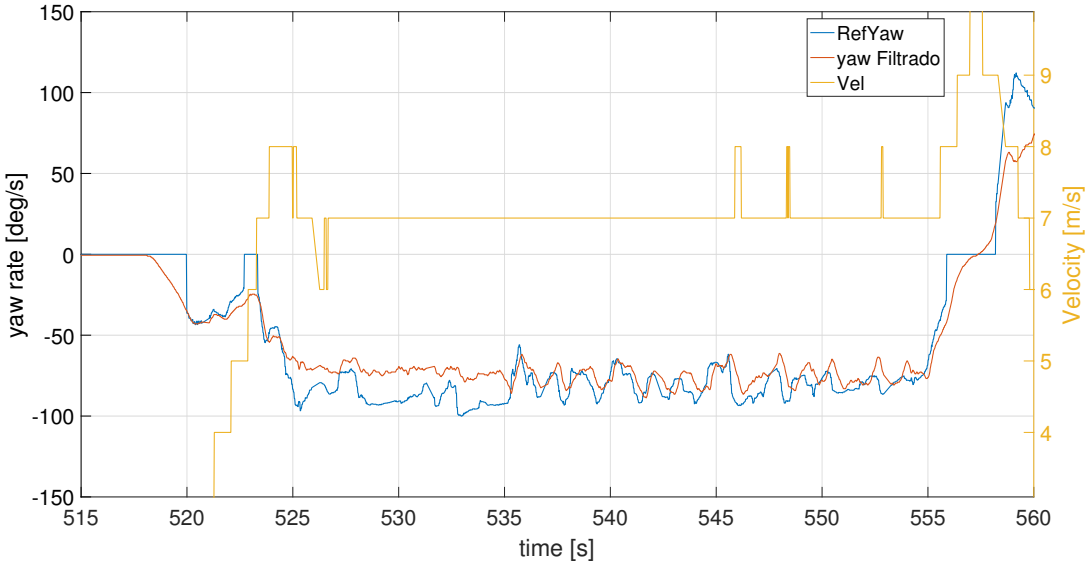


Figure 6.3: Data logged from the vehicle during the test with the PI controller. The variables presented are: Yaw reference, yaw rate, and global velocity

As the driver is starting the corner, like in figure 6.2 at first (525s-535s) the driver is inconsistent but starts to be more consistent the more time he is in the corner (535s-555s).

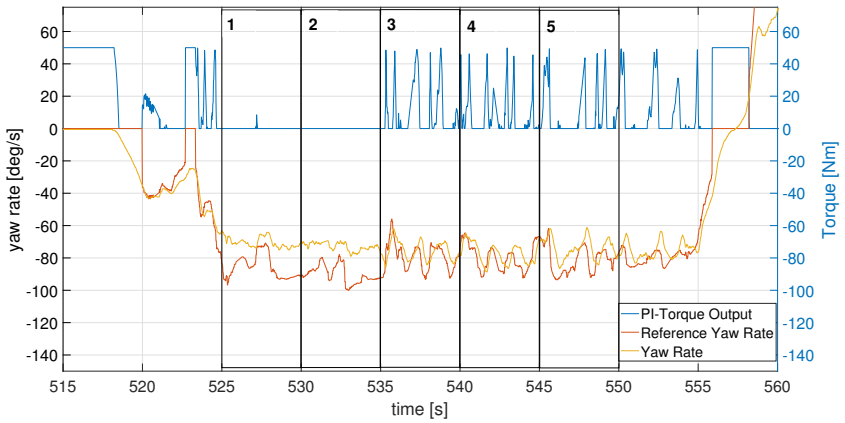


Figure 6.4: Data logged from the vehicle which shows, PI - controller output, desired yaw rate and yaw rate

As a final remark, figure 6.5 is a close up of figure 6.4 where we can see that if the controller is not given any big or small values, the response of the system is approximately 0.2s.

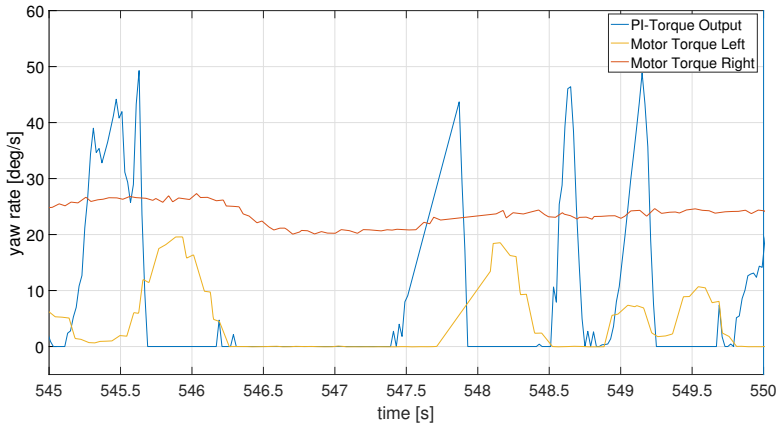


Figure 6.5: Closer view from figure 6.4. PI-Controller is the torque sent to the motors, Motor Torque Left is the torque at the left wheel and Motor Torque Right is the torque at the right motor

Figure 6.6 shows the torque given to each wheel (yellow and purple color). What can be seen is that effectively the controller is changing the torque in each wheel depending on the drivers input, such that when the car is producing too much yaw moment the controller puts more torque in the wheel, and when the car is not having enough yaw moment, the controller removes torque. Also in the same figure in blue and red colors is the simulated torque requested. After the tests, the same inputs were given to the model, to check if the simulation was near the desired results.

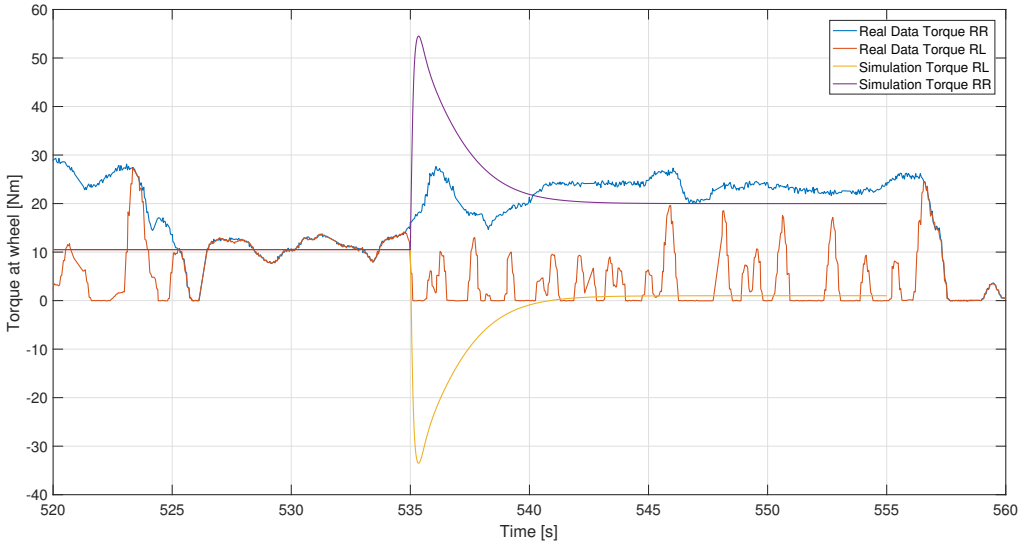


Figure 6.6: Comparison of torques between simulation and real data from the car

Typically to evaluate the performance or increase in performance of a car, it is common to use the gg diagram. This diagram is a graph that in the x label has the longitudinal acceleration and in the y label the lateral acceleration, both in g's. The higher the g's, the better the car will perform, so the goal is to try and increase the gg diagram.

Figure 6.7 shows the gg diagram of the vehicle with torque vectoring (blue color) and without torque vectoring (red color). It can be seen that with torque vectoring the vehicle achieved a higher lateral g, thus increasing its performance.

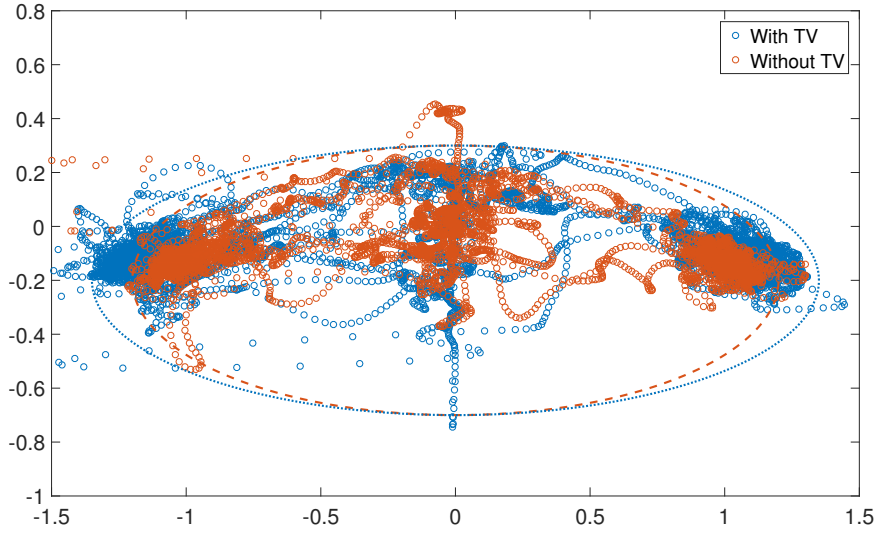


Figure 6.7: GG diagram, which compares the lateral acceleration of the FST06e with and without torque vectoring

6.1.3 LQR

Once the tests of the PI controller are done, the LQR controller is tested. Due to tire wear and electronic failure it was not possible to test the LQR controller in the FST06e. Figure 6.8 compares the real data with the PI controller and the simulation of the LQR controller, it can be seen that the LQR controller would be very similar.

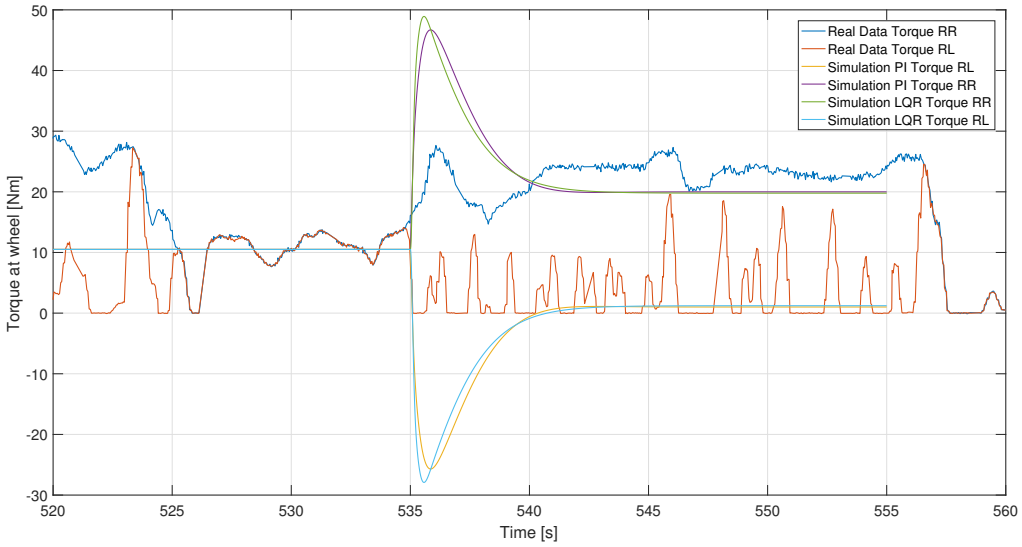


Figure 6.8: Comparison, between simulation and real data values for the LQR and PI controllers.

6.2 Summary and Conclusions

This thesis reported the design of a torque vectoring application for an electric Formula Student prototype, the FST06e, with the objective of reducing the response of the vehicle while maintaining stability. For this purpose, a PI and a LQR control architecture were proposed for enhancing the performance.

For the torque vectoring design process and simulation, the self-developed linear and non linear models, using the principles of vehicle dynamics, were validated through a series of tests. The full model was divided into sub-modulus: horizontal model which described the model position and orientation of the vehicle; a model of the steering kinematics, which describes the relation between the steering wheel and the actual steering of each wheel; a simplified tire model to calculate the forces acting on the wheels; and a balance at the wheel between the applied torque and the tire longitudinal force.
The linear model is a simplification in which only the lateral forces acting on the model are considered. A new term M_z is added to model the additional momentum given by the torque vectoring controller. A relation between the difference in torque and the yaw moment is developed and added to the model.
From figure 2.17 it is possible to conclude that the logged data correlates with both models. Some differences are noticeable, due to the simplifications of the model and also from using a real model in a real track. The noise is present but tendencies can be noticed.

Chapter 3 starts with an analysis of the quality and availability of the FST06e sensors. Based on the available sensors, a control scheme was chosen as well as a reference value. From there, a PI controller was proposed for controlling the wheel torques based on the yaw rate $\dot{\psi}$. Then a LQR approach is analysed to also control the wheel torque based on the lateral velocity v_y and yaw rate $\dot{\psi}$. It was concluded that the lateral velocity gain of the LQR for low velocities did not affect the performance of the vehicle, and in terms of response it was quite similar to the PI controller. The main advantage of the LQR is its robustness and lack of a gain scheduling when compared to the PI controller.

A discretization of the linear model is discussed in chapter 4. The advantage of being able to feed the simulation with real data (noise values and sampling times) proved useful to evaluate the controller performance.

In chapter 5, the implementation of the controller is discussed. It starts by explaining how the communication of the FST06e works. Based on the communication protocol a workflow for the controller is proposed. It starts by receiving the messages, converting them to SI units, then the necessary computations are done and the values converted again to messages to be sent into the car.
This workflow was tested in real time simulation using simulink to evaluate if it would work within the desired time. Furthermore it was also possible to test input signals, such as steering δ, velocity v and pedal value, output signals such as torque T and reference signal $\dot{\psi}_{des}$, saving a lot testing time with the real vehicle.

Additionally, based on the experience acquired with testing the vehicle, safety features where added to ensure that in case of failure the car would stop in a controlled way.
Based on testing with the FST06e, it is concluded that the using the GPS signal for the velocity proved unreliable. To change this, the velocity of the wheels is used to measure the velocity of the vehicle. Also the measures of the yaw rate from the IMU proved too noisy, for that it was necessary to design a first order low pass filter.

Results shown in chapter 6 are summarized in table 6.1 and figure 6.7. These tests were done in the same conditions as the ones used to validate the model. The tests show that the proposed controller exhibits an increase in lateral performance, which translates in a reduction of 0.4s per lap. The effort to reduce complexity as much as possible during the design phase, and with sensor data validation, open loop communication, and a well planned C code are the reasons for the success on the implementation of the controllers.

In the end, the controller contributed to a 5% performance gain, which translates into a lap time gain of 4.59s, 0.38s quicker than without torque vectoring.

Table 6.1: Comparison between torque vectoring and no torque vectoring

	Yaw Rate [deg/s]	Velocity [m/s]	Time [s]
No TV	70	8.4	4.97
TV	74	8.8	4.59

6.3 Future Work and Research

As stated in the thesis objectives in section 1.4, physical implementation of the controller was a concern. With this in mind, during the modelling and design phases special attention has already taken into proving the feasibility and workflow for the design of a TV controller. Further research on the following topics is recommended:

- In this thesis the Burckhard method is used to calculate the forces of the tires. This method gives a relatively accurate value of friction without much knowledge of the tires. When more knowledge of the tires is available, more complex models, such as Pajecka or Dugoff's, can be used. These models capture the dynamic behavior of the tires, therefore giving more accurate forces values.
- The steering angle δ of the front wheels is determined using the steering Ackerman which is a model based on the kinematics of the steering. Submitting the car to a kinematics and compliance test (K&C) would allow to have an accurate representation of the steering kinematics.
- In chapter 5 the simulation with the microcontroller was in open loop, which allowed testing the signals and fail safe system, but not the controller. So a closed loop simulation is suggested.
- An additional improvement to the controllers would be to calculate the vertical load at the wheels, so instead of evenly distributing torque, it could be distributed based on the vertical load. For this, it is necessary to have linear potentiometers in the shock absorbers, to directly measure the vertical load.
- Another improvement would be to also implement a controller for the slippage of the wheels. For ratios bigger than 15% between the velocity of the car and the wheel, adding more torque will not be useful. So bounding the maximum torque would help the overall dynamic of the car.
- Adding a Kalman Filter to estimate the lateral velocity, and thus using the LQR to the full extent.
- Having a direct and easy access to the logged data from the vehicle would reduce the testing time with the vehicle. Ideally having a WI-FI connection in order to see the data in real time.

Bibliography

[1] R. B. Dieter Schramm, Manfred Hiller. *Vehicle Dynamics: Modeling and Simulation*. Springer, 1^{st} edition, 2014. ISBN 978-3-540-36045-2.

[2] R. Rajamani. *Vehicle Dynamics and Control, Mechanical Engineering Series*. Springer, 1^{st} edition, 2005. ISBN 978-0387263960.

[3] R. N. Jazar. *Vehicle Dynamics: Theory and Application*. Springer, 2^{nd} edition, 2014. ISBN:978-1-4614-8544-5.

[4] C. Smith. *Tune to Win: The art and science of race car development and tuning*. Aero Publishers, 1^{st} edition, 1978. ISBN 978-0879380717.

[5] L. N. Uwe Kiencke. *Automotive Control Systems: For Engine, Driveline and Vehicle*. Springer, 2^{nd} edition, 2005. ISBN 978-3-540-26484-2.

[6] T. D. Gillespie. *Fundamentals of Vehicle Dynamics*. Society of Automotive Engineers, 1^{st} edition, 1992. ISBN 978-1560911999.

[7] G. Kaiser, Q. Liu, C. Hoffmann, M. Korte, and H. Werner. Torque vectoring for an electric vehicle using an lpv drive controller and a torque and slip limiter. *2012 IEEE 51st IEEE Conference on Decision and Control (CDC)*, pages 5016–5021, Dec 2012. ISSN 0191-2216. doi: 10.1109/CDC.2012.6426553.

[8] J. Ghosh, A. Tonoli, and N. Amati. A torque vectoring strategy for improving the performance of a rear wheel drive electric vehicle. *2015 IEEE Vehicle Power and Propulsion Conference (VPPC)*, pages 1–6, Oct 2015.

[9] C. Zhao, W. Xiang, and P. Richardson. Vehicle lateral control and yaw stability control through differential braking. pages 384–389, July 2006. ISSN 2163-5137. doi: 10.1109/ISIE.2006.295624.

[10] *Passengers cars – Steady-state circular driving behaviour – Open-loop test methods ISO 4138:2004*. Fourth edition edition, June 2012.

[11] L. D. Novellis, A. Sorniotti, P. Gruber, and A. Pennycott. Comparison of feedback control techniques for torque-vectoring control of fully electric vehicles. *IEEE Transactions on Vehicular Technology*, 63(8):3612–3623, Oct 2014. ISSN 0018-9545.

[12] A. Stoop. Design and implementation of torque vectoring for the forze racing car. Master's thesis, Delft Center for Systems and Control, 2014.

[13] D. Rubin and S. Arogeti. Vehicle yaw stability control using rear active differential via sliding mode control methods. *21st Mediterranean Conference on Control and Automation*, pages 317–322, June 2013. doi: 10.1109/MED.2013.6608740.

[14] R. Rajamani and D. N. Piyabongkarn. New paradigms for the integration of yaw stability and rollover prevention functions in vehicle stability control. *IEEE Transactions on Intelligent Transportation Systems*, 14(1):249–261, March 2013.

[15] S. Anwar. Yaw stability control of an automotive vehicle via generalized predictive algorithm. *Proceedings of the 2005, American Control Conference, 2005.*, pages 435–440, June 2005. ISSN 0743-1619. doi: 10.1109/ACC.2005.1469974.

[16] S. D. Cairano, H. E. Tseng, D. Bernardini, and A. Bemporad. Vehicle yaw stability control by coordinated active front steering and differential braking in the tire sideslip angles domain. *IEEE Transactions on Control Systems Technology*, 21(4):1236–1248, July 2013. ISSN 1063-6536.

[17] L. Xiong, Y. H. Gan, Y. Feng, and F. Martinez. A torque vectoring control system for maneuverability improvement of 4wd ev. *Instruments, Measurement, Electronics and Information Engineering*, 347: 899–903, October 2013. doi: 10.4028347-350.899.

[18] G. Kaiser, F. Holzmann, B. Chretien, M. Korte, and H. Werner. Torque vectoring with a feedback and feed forward controller - applied to a through the road hybrid electric vehicle. *2011 IEEE Intelligent Vehicles Symposium (IV)*, pages 448–453, June 2011.